Designs with angelic nondeterminism

Pedro Ribeiro

Department of Computer Science
University of York

Technical Report

Revision: 704f887 (2014-02-04)

Supervisor: Ana Cavalcanti

February 2013

Abstract

The Unifying Theories of Programming (UTP) of Hoare and He is a predicative framework of relations suitable for the rigorous study of different programming paradigms. It promotes the reuse of results through the linking of theories. Particular aspects of programs can also be studied in isolation.

In the UTP, the theory of designs provides not only a model for terminating programs (where pre and postcondition pairs can be specified), but also a basis for characterising state-rich concurrent and reactive programs. These are programs whose interactions with the environment cannot simply be described by relations between inputs and outputs. In this context, process calculi such as Communicating Sequential Processes (CSP) and *Circus* have been given semantics in the UTP through the theory of reactive designs.

Angelic nondeterminism is a useful specification construct that allows for a high degree of abstraction. It has traditionally been studied within the refinement calculus. Previous work has proposed a theory of angelic nondeterminism in the UTP through a predicative model of binary multirelations. Such models, however, can only model terminating programs. In this report we propose a new UTP theory of designs with angelic nondeterminism with the long-term aim of developing a model for process calculi.

Contents

Chapter 1

Introduction

In this chapter we present the motivation and objectives of this work. The motivation is presented in Section 1.1, while the objectives are discussed in Section 1.2. In Section 1.3 we discuss the overall approach followed in this work and how our model relates to existing theories. Finally the structure of this report is outlined in Section 1.4

1.1 Motivation

The UTP of Hoare and He [1] provides a relational framework suitable for the study of different programming paradigms. Relations are characterised by their alphabet and a predicate that restricts the possible values of the variables in the alphabet. The alphabet consists of program variables and auxiliary variables that capture additional information, such as time. A collection of UTP theories exist for a variety of programming paradigms and techniques such as concurrency [1], logic programming [1], high-order programming [1], object-orientation [2], pointers [3], time [4–6] and others. The UTP distinguishes itself in the ability to promote unification of results through the linking of theories, while allowing concepts to be studied in isolation.

The theory of designs is the definitive treatment of total correctness for sequential programs in the UTP. It considers an alphabet containing program variables as well as auxiliary variables that capture the start and termination of programs. Designs can be understood as encoding the traditional pre and postcondition pairs. In order to characterise reactive programs, the relationship between initial and final states is not sufficient. Instead, intermediate information also needs to be recorded [7]. This is captured in the UTP through the theory of reactive programs that includes additional observational variables for this purpose.

The combination of the theory of designs and the theory of reactive processes characterises theories for process calculi such as CSP [8] and *Circus* [9, 10]. Every predicate of the theory of CSP can be specified as a reactive design [1]. These are designs whose preconditions depend on observations of the final or later values of the variables, and whose termination is not necessarily guaranteed. This corresponds to designs that do not necessarily satisfy the healthiness condition **H3** [1], a necessary condition to establish the link with the theory of CSP [1].

Angelic nondeterminism is a useful abstraction in the context of formal specifications. It has traditionally been studied in the refinement calculus [11–13] through the monotonic predicate transformers. There it is defined precisely as the dual of demonic nondeterminism. Its characterisation in a relational setting, such as that of the UTP, however, is more challenging and has required the use of multirelational models [14].

Multirelations are relations that relate initial states to sets of final states. In [15] Rewitzky presents the foundational work on multirelations that allows both forms of nondeterminism to be expressed in the same relational model. The set of final states can either be interpreted as encoding angelic or demonic choices. If the sets of final states encode angelic choices, then the relation between an initial state and sets of final states encodes demonic choices, or vice-versa. The model of up-closed binary multirelations is the most important as it has a lattice-theoretic structure [15].

In [14], Cavalcanti et al. propose a UTP theory based on multirelations that can encode angelic nondeterminism. Although the model in [14] does not make use of the observational variables of the original theory of designs, it captures termination. Its focus on sequential programs makes it not applicable to reactive programs.

Morris and Tyrrel [16–19], and Hesselink [20] have pursued the modelling of both types of nondeterminism at the expression or term level. Their focus is on functional languages. Tyrrell et al. [21] have attempted an axiomatization for an algebra resembling CSP where external choice is referred to as "angelic choice", however this is different from standard CSP semantics [8].

In summary, despite the different attempts at modelling angelic nondeterminism, to the best of our knowledge, no suitable model has been developed for process calculi. The model that we propose in this work presents a first step towards addressing this problem.

1.2 Objectives

In light of our discussion, in this work we propose a new UTP theory of designs that is capable of expressing both demonic and angelic nondeterminism. In order to exploit existing theories, it is our aim to develop a theory that uses the auxiliary variables of the original theory of designs. Furthermore, as a prerequisite for modelling reactive programs, such a theory needs to encompass designs whose preconditions refer to the value of final states.

In addition, it is essential that we can validate the model we propose with respect to the existing theories. Following the spirit of the UTP, we explore the relationship with both the theory of binary multirelations [15] and that of [14] by establishing links with them.

In our account we define the basic operators of the new theory and prove expected properties based on results from the literature. Since we provide a new model where preconditions may refer to the final set of states, not all results are immediately obvious. This is the case, for instance, for the sequential composition operator.

1.3 Overview of theories

As an aid to the development of our theory, we develop an extended model of binary multirelations. This isomorphic model provides insights into the definition of certain aspects of the theory, such as the sequential composition operator, whose definition is not trivial.

Below we provide an overview of the relationship between the theory proposed and an extended binary multirelational model. Their respective relationship with each of the established models of [14] and [15] is also discussed. The overall relationship between the theories is illustrated in Figure 1.1. Each theory is named after its characterising healthiness condition and the respective isomorphisms are established by pairs of functions. The definition of these is established in Chapters 3 to 5, while a full account of the existing theories can be found in [14, 15].

In the UTP, the theory of designs is characterised by the healthiness

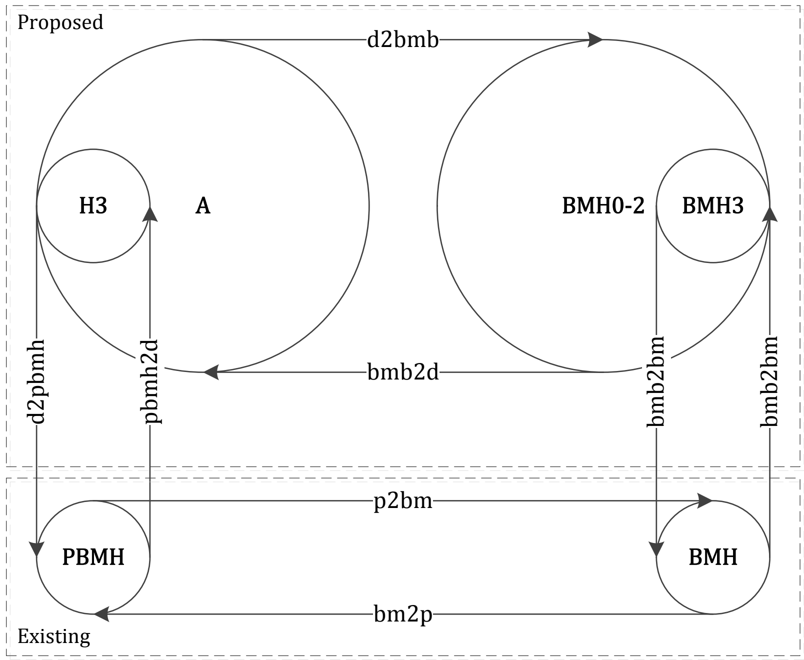

Figure 1.1: Link between the theories.

conditions **H1** and **H2** [1]. The theory that we propose is, in addition, characterised by the healthiness condition **A**. It is based on that of [14], whose only healthiness condition is specified by the function **PBMH**.

Since the precondition of reactive designs may impose requirements on final states, this is also allowed in our theory. As a result, it becomes possible for designs to specify sets of final states available for angelic choice, even when termination is not guaranteed. This means that designs in our theory do not necessarily satisfy the healthiness condition **H3** of designs.

In order to motivate the development of the new model, we develop an isomorphic model that can describe **A**-healthy designs as an extended version of binary multirelations. The difference with respect to the original model of binary multirelations [15] is that we can distinguish sets of final states that terminate from those that may not terminate. This binary multirelational model is characterised by the healthiness conditions **BMH0-BMH2**. Its subset that corresponds to the original binary multirelations is characterised, in addition, by the healthiness condition **BMH3**.

Finally, in this report we establish that both models that we propose are isomorphic through the pair of functions $d2bmb$ and $bmb2d$. In the following section we focus on subsets of interest of both models and their relationship with the existing theories.

Cavalcanti et al. [14] establish that their UTP model is isomorphic to the model of up-closed binary multirelations, whose healthiness condition we denote as **BMH**. This relationship is established by a pair of functions, $p2bm$ and $bm2p$ [14], respectively, which we include as part of Figure 1.1.

The functions $d2pbmh$ and $pbmh2d$ establish that the model of **A**-healthy designs that also satisfy the healthiness condition **H3** is isomorphic to that of [14]. Finally, the pair of functions $bmb2bm$ and $bmb2bm$ establish that the subset of the extended binary multirelational model that also satisfies **BMH3** is isomorphic to the original model of binary multirelations [15].

This concludes our overview on how the theory that we propose relates to both the extended binary multirelational model and the existing theories.

1.4 Outline

In Chapter 2 the UTP is introduced based on the full account in [1]. The general notions of UTP theories are presented, followed by the theory of designs. We also briefly explain how theories can be related in the UTP.

Chapter 3 introduces the original theory of binary multirelations [15]. This includes the healthiness conditions, the refinement ordering and the main operators of the theory.

In Chapter 4 we introduce an extended model of binary multirelations that can cater for sets of final states that may not terminate. The healthiness conditions are defined and the main operators presented. In addition, we study the subset that is isomorphic to the original theory of binary multirelations.

Chapter 5 describes the new UTP theory of designs with angelic nondeterminism. It introduces the healthiness conditions and defines the main operators. Likewise, we also study the subset that is isomorphic with the model of [14].

Finally, in Chapter 6 we present the conclusions of this work, including indications for future work.

Chapter 2

UTP

In this chapter we introduce the underlying mathematical theory [1] used in the definition of the theories of interest: the UTP. We begin by characterising the components of UTP theories in Section 2.1. We then focus our attention on the theory of designs in Section 2.2. Finally, we explain how theories can be linked in Section 2.3. A full account on the UTP and the theory of designs can be found in [1, 22].

2.1 Theories

The UTP of Hoare and He [1] is a relational mathematically rigorous approach to characterising and reasoning about programs based on the principle of observation. The UTP promotes unification while allowing different aspects of programs to be considered in isolation. In [1] a collection of theories are presented that target multiple aspects of different programming paradigms, such as functionality, concurrency, logic programming and high-order programming. Recent publications have added to the strength of the UTP by proposing new theories capable of handling angelic nondeterminism [23], object-orientation [2], pointers [3], time [4–6] and others.

A UTP theory is characterised by three main components: an alphabet, a set of healthiness conditions and a set of operators. In Section 2.1.1 we introduce the notion of an alphabet. In Section 2.1.2 we discuss how the healthiness conditions characterise a theory. Finally, in Section 2.1.3 the core notion of refinement in the UTP is explained followed by the operators of theories in Section 2.1.4.

2.1.1 Alphabets

The alphabet of a UTP theory consists of a set of variables that can take values corresponding to observations made of a program behaviour. These can be either program variables, or alternatively, auxiliary variables that capture information like termination, execution time, and so on. Similar to the conventions of Z, in the UTP initial states are characterised by a set of undashed variables (for example, the set: $\{ok, v\}$), while final or subsequent states are characterised by a set of dashed variables (for example, the corresponding set: $\{ok', v'\}$).

A UTP relation consists of an alphabet and a logical predicate over the variables in its alphabet that describes the relationship between initial and after states. For example, in the case of a program whose only purpose is incrementing the initial value of x we could describe it using the relation: $x' = x + 1$. This relation concisely describes all pairs of values (x, x') that satisfy the given predicate. Thus relations characterise the possible observations of a program.

The alphabet of a relation is split into two disjoint subsets: the set of undashed variables characterises the input values while the set of dashed variables characterises the after values. For a relation R these are specified by $in\alpha(R)$ and $out\alpha(R)$, for the input and output alphabets, respectively.

A relation is homogeneous if and only if the input and output alphabets are exactly the same, except for the fact that variables are undashed and dashed in either set, respectively. This is formally captured by the following definition, where $(in\alpha(R))'$ is the set of variables obtained by dashing every variable contained in the set $in\alpha(R)$.

Definition 1 (Homogeneous relation) *A relation R is homogeneous if and only if $(in\alpha(R))' = out\alpha(R)$.*

When defining a theory it is also necessary to restrict the set of predicates that are valid in a given theory. This is addressed by defining healthiness conditions.

2.1.2 Healthiness conditions

In the UTP, the set of predicates valid in a certain theory is defined by what are known as healthiness conditions. These are normally specified by idempotent monotonic functions whose fixed points are the valid predicates of the theory. These properties ensure that correctness is preserved through refinement.

For instance, in the context of theories concerning time, it is often possible to make observations of a system in discrete-time units using a variable t. It is expected that any plausible theory describing such a system must guarantee that time is increasingly monotonic, thus this can be enforced by defining the healthiness condition HC.

Example 1

$$HC(P) \mathrel{\widehat=} P \land t \le t'$$

This healthiness condition is defined in terms of conjunction, so it is called a conjunctive healthiness condition [3]. A general result on conjunctive healthiness conditions [3] enables us to establish that HC is idempotent and monotonic with respect to the refinement ordering. An observation in this theory is valid if and only if it is a fixed point of HC.

2.1.3 Refinement

The theory of relations forms a complete lattice [1], where the ordering is given by (reverse) universal implication. The top of the lattice is *false* and the bottom is *true*. This ordering corresponds to the notion of refinement. Its definition is presented below, where the square brackets stand for universal quantification over all the variables in the alphabet [1].

Definition 2 (Refinement)

$$P \sqsubseteq Q \mathrel{\widehat=} [Q \Rightarrow P]$$

Refinement can be understood as preserving the notion of correctness in the sense that, if a predicate Q refines P, then all possible behaviours exhibited by Q are permitted by P. This notion is paramount for the UTP framework and it is the same across the different theories. The relation *true* imposes no restriction and permits the observation of any value for all variables in the alphabet, while *false* permits none.

2.1.4 Operators

A UTP theory comprises a number of operators that characterise how the theory may be used algebraically to specify more complex behaviours. In the theory of relations there are a number of core operators that correspond to typical constructs found in programming languages, such as assignment (:=), conditional ($A \lhd c \rhd B$), and sequential composition (;). In what follows we present some of the most important operators of the theory of relations.

Sequential composition

In UTP theories whose relations are homogeneous, sequential composition is defined in a consistent way through the notion of substitution as shown in the following definition.

Definition 3 (Sequential composition)

$$P \; ; \; Q \mathrel{\widehat=} \exists\, v_0 \bullet P[v_0/v'] \land Q[v_0/v]$$

The intuition here is that the sequential composition of two relations P and Q involves some intermediate, unobservable state, whose vector of variables is represented by v_0. This vector is substituted in place for the final values of P, as represented by v', as well as substituted for the initial values of Q, as represented by v. It is finally hidden by the existential quantifier.

Skip

An important construct in the relational theory is the program $\mathit{I\!I}_{\mathcal{R}}$, otherwise also known as **Skip**, whose definition is presented below.

Definition 4 (Skip)

$$\mathit{I\!I}_{\mathcal{R}} \mathrel{\widehat=} (v' = v)$$

This is a program that always terminates successfully and upon termination guarantees that all variables maintain their initial values. The most interesting property of $\mathit{I\!I}_{\mathcal{R}}$ is that it is the left-unit for sequential composition [1].

Demonic choice

Due to the lattice-theoretic approach of the UTP, demonic choice ($\sqcap$) corresponds to the greatest lower bound of the refinement ordering. This means that its definition is simply disjunction.

Definition 5 (Demonic choice)

$$P \sqcap Q \mathrel{\widehat=} P \vee Q$$

Unfortunately the least upper bound, which is conjunction, does not correspond to the notion of angelic choice. As mentioned previously, it is not possible to represent both choices directly within the relational model, unless a binary multirelational model is used [14].

Recursion

Recursion is defined in the UTP as the weakest fixed point. Since we have a complete lattice it is possible to find a complete lattice of fixed points due to a result by Tarski [1, 24]. In the following definition F is a monotonic function and $\bigsqcap$ is the greatest lower bound.

Definition 6 (Recursion)

$$\mu X \bullet F(X) \mathrel{\widehat=} \bigsqcap\{X \mid [F(X) \sqsubseteq X]\}$$

A non-terminating recursion, such as $(\mu\, Y \bullet Y)$, is equated with the bottom of the lattice, *true* [1]. Intuitively this means that it does not terminate, but if we sequentially compose this recursion with another program, then it becomes possible to recover from the non-terminating recursion as shown in the following example [22].

Example 2

$$\begin{aligned}
&(\mu\, Y \bullet Y) \mathbin{;} x' = 0 && \{\text{Definition of recursion}\}\\
&= \bigsqcap\{X \mid [(\mu\, Y \bullet Y)(X) \sqsubseteq X]\} \mathbin{;} x' = 0 && \{\text{Function application}\}\\
&= \bigsqcap\{X \mid [X \sqsubseteq X]\} \mathbin{;} x' = 0 && \{\text{Reflexivity of } \sqsubseteq\}\\
&= \bigsqcap\{X \mid \mathit{true}\} \mathbin{;} x' = 0 && \{\text{Property of } \sqcap\}
\end{aligned}$$

$$
\begin{aligned}
&= \mathit{true}\ ;\ x' = 0 && \{\text{Definition of sequential composition}\}\\
&= \exists\, v_0 \bullet \mathit{true} \land x' = 0 && \{\text{Propositional calculus}\}\\
&= x' = 0
\end{aligned}
$$

This issue motivated the definition of the theory of designs that we present in the following section.

2.2 Designs

As already mentioned, when considering theories of total correctness for reasoning about programs, the theory of relations is not appropriate due to the fact that it is possible to recover from non-terminating programs successfully [1, 22]. In other words, the bottom of the lattice, *true*, is not necessarily a left-zero of sequential composition as would be needed. As a result, Hoare and He [1] have introduced the theory of designs, which addresses this issue.

2.2.1 Alphabet

The theory of designs is defined by considering the addition of two auxiliary variables to the alphabet: ok and ok'.

$$ok, ok' : \{\mathit{true}, \mathit{false}\}$$

Their purpose is to track whether a program has been started, in which case ok is *true*, and whether a program has successfully terminated, in which case ok' is *true*.

In the following section we present the healthiness conditions that define the theory of designs. Finally we discuss how designs can be refined.

2.2.2 Healthiness conditions

Any valid predicate of this theory has to obey two basic principles: that no guarantees can be made by a program before it has started, and, that no program may require non-termination. These two principles are formally characterised by the healthiness conditions **H1**, and **H2**, respectively [1]. We include their definitions [1] below.

Definition 7 (H1)

$$\mathbf{H1}(P) \mathrel{\widehat{=}} ok \Rightarrow P$$

The definition of **H1** states that any guarantees made by P can only be established once it has started. Otherwise, any observation is permitted and it behaves like the bottom of the lattice, which is the same as the one for relations: *true*.

Definition 8 (H2)

$$\mathbf{H2}(P) \mathrel{\widehat{=}} \neg\, P[\mathit{false}/ok'] \Rightarrow (P[\mathit{true}/ok'] \wedge ok')$$

The definition of **H2** states that if it is possible for a program P not to terminate, that is with ok' being *false*, then it must also be possible for it to terminate, that is with ok' being *true*. The definition presented here is equivalent to that originally presented by Hoare and He [1], but instead considers **H2** in isolation. In Appendix A we prove that it is equivalent.

A predicate that is both **H1** and **H2** satisfies the following property of designs.

Law 2.2.1 (H1 ∘ H2)

$$\mathbf{H1} \circ \mathbf{H2}(P) = (ok \wedge \neg\, P[\mathit{false}/ok']) \Rightarrow (P[\mathit{true}/ok'] \wedge ok')$$

Here the design is split into two parts: a precondition and a postcondition. It is defined using the notation of Hoare and He [1] as shown in the following definition.

Definition 9 (Design)

$$(P \vdash Q) \mathrel{\widehat{=}} (ok \wedge P) \Rightarrow (ok' \wedge Q)$$

In fact, a design is more commonly written using the following notation, where we use the shorthand notation $P^a = P[a/ok']$, with $t = \mathit{true}$ and $f = \mathit{false}$, as introduced by Woodcock and Cavalcanti [22].

Law 2.2.2 (Design) *A predicate P is a design if and only if it can be written in the following form*

$$\mathbf{H1} \circ \mathbf{H2}(P) = (\neg\, P^f \vdash P^t)$$

It is worth noting that the functions **H1** and **H2** (and indeed all of the healthiness conditions of designs) are idempotent and monotonic with respect to refinement [1]. Furthermore none of the proofs establishing these results rely on the property of homogeneity. Therefore it is possible to define a non-homogeneous theory of designs.

Hoare and He [1] identified another two healthiness conditions of interest which we discuss further below. The third healthiness condition **H3** requires $I\!I_{\mathcal{D}}$, the **Skip** of designs, to be a right-unit for sequential composition [1].

Definition 10 ($I\!I_{\mathcal{D}}$)

$$I\!I_{\mathcal{D}} \mathrel{\widehat{=}} (\mathit{true} \vdash v' = v)$$

Skip is a program that always terminates successfully and does not change the program variables.

Definition 11 (H3)

$$\mathbf{H3}(P) \mathrel{\widehat{=}} P \ ; I\!I_{\mathcal{D}}$$

From this definition it may not be immediately obvious how designs are further restricted by **H3**. In fact, it requires the precondition not to have any dashed variables (as confirmed by Theorem 2.2.1). In order to understand the intuition behind it we consider an example of a design that is not **H3**-healthy.

Example 3

$$\begin{aligned}
&(x' \neq 2 \vdash \mathit{true}) && \{\text{Definition of designs}\}\\
&= (ok \land x' \neq 2) \Rightarrow ok' && \{\text{Propositional calculus}\}\\
&= ok \Rightarrow (x' = 2 \lor ok')
\end{aligned}$$

In this case we have a program that upon having started can either terminate and any final values are permitted, or can assign the value 2 to the variable x and termination is then not required. In the context of a theory of total correctness for sequential programs this is a behaviour that would not normally be expected. However it is worth noting that in the context of reactive processes non **H3**-designs are important, since there are some requirements imposed on programs even when they diverge [7, 14].

The healthiness condition **H3** can also be interpreted as guaranteeing that if a program may not terminate, then it has arbitrary behaviour. Thus a predicate that is **H3**-healthy is also necessarily **H2**-healthy [14].

If we expand the definition of **H3** by applying the definition of sequential definition for designs we obtain the following result [1, 22].

Theorem 2.2.1 (P-sequence-$I\!I_{\mathcal{D}}$)

$$(\neg P^f \vdash P^t) = (\neg P^f \vdash P^t) \mathbin{;} I\!I_{\mathcal{D}} \Leftrightarrow \neg P^f = \exists v' \bullet \neg P^f$$

This theorem shows that the value of any dashed variables in $\neg P^f$ must be irrelevant. Therefore any design that is **H3**-healthy can only have a condition as its precondition, that is, a predicate that only mentions undashed variables, and thus can only impose restrictions on previous programs.

Finally the last healthiness condition of interest is **H4** that restricts designs to feasible programs. It is defined by the following algebraic equation [1] that requires that *true* be a right-zero.

Definition 12 (H4)

$$P \mathbin{;} true = true$$

The intuition here is that this prevents the top of the lattice, **Miracle**, itself a trivial refinement of any program, from being allowed. In order to understand the reason for this, consider the definition of **Miracle**.

Definition 13 (Miracle)

$$\begin{aligned}
\mathbf{Miracle} &\mathrel{\widehat=} (true \vdash false) && \{\text{Property of designs}\}\\
&= ok \Rightarrow false && \{\text{Propositional calculus}\}\\
&= \neg\, ok
\end{aligned}$$

Miracle represents a program that could never be started ($\neg\, ok$). Furthermore, if it could, and indeed its precondition makes no restriction, it would establish the impossible: *false*. Any conceivable implementable program must not behave in this way. However, **Miracle** is an important construct in refinement calculi [14, 22].

For completeness we also provide the definition of the bottom of the lattice of designs, which is called **Abort**. There are in fact two possible ways of expressing it as a design.

Definition 14 (Abort)

$$
\begin{aligned}
\mathbf{Abort} \mathrel{\widehat{=}} & (\mathit{false} \vdash \mathit{true}) && \{\text{Property of designs}\} \\
= & (\mathit{false} \wedge \mathit{ok}) \Rightarrow \mathit{ok}' && \{\text{Propositional calculus}\} \\
= & (\mathit{false} \wedge \mathit{ok}) \Rightarrow (\mathit{false} \wedge \mathit{ok}') && \{\text{Property of designs}\} \\
= & (\mathit{false} \vdash \mathit{false})
\end{aligned}
$$

Abort provides no guarantees at all: it may fail to terminate, and if it does terminate there are no guarantees on the final values. Indeed it is not required to guarantee anything at all since its precondition is *false*.

2.2.3 Operators

In the following theorems we introduce the meet and join of the lattice of designs as presented in [22]. Like in the lattice of relations, the greatest lower bound corresponds to demonic choice.

Theorem 2.2.2 (Greatest lower bound)

$$\textstyle\bigsqcap_i (P_i \vdash Q_i) = (\bigwedge_i P_i) \vdash (\bigvee_i Q_i)$$

Theorem 2.2.3 (Least upper bound)

$$\textstyle\bigsqcup_i (P_i \vdash Q_i) = (\bigvee_i P_i) \vdash (\bigvee_i P_i \Rightarrow Q_i)$$

Sequential composition

The definition of sequential composition for designs can be deduced from Definition 3. Here we present the result as proved in [1, 22].

Theorem 2.2.4 (Sequential composition of designs)

$$(P_0 \vdash P_1) \; ; (Q_0 \vdash Q_1) = (\neg\, (\neg\, P_0 \; ; \mathit{true}) \wedge \neg\, (P_1 \; ; \neg\, Q_0) \vdash P_1 \; ; Q_1)$$

This definition can be interpreted as establishing P_1 followed by Q_1 provided that P_0 holds and P_1 satisfies Q_0. As pointed out in [22] if P_0 is a condition then the definition can be further simplified.

2.2.4 Refinement

As in other UTP theories, the refinement ordering in the theory of designs is the same: universal (reverse) implication. This can be used to establish the following result [22].

Theorem 2.2.5 (Refinement)

$$(P_0 \vdash P_1) \sqsubseteq (Q_0 \vdash Q_1) = [P_0 \land Q_1 \Rightarrow P_1] \land [P_0 \Rightarrow Q_0]$$

Theorem 2.2.5 confirms the intuition about refinement as found in other calculi: preconditions can be weakened while postconditions can be strengthened.

This section concludes our overview of the theory of designs. In the following section we focus on how theories can be related and combined.

2.3 Linking theories

The UTP provides a very powerful framework that allows relationships to be established between different theories. This means that results in different theories can be re-used. We elaborate on some of principles behind the linking of theories in the following sections. A full account is available in [1].

Following the convention of Hoare and He [1] we assume the existence of a pair of functions L and R that map one theory into another: L maps the (potentially) more expressive theory into the (potentially) weaker theory and R vice-versa.

2.3.1 Subset theories

The simplest form of relationship that can be established is that between subset theories [1]. Consider the case where a theory T is a subset of S, then it is possible to find a function $R : T \mapsto S$ which is simply the identity [1]. Defining $L : S \mapsto T$ for the reverse direction may be slightly more complicated as the subset theory is normally less expressive.

Hoare and He [1] pinpoint the most important properties of such a function $L : S \mapsto T$: weakening or strengthening, idempotence and ideally monotonicity. As highlighted in [1] monotonicity is not always necessarily observed. We reproduce the respective definitions below.

Definition 15 (Weakening)

$$\forall X \in S \bullet L(X) \sqsubseteq X$$

Definition 16 (Strengthening)

$$\forall X \in S \bullet X \sqsubseteq L(X)$$

We follow Hoare and He's convention and refer to a function that is both weakening and idempotent as a *link* and, if it is also monotonic we refer to it as a *retract*.

2.3.2 Bijective links

When two theories have equal expressive power, the pair of linking functions between them can be proved to form a bijection. In other words, each function undoes exactly the other and thus as expected the following identities hold.

Definition 17 (Bijection) *A function L is a bijection if and only if $R = L^{-1}$, where the inverse function of L, $L^{-}1$ exists, and the following identities hold for all P.*

$$L \circ R(P) = P \wedge R \circ L(P) = P$$

A bijection constitutes the strongest form of relationship between theories. It can apply even when the alphabets are different or when theories are presented in different styles [1]. Indeed this is often what is sought: proving that two theories have exactly the same expressive power, yet their shape may suit different contexts better.

2.3.3 Galois connections

Often, though, and as seen previously in subset theories, a theory is more expressive than its counterpart. Therefore the linking function is not a bijection as there has to be some weakening or strengthening in either direction. A pair of functions describing this relationship constitutes what is known as a Galois connection. Here we reproduce the definition of [1] and provide a pictorial illustration in Figure 2.1.

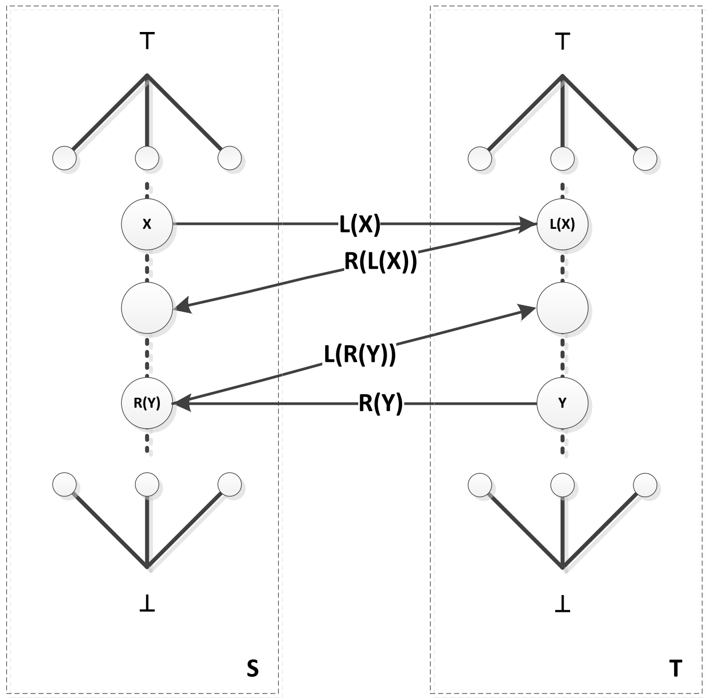

Figure 2.1: Galois connection between two lattices, S and T.

Definition 18 (Galois connection) *Let S and T be lattices, and let $L : S \mapsto T$ and $R : T \mapsto S$, the pair (L, R) is a Galois connection if and only if for all $X \in S$ and $Y \in T$:*

$$R(Y) \sqsubseteq X \Leftrightarrow Y \sqsubseteq L(X)$$

As pointed out earlier, a bijection presents a stronger relationship than a Galois connection. However, it is not the case that every bijection is a Galois connection [1]. Hoare and He [1] give the example of negation whose inverse is precisely itself, however negation is not monotonic.

2.4 Final considerations

The UTP framework provides a way of rigorously formalising programs in a relational setting. A UTP theory consists of an alphabet, a set of of healthiness conditions and a set of operators, whose syntax forms the *signature* of the theory [1]. The most general theory in the UTP is that of relations. Unfortunately it is not sufficient on its own to appropriately define theories of total correctness for programs.

The theory of designs provides a compromise, where by extending the alphabet with additional observational variables, termination can be characterised appropriately. The set of valid predicates is defined by a set of healthiness conditions: **H1** and **H2** characterise designs, and equally determine a unique syntactic form. The other healthiness conditions, while optional, are also important from the point of view of refinement of sequential programs. However in the context of theories such as those for reactive processes it is essential that we can consider designs that are not necessarily **H3**-healthy.

Finally we have briefly considered how UTP theories can be related. This is achieved by linking functions that can map predicates from one theory into another. When considering theories that have equal expressive power the linking function is a bijection. However, often theories have different expressive power, therefore there must be some weakening or strengthening. In this case the pair of linking functions forms a Galois connection. In addition, it is also possible to establish relationships with sub theories. The importance of these linking functions is that results can be borrowed from other theories and then re-used in different contexts. This forms part of the toolkit that is in the essence of the UTP: unification.

Chapter 3

Binary multirelations

In this chapter the theory of binary multirelations [15] is presented. In Section 3.1 the theory is introduced and formally defined. The signle healthiness condition of the theory is explored in Section 3.2 along with its characterisation as a fixed point. Section 3.3 describes the refinement ordering and its extreme points. Finally, the operators are presented in Section 3.4.

3.1 Introduction

A binary multirelation, an element of a type named *BM* here, is a relation between an initial program state and a set of final states, where a *State* is the type of records with a component for each program variable.

Definition 19

$$BM == State \leftrightarrow \mathbb{P}\, State$$

For instance, the program that assigns the number 1 to the only program variable x when started from any initial state is defined as follows.

Example 4

$$(x := 1)_{BM} \mathrel{\widehat=} \{s : State, ss : \mathbb{P}\, State \mid (x \mapsto 1) \in ss\}$$

Following [14], the notation $(x \mapsto 1)$ denotes a record whose only component is x and its respective value is 1.

The binary multirelational model is richer than the relational model in that it relates each initial state to a set of final states. This set can be interpreted as either encoding angelic or demonic choices, depending on which model is chosen [14, 15]. In our discussion we choose to present a model where the set of final states encodes angelic choices. This deliberate choice is justified in [14, 25] as maintaining the refinement order of the isomorphic UTP model introduced in [14]. Since it is our goal to study an extended version of binary multirelations and its relationship with an equivalent UTP model, it is desirable also in our context that the refinement order is maintained.

Demonic choices are encoded by the different ways in which the set of final states can be chosen. For example, the program that angelically assigns the value 1 or 2 to the only program variable x is specified by the following relation, where $\sqcup_{BM}$ is the angelic choice operator for binary multirelations.

Example 5

$$\begin{aligned}&(x := 1)_{BM} \sqcup_{BM} (x := 2)_{BM}\\ &=\\ &\{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State} \mid (x \mapsto 1) \in ss \land (x \mapsto 2) \in ss\}\end{aligned}$$

This definition allows any superset of the set $\{(x \mapsto 1), (x \mapsto 2)\}$ to be chosen. The choice of values 1 and 2 for the program variable x are available in every set of final states ss, and so are available in every demonic choice.

3.2 Healthiness conditions

In general, not all relations of type BM are valid. The subset of interest is that of upward-closed binary multirelations [15, 26]. The following healthiness condition [14] characterises it.

Definition 20 (BMH)

$$\begin{aligned}&\mathbf{BMH}\\ &\widehat{=}\\ &\forall\, s : \mathit{State};\ ss_0, ss_1 : \mathbb{P}\,\mathit{State} \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss_1) \Rightarrow (s, ss_1) \in B\end{aligned}$$

If a particular initial state s is related to a set of final states ss_0, then it is also related to any superset of ss_0. This means that if it is possible to terminate in some final state that is in ss_0, then the addition of any other final states to that same set does not change the final states available for angelic choice, which correspond to those in the distributed intersection of all sets of final states available for demonic choice.

The set of binary multirelations of interest can alternatively be characterised by the fixed points of the following function.

Definition 21 ($\mathbf{bmh_{upclosed}}$)

$$\begin{array}{l}\mathbf{bmh_{upclosed}}(B)\\ \widehat{=}\\ \{s : State, ss : \mathbb{P}\, State \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss\}\end{array}$$

This equivalence is established by the following Law 3.2.1.

Law 3.2.1 ($\mathbf{bmh_{upclosed}}$-BMH)

$$\mathbf{BMH} \Leftrightarrow \mathbf{bmh_{upclosed}}(B) = B$$

Proof.

$$\begin{array}{l}
\mathbf{BMH} \hfill \{\text{Definition of } \mathbf{BMH}\}\\
\Leftrightarrow \forall\, s : State;\ ss_0, ss_1 : \mathbb{P}\, State \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss_1) \Rightarrow (s, ss_1) \in B\\
\hfill \{\text{Predicate calculus: quantifier scope}\}\\
\Leftrightarrow \left(\begin{array}{l}\forall\, s : State;\ ss_1 : \mathbb{P}\, State \bullet\\ (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss_1) \Rightarrow (s, ss_1) \in B\end{array}\right)\\
\hfill \{\text{Property of sets: subset inclusion}\}\\
\Leftrightarrow \{s : State, ss : \mathbb{P}\, State \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss\} \subseteq B\\
\hfill \{\text{Property of sets: subset inclusion}\}\\
\Leftrightarrow \{s : State, ss : \mathbb{P}\, State \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss\} \cup B = B\\
\hfill \{\text{Property of sets: set union}\}\\
\Leftrightarrow \left\{\ s : State, ss : \mathbb{P}\, State \left|\ \begin{array}{l}(\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss)\\ \lor\\ (s, ss) \in B\end{array}\right.\right\} = B\\
\hfill \{\text{Predicate calculus: instantiation of existential quantifier for } ss_0 = ss\}\\
\Leftrightarrow \{s : State, ss : \mathbb{P}\, State \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss\} = B\\
\hfill \{\text{Definition of } \mathbf{bmh_{upclosed}}\}\\
\Leftrightarrow \mathbf{bmh_{upclosed}}(B)
\end{array}$$

$\square$

The set of fixed points can be used interchangeably with the healthiness condition as it characterises exactly the upward-closed binary multirelations.

3.3 Refinement ordering

The refinement order for healthy binary multirelations B_0 and B_1, as presented in [14] is reproduced below.

Definition 22 ($\sqsubseteq_{BM}$)

$$B_0 \sqsubseteq_{BM} B_1 \mathrel{\widehat=} B_0 \supseteq B_1$$

It is defined as subset inclusion, similarly to the refinement order for set-based relations [14]. This partial order over BM forms a lattice. It allows an increase in the degree of angelic nondeterminism and a decrease in demonic nondeterminism. This aspect is discussed further in Section 3.4.

In what follows we define the extreme points of the lattice as given by the subset ordering. These correspond respectively to the notions of a miraculous program, as defined by $\top_{BM}$, and abort, as defined by $\bot_{BM}$.

Definition 23 (Miracle)

$$\top_{BM} \mathrel{\widehat=} \emptyset$$

Definition 24 (Abort)

$$\bot_{BM} \mathrel{\widehat=} State \times \mathbb{P}\, State$$

The top of the lattice $\top_{BM}$ is defined as the empty set while the bottom $\bot_{BM}$ is defined as the universal relation. The consequence is that a miraculous program cannot be executed, while abort exhibits arbitrary behaviour for every possible initial state. This allows us to establish the following law.

Law 3.3.1 (Refinement)

$$\bot_{BM} \sqsubseteq_{BM} B \sqsubseteq_{BM} \top_{BM}$$

Proof. Follows from the subset ordering. $\square$

Having presented the refinement ordering and its extreme points, in the following section we introduce the operators of the theory, including interesting properties regarding refinement.

3.4 Operators

In this section we present the main operators of the theory of binary multirelations [15] and discuss their properties.

3.4.1 Assignment

The assignment operator is defined as follows.

Definition 25

$$(x := e)_{BM} \mathrel{\widehat{=}} \{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto e) \in ss\}$$

It relates every initial state s to every possible set of final states ss, such that ss includes a state where s is overridden with a record where x has the value of the expression e.

3.4.2 Angelic choice

The angelic choice operator is defined as intersection.

Definition 26 **($\sqcup_{BM}$)**

$$B_0 \sqcup_{BM} B_1 \mathrel{\widehat{=}} B_0 \cap B_1$$

This operator corresponds to the least upper bound of the lattice. It captures the intuition that the final states available to the angel must be in the intersection of all choices available for demonic choice. Consequently, the operator observes the following law with respect to refinement.

Law 3.4.1

$$B_0 \sqsubseteq_{BM} (B_0 \sqcup_{BM} B_1)$$

Proof.

$$
\begin{aligned}
& B_0 \sqsubseteq_{BM} (B_0 \sqcup_{BM} B_1) && \{\text{Definition of } \sqsubseteq_{BM} \text{ and } \sqcup_{BM}\} \\
& = B_0 \supseteq (B_0 \cap B_1) && \{\text{Property of sets}\} \\
& = \mathit{true}
\end{aligned}
$$

□

As expected, this allows the degree of angelic nondeterminism to be increased. We observe that the proofs shown follow from the original model of [15]. Here we simply prove them as they provide auxiliary results for our discussion.

3.4.3 Demonic choice

The demonic choice operator is precisely defined as the dual of the angelic choice operator by considering set union.

Definition 27 ($\sqcap_{BM}$)

$$B_0 \sqcap_{BM} B_1 \mathrel{\widehat{=}} B_0 \cup B_1$$

The sets of final states available for demonic choice correspond to those in either B_0 or B_1. It corresponds to the greatest lower bound of the lattice. Therefore it observes the following law with respect to the refinement order.

Law 3.4.2

$$(B_0 \sqcap_{BM} B_1) \sqsubseteq_{BM} B_0$$

Proof.

$$
\begin{aligned}
& (B_0 \sqcap_{BM} B_1) \sqsubseteq_{BM} B_0 && \{\text{Definition of } \sqsubseteq_{BM} \text{ and } \sqcup_{BM}\} \\
& = (B_0 \cup B_1) \supseteq B_0 && \{\text{Property of sets}\} \\
& = \mathit{true}
\end{aligned}
$$

□

For an example, we consider the demonic choice over two assignments.

Example 6

$$(x := 1)_{BM} \sqcap_{BM} (x := 2)_{BM}$$
$$=$$
$$\{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto 1) \in ss \lor s \oplus (x \mapsto 2) \in ss\}$$

In this case, all initial states s are related to every set of final states ss that contains either a component where x is mapped to 1 or 2, or both. This means that it is impossible for the angel to enforce a particular choice, as the intersection of all sets of final states for a particular initial state, is empty.

The angelic and demonic choice operators distribute over one another.

Law 3.4.3

$$B_0 \sqcap_{BM} (B_1 \sqcup_{BM} B_2) = (B_0 \sqcap_{BM} B_1) \sqcup_{BM} (B_0 \sqcap_{BM} B_2)$$

Proof. Follows from the definition of $\sqcap_{BM}$, $\sqcup_{BM}$ and property of sets. □

This property follows from the distributive properties of set union and set intersection. This property is equally applicable in the theory of predicate transformers and the UTP model of [14].

3.4.4 Sequential composition

Sequential composition for binary multirelations [14, 15] is defined below.

Definition 28 (; $_{BM}$)

$$B_0 \;\;_{BM}\; B_1$$
$$\widehat{=}$$
$$\left\{ \begin{array}{l} s : State, ss_1 : \mathbb{P}\, State \\ \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B_0 \land ss_0 \subseteq \{s : State \mid (s, ss_1) \in B_1\} \end{array} \right\}$$

It is defined by considering every initial state s in B_0 and set of final states ss_1, such that there is some intermediate set of states ss_0 that is related from s in B_0, and ss_0 is a subset of the set of initial states in B_1 that achieve ss_1. As noted in [14] for healthy binary multirelations this definition can be simplified as shown in the following law.

Law 3.4.4 ($;_{BM}$-healthy-BM) *Provided* B_0 *is* **BMH**-*healthy.*

$$B_0 \;;_{BM}\; B_1 = \{s : State, ss : \mathbb{P}\, State \mid (s, \{s : State \mid (s, ss) \in B_1\}) \in B_0\}$$

Proof.

$$\begin{aligned}
&B_0 \;;_{BM}\; B_1 && \{\text{Definition of } ;_{BM}\}\\
&= \left\{\begin{array}{l} s : State, ss_1 : \mathbb{P}\, State \\ \mid \exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B_0 \land ss_0 \subseteq \{s : State \mid (s, ss_1) \in B_1\} \end{array}\right\} \\
& && \{\text{Assumption: } B_0 \text{ is } \textbf{BMH}\text{-healthy}\}\\
&= \{s : State, ss_1 : \mathbb{P}\, State \mid (s, \{s : State \mid (s, ss_1) \in B_1\}) \in B_0\}
\end{aligned}$$

□

This definition is used as the basis for the definition of sequential composition in the isomorphic UTP model of [14]. This is also the basis for our interpretation of the definition of sequential composition in the extended binary mutlirelational model that we present in Chapter 4.

3.5 Final considerations

In this chapter we have introduced the theory of binary multirelations. This model allows the specification of programs that have both angelic and demonic nondeterminism in a relational setting. It is known to be isomorphic to the predicate transformers model [14, 15].

In addition, the model is also isomorphic to the UTP model of [14], a theory of designs with angelic nondeterminism. However, these models can only consider final states that are necessarily terminating. This corresponds to designs with angelic nondeterminism that observe **H3**.

The binary multirelational theory presented, along with the isomorphic predicative UTP model of [14] provide the basis for developing an extended multirelational theory in the following Chapter 4.

Chapter 4

Binary multirelational model

In this chapter we introduce an extended binary multirelational model that can model sets of final states that are not necessarily terminating. This is achieved by extending the original model of [15], presented in the previous chapter, using an extra symbol that denotes the possibility for non-termination.

The following Section 4.1 introduces the model and formally defines the binary multirelations of interest. In Section 4.2 the healthiness conditions are defined. Their characterisation as fixed points is presented in Section 4.3. In Section 4.4 the refinement order is defined. The operators of the theory are explored in Section 4.5. Finally, Section 4.6 formalizes the relationship between this model and that of [15].

4.1 Introduction

Similar to the original model of binary multirelations, a relation in this model associates to each initial program state a set of final states. The notion of final state, however, is different, as formalised by the following type $BM_{\perp}$.

Definition 29

$$State_{\perp} == (State \cup \{\perp\})$$
$$BM_{\perp} == State \leftrightarrow \mathbb{P}\, State_{\perp}$$

Each initial state is related to a set of final states of type $State_{\perp}$, a final state that may include the symbol $\perp$. This symbol indicates that for a particular set of final states, the program may or may not terminate. If a set of final states does not contain $\perp$ then the program must terminate.

For example, consider the program that assigns the value 1 to the variable x but may or may not terminate. This is specified by the following relation, where $:=_{BM_\perp}$ is the assignment operator that does not require termination.

Example 7

$$(x :=_{BM_\perp} 1) = \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid s \oplus (x \mapsto 1) \in ss\}$$

Every initial state s is associated with a set of final states ss where the state obtained from s by overriding the value of the component x with 1 is included. Since ss is of type $\mathit{State}_\perp$, all sets of final states in ss include those with and without $\perp$.

It is also possible to specify a program that must terminate for certain sets of final states but not necessarily for others as shown in the following example, where $\sqcap_{BM_\perp}$ is the demonic choice operator of the theory.

Example 8

$$\begin{array}{l}(x :=_{BM} 1) \sqcap_{BM_\perp} (x :=_{BM_\perp} 2) \\ = \\ \left\{\begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \mid (s \oplus (x \mapsto 1) \in ss \land \perp \notin ss) \lor (s \oplus (x \mapsto 2) \in ss) \end{array}\right\}\end{array}$$

Since BM is in fact a subset of $BM_\perp$, it is possible to use some of the existing operators, such as the terminating assignment operator $:=_{BM}$. In this case, there is a demonic choice between the terminating assignment of 1 to x, and the assignment of 2 to x that does not require termination.

Similar to the original theory of binary multirelations, the set of final states encodes the choices available to the angel. The demonic choices are encoded by the different ways in which the set of final states can be chosen.

4.2 Healthiness conditions

In this section the healthiness conditions of the theory are introduced as predicates. Their characterisation as fixed points is developed in Section 4.3.

4.2.1 BMH0

The first healthiness condition of interest is **BMH0**. It enforces the upward closure of the original theory of binary multirelations [15] for sets of final states that are necessarily terminating, but in addition enforces a similar property for sets of final states that are not required to terminate.

Definition 30 (BMH0)

$$\begin{array}{l}\forall s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)) \Rightarrow (s, ss_1) \in B\end{array}$$

It states that for every initial state s, and for every set of final states ss_0 in a relation B, any superset ss_1 of that final set of states is also associated with s such that $\perp$ is in ss_0 if and only if it is in ss_1. That is, **BMH0** requires the upward closure for sets of final states that terminate, and for those that that may or may not terminate, but separately.

The definition of **BMH0** can actually be split into two conjunctions as shown in the following Law 4.2.1. **BMH** is the healthiness condition of the original theory and is defined in the previous Chapter 3.

Law 4.2.1

$$\begin{array}{l}\mathbf{BMH0} \\ \Leftrightarrow \\ \left(\begin{array}{l}\left(\begin{array}{l}\forall s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \Rightarrow (s, ss_1) \in B\end{array}\right) \\ \land \\ \mathbf{BMH}\end{array}\right)\end{array}$$

Proof.

$$\begin{array}{ll}\mathbf{BMH0} & \{\text{Definition of } \mathbf{BMH0}\} \\ \Leftrightarrow \left(\begin{array}{l}\forall s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)) \Rightarrow (s, ss_1) \in B\end{array}\right) & \\ & \{\text{Propositional calculus}\} \\ \Leftrightarrow \left(\begin{array}{l}\forall s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l}(s, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land ((\perp \in ss_0 \land \perp \in ss_1) \lor (\perp \notin ss_1 \land \perp \notin ss_0))\end{array}\right) \Rightarrow (s, ss_1) \in B\end{array}\right) & \end{array}$$

$$\{\text{Propositional calculus}\}$$

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \Rightarrow (s, ss_1) \in B \right) \\ \land \\ \left(((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \notin ss_0 \land \perp \notin ss_1) \Rightarrow (s, ss_1) \in B \right) \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \Rightarrow (s, ss_1) \in B \end{array} \right) \\ \land \\ \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \notin ss_0 \land \perp \notin ss_1) \Rightarrow (s, ss_1) \in B \end{array} \right) \end{array} \right)$$

$$\{\text{Predicate calculus: type restriction}\}$$

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \Rightarrow (s, ss_1) \in B \end{array} \right) \\ \land \\ \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1) \Rightarrow (s, ss_1) \in B \end{array} \right) \end{array} \right)$$

$$\{\text{Definition of } \mathbf{BMH} \text{ (Definition 20)}\}$$

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \Rightarrow (s, ss_1) \in B \end{array} \right) \\ \land \\ \mathbf{BMH} \end{array} \right)$$

□

This result confirms that for sets of final states that terminate this healthiness condition enforces **BMH** exactly as in the original theory of binary multirelations [15]. This ensures that if it is possible to terminate in some final state, then termination is also guaranteed in any superset.

4.2.2 BMH1

The second healthiness condition **BMH1** requires that if it possible to choose a set of final states where termination is not guaranteed, then it must also be possible to choose an equivalent set of states where termination is guaranteed. This healthiness condition is similar in nature to **H2** in the theory of designs.

Definition 31 (BMH1)

$$\forall s : State;\ ss : \mathbb{P}\, State_\perp \bullet (s, ss \cup \{\perp\}) \in B \Rightarrow (s, ss) \in B$$

If it is possible to reach a set of final states ($ss \cup \{\perp\}$) from some initial state s, where termination is not required, then the set of final states ss, possibly without $\perp$, so that termination is required is also associated with s.

This healthiness condition excludes relations that only offer sets of final states that may not terminate. Consider the following example.

Example 9

$$\{s : State, ss : \mathbb{P}\, State_\perp \mid (x \mapsto 1) \in ss \land \perp \in ss\}$$

This relation describes an assignment to the only program variable x where termination is not guaranteed. However, it discards the inclusive situation where termination may indeed occur. The inclusion of an equivalent final set of states that requires termination does not change the choices available to the angel as it is still impossible to guarantee termination.

The definition of **BMH1** can be stated in a slightly different way by strengthening the antecedent as shown in the following Lemma 4.2.1.

Lemma 4.2.1

$$\begin{array}{l} \mathbf{BMH1} \\ \Leftrightarrow \\ \forall s : State, ss : \mathbb{P}\, State_\perp \bullet (s, ss \cup \{\perp\}) \in B \land \perp \notin ss \Rightarrow (s, ss) \in B \end{array}$$

Proof.

$$\begin{array}{ll}
\mathbf{BMH1} & \{\text{Definition of } \mathbf{BMH1}\} \\
\Leftrightarrow \forall s : State, ss : \mathbb{P}\, State_\perp \bullet (s, ss \cup \{\perp\}) \in B \Rightarrow (s, ss) \in B & \\
& \{\text{Predicate calculus}\} \\
\Leftrightarrow \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} (s, ss \cup \{\perp\}) \in B \land (\perp \in ss \lor \perp \notin ss)) \\ \Rightarrow \\ (s, ss) \in B \end{array}\right) & \\
& \{\text{Predicate calculus}\} \\
\Leftrightarrow \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss \cup \{\perp\}) \in B \land \perp \in ss) \Rightarrow (s, ss) \in B \\ \land \\ ((s, ss \cup \{\perp\}) \in B \land \perp \notin ss) \Rightarrow (s, ss) \in B \end{array}\right) & \\
& \{\text{Property of sets (Lemma B.3.5)}\}
\end{array}$$

$$\Leftrightarrow \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss) \in B \land \perp \in ss) \Rightarrow (s, ss) \in B \\ \land \\ ((s, ss \cup \{\perp\}) \in B \land \perp \notin ss) \Rightarrow (s, ss) \in B \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \forall s : State, ss : \mathbb{P}\, State_\perp \bullet ((s, ss \cup \{\perp\}) \in B \land \perp \notin ss) \Rightarrow (s, ss) \in B$$

□

This property could alternatively be restated by restricting the type of ss to $\mathbb{P}\, State$. This concludes our discussion regarding **BMH1**.

4.2.3 BMH2

The third healthiness condition captures a redundancy in the model, namely that a set of final states defined by either the empty set or the set $\{\perp\}$ characterises abortion.

Definition 32 (BMH2)

$$\forall s : State \bullet (s, \emptyset) \in B \Leftrightarrow (s, \{\perp\}) \in B$$

Therefore we require that for all initial states s, it is related to the empty set of final states if, and only if, it is also related to the set of final states $\{\perp\}$.

If we consider **BMH1** in isolation, it covers the reverse implication of **BMH2** because if $(s, \{\perp\})$ is in the relation, so is $(s, \emptyset)$. However, the implication of **BMH2** is stronger than **BMH1** by requiring $(s, \{\perp\})$ to be in the relation if $(s, \emptyset)$ is in the relation.

The reason for letting this redundancy persist in the model is to keep it as similar as possible to the original model of binary multirelations. This is of particular interest as it helps with linking these models.

4.2.4 BMH3

The fourth healthiness condition characterises a subset of the model, of type $BM_\perp$, that corresponds to the original theory of binary multirelations.

Definition 33 (BMH3)

$$\forall s : State \bullet \left(\begin{array}{l} ((s, \emptyset) \notin B) \\ \Rightarrow \\ (\forall ss : \mathbb{P}\, State_\perp \bullet (s, ss) \in B \Rightarrow \perp \notin ss) \end{array}\right)$$

If an initial state s is not related to the empty set, then it must also be the case that for all sets of final states ss related to s, $\perp$ is not included in the set of final states ss.

This healthiness condition excludes relations that do not guarantee termination for particular initial states, yet establish some set of final states. Example 7 is an instance of such a relation. This is also the case for the original theory of binary multirelations. If it is possible for a program not to terminate when started from some initial state, then execution from that state must lead to arbitrary behaviour. This is the same intuition behind **H3** in the theory of designs.

It is precisely the restriction imposed by **BMH3** that we avoid with the binary multirelational model proposed. However, in order to study its relationship with the existing models the subset of **BMH3**-healthy relations is of interest.

4.3 Healthiness conditions as fixed points

In this section we specify functions whose fixed points characterise the new model of binary multirelations. We also specify functions that characterise the subset corresponding to the original model of [15]. This characterisation allows, for example, to prove that the healthiness conditions are idempotent.

In Sections 4.3.1 to 4.3.4, each healthiness condition is characterised by a corresponding function. The systematic exploration of the properties of the functional composition of each function is deferred to Appendices B.1 to B.2. Finally, in Sections 4.3.5 and 4.3.6 the two functions that characterise the model as a whole, and its subset of interest, are presented. Furthermore, we prove that the fixed points correspond exactly to the relations satisfied by the predicative healthiness conditions defined earlier.

In general, for each healthiness condition of interest, we use the notation $\mathbf{bmh_x}$ to denote the function whose fixed points correspond exactly to the relations characterised by the healthiness condition **BMHx**.

$$\mathbf{bmh_x}(B) = B \Leftrightarrow \mathbf{BMHx}$$

Furthermore, the notation $\mathbf{bmh_{x,y}}$ denotes the functional composition of the respective functions:

$$\mathbf{bmh_{x,y}}(B) = \mathbf{bmh_x} \circ \mathbf{bmh_y}(B)$$

This concludes the discussion of the notation used in the following sections.

4.3.1 $\mathbf{bmh_0}$

The first function of interest is $\mathbf{bmh_0}$ whose fixed points are the **BMH0**-healthy binary multirelations.

Definition 34 ($\mathbf{bmh_0}$)

$$\mathbf{bmh_0}(B) = \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

This definition is justified by the following Lemma 4.3.1.

Lemma 4.3.1 (**BMH0-iff-**$\mathbf{bmh_0}$)

$$\mathbf{BMH0} \Leftrightarrow \mathbf{bmh_0}(B) = B$$

Proof.

$$\begin{aligned}
&\mathbf{BMH0} && \{\text{Definition of } \mathbf{BMH0}\}\\
&\Leftrightarrow \left(\begin{array}{l} \forall\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)) \Rightarrow (s_0, ss_1) \in B \end{array} \right) \\
& && \{\text{Predicate calculus: quantifier scope}\}\\
&\Leftrightarrow \left(\begin{array}{l} \forall\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right) \Rightarrow (s_0, ss_1) \in B \end{array} \right) \\
& && \{\text{Property of sets: subset inclusion}\}\\
&\Leftrightarrow \left\{ \begin{array}{l} s_0 : State, ss_1 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right. \end{array} \right\} \subseteq B \\
& && \{\text{Property of sets}\}\\
&\Leftrightarrow \left(\left\{ \begin{array}{l} s_0 : State, ss_1 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right. \end{array} \right\} \cup B \right) = B \\
& && \{\text{Property of sets}\}\\
&\Leftrightarrow \left(\left\{ \begin{array}{l} s_0 : State, ss_1 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right) \\ \lor (s_0, ss_1) \in B \end{array} \right. \end{array} \right\} \right) = B \\
& && \{\text{Instantiation of existential quantifier for } ss_0 = ss_1\}
\end{aligned}$$

$$\Leftrightarrow \left(\left\{ \begin{array}{l} s_0 : State, ss_1 : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land\, (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right\}\right) = B$$

$$\{\text{Definition of } \mathbf{bmh_0}\}$$

$$\Leftrightarrow \mathbf{bmh_0}(B) = B$$

□

When healthiness conditions are expressed as fixed points of a function it is essential that they are idempotent [1]. This is established for each of the functions **bmh** in Appendix B.1. In the case of $\mathbf{bmh_0}$ this is established by the Lemma B.1.1.

4.3.2 $\mathbf{bmh_1}$

In this section the function $\mathbf{bmh_1}$ that characterises **BMH1**-healthy relations is presented.

Definition 35 ($\mathbf{bmh_1}$)

$$\mathbf{bmh_1}(B) = \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\}$$

The function returns all pairs (s, ss) in B, such that if a set of final states includes $\perp$ then there is also a set of final states without $\perp$. Its relationship with **BMH1** is justified by the following Lemma 4.3.2.

Lemma 4.3.2 (BMH1-iff-$\mathbf{bmh_1}$)

$$\mathbf{BMH1} \Leftrightarrow \mathbf{bmh_1}(B) = B$$

Proof.

BMH1 $\{\text{Definition of } \mathbf{BMH1}\}$

$$\Leftrightarrow \forall\, s : State;\ ss : \mathbb{P}\, State_\perp \bullet (s, ss \cup \{\perp\}) \in B \Rightarrow (s, ss) \in B$$

$$\{\text{Property of sets and definition of subset inclusion}\}$$

$$\Leftrightarrow \{s : State;\ ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B\} \subseteq B \qquad \{\text{Property of sets}\}$$

$$\Leftrightarrow (\{s : State;\ ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B\} \cup B) = B$$

$$\{\text{Property of sets}\}$$

$$\Leftrightarrow (\{s : State;\ ss : \mathbb{P}\, State_{\perp} \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\}) = B$$

$$\{\text{Definition of } \mathbf{bmh_1}\}$$

$$\Leftrightarrow \mathbf{bmh_1}(B) = B$$

□

Lemma B.1.2 establishes that $\mathbf{bmh_1}$ is idempotent. This concludes our discussion regarding the definition of $\mathbf{bmh_1}$.

4.3.3 $\mathbf{bmh_2}$

The healthiness condition **BMH2** is characterised by the function $\mathbf{bmh_2}$.

Definition 36

$$\mathbf{bmh_2}(B) \mathrel{\widehat{=}} \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\}$$

The definition considers every pair (s, ss) in B and requires that $(s, \{\perp\})$ is in B if and only if $(s, \emptyset)$ is also in B. If the equivalence is not satisfied then $\mathbf{bmh_2}$ yields the empty set. This definition is justified by the following Lemma 4.3.3.

Lemma 4.3.3 (BMH2-iff-$\mathbf{bmh_2}$)

$$\mathbf{BMH2} \Leftrightarrow \mathbf{bmh_2}(B) = B$$

Proof.

$$\mathbf{BMH2} \qquad \{\text{Definition of } \mathbf{BMH2}\}$$

$$\Leftrightarrow \forall s : State \bullet (s, \emptyset) \in B \Leftrightarrow (s, \{\perp\}) \in B \qquad \{\text{Predicate calculus}\}$$

$$\Leftrightarrow \forall s : State \bullet \left(\begin{array}{l} (s, \emptyset) \in B \Rightarrow (s, \{\perp\}) \in B \\ \land \\ (s, \{\perp\}) \in B \Rightarrow (s, \emptyset) \in B \end{array} \right) \qquad \{\text{Predicate calculus}\}$$

$$\Leftrightarrow \forall s : State \bullet \left(\begin{array}{l} (\exists ss_0 : \mathbb{P}\, State_{\perp} \bullet (s, \emptyset) \in B \land (s, ss_0) \in B) \Rightarrow (s, \{\perp\}) \in B \\ \land \\ (\exists ss_0 : \mathbb{P}\, State_{\perp} \bullet (s, \{\perp\}) \in B \land (s, ss_0) \in B) \Rightarrow (s, \emptyset) \in B \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$\Leftrightarrow \forall s : State, ss_0 : \mathbb{P}\, State_{\perp} \bullet \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, ss_0) \in B) \Rightarrow (s, \{\perp\}) \in B \\ \land \\ ((s, \{\perp\}) \in B \land (s, ss_0) \in B) \Rightarrow (s, \emptyset) \in B \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$\Leftrightarrow \forall s : State, ss_0 : \mathbb{P}\, State_\perp \bullet \begin{pmatrix} (s, ss_0) \in B \Rightarrow ((s, \{\perp\}) \in B \lor (s, \emptyset) \notin B) \\ \land \\ (s, ss_0) \in B \Rightarrow ((s, \emptyset) \in B \lor (s, \{\perp\}) \notin B) \end{pmatrix}$$

{Predicate calculus}

$$\Leftrightarrow \forall s : State, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \Rightarrow \begin{pmatrix} (s, \{\perp\}) \in B \lor (s, \emptyset) \notin B) \\ \land \\ ((s, \emptyset) \in B \lor (s, \{\perp\}) \notin B) \end{pmatrix}$$

{Predicate calculus}

$$\Leftrightarrow \forall s : State, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \Rightarrow ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)$$

{Property of sets}

$$\Leftrightarrow B \subseteq \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B\}$$

{Property of sets}

$$\Leftrightarrow B = (B \cap \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B\})$$

{Property of sets}

$$\Leftrightarrow B = \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)\}$$

{Definition of $\mathbf{bmh_2}$}

$$\Leftrightarrow B = \mathbf{bmh_2}(B)$$

□

Similarly, Lemma B.1.3 establishes that $\mathbf{bmh_2}$ is an idempotent function.

4.3.4 $\mathbf{bmh_3}$

This section introduces the definition of $\mathbf{bmh_3}$ whose fixed points are **BMH3**-healthy relations.

Definition 37

$$\begin{aligned} &\mathbf{bmh_3}(B) \\ &\widehat{=} \\ &\{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B\} \end{aligned}$$

The definition considers every pair (s, ss) in B and requires that either ss is a set of final states with guaranteed termination or $(s, \emptyset)$ is in B, and thus the initial state s leads to arbitrary behaviour. This is justified by the following Law 4.3.1.

Law 4.3.1 (BMH3-$\mathbf{bmh_3}$)

$$\mathbf{BMH3} \Leftrightarrow \mathbf{bmh_3}(B) = B$$

Proof.

$$\begin{aligned}
&\mathbf{BMH3} && \{\text{Definition of } \mathbf{BMH3}\}\\
&\Leftrightarrow \forall s : \mathit{State} \bullet ((s, \emptyset) \notin B) \Rightarrow (\forall ss : \mathbb{P}\,\mathit{State}_\perp \bullet (s, ss) \in B \Rightarrow \perp \notin ss)\\
& && \{\text{Predicate calculus}\}\\
&\Leftrightarrow \forall s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \bullet ((s, \emptyset) \notin B) \Rightarrow ((s, ss) \in B \Rightarrow \perp \notin ss)\\
& && \{\text{Predicate calculus}\}\\
&\Leftrightarrow \forall s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \bullet ((s, ss) \in B \land \perp \in ss) \Rightarrow (s, \emptyset) \in B\\
& && \{\text{Predicate calculus}\}\\
&\Leftrightarrow \forall s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \bullet (s, ss) \in B \Rightarrow ((s, \emptyset) \in B \lor \perp \notin ss)\\
& && \{\text{Property of sets and subset inclusion}\}\\
&\Leftrightarrow B \subseteq \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid ((s, \emptyset) \in B \lor \perp \notin ss)\}\\
& && \{\text{Property of sets}\}\\
&\Leftrightarrow B = (B \cap \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid ((s, \emptyset) \in B \lor \perp \notin ss)\})\\
& && \{\text{Property of sets}\}\\
&\Leftrightarrow B = \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B\}\\
& && \{\text{Definition of } \mathbf{bmh_3}\}\\
&\Leftrightarrow B = \mathbf{bmh_3}(B)
\end{aligned}$$

□

Finally, Lemma B.1.4 establishes that $\mathbf{bmh_3}$ is an idempotent function.

This section concludes our discussion regarding the definition of the $\mathbf{bmh_x}$ functions. Their functional composition is studied in detail in Appendix B.1. In the following sections we focus our attention only on the functional compositions that characterise the theory and its subset of interest.

4.3.5 BMH0-BMH2 as a fixed point ($\mathbf{bmh_{0,1,2}}$)

The relations in the theory are characterised by the conjunction of the healthiness conditions **BMH0-BMH2**. These relations can also be characterised as fixed points of the function $\mathbf{bmh_{0,1,2}}$ as defined below.

Definition 38

$$\mathbf{bmh_{0,1,2}}(B) \mathrel{\widehat=} \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right. \end{array}\right\}$$

This definition is justified by the functional composition of the respective **bmh** functions as shown in the following Lemma 4.3.4.

Lemma 4.3.4

$$\begin{array}{l} \mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_2}(B) \\ = \\ \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right. \end{array}\right\} \end{array}$$

Proof.

$$\mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_2}(B) \qquad \{\text{Definition of } \mathbf{bmh_0} \circ \mathbf{bmh_1}\}$$

$$= \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in \mathbf{bmh_2}(B) \lor (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_2}(B)) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array}\right. \end{array}\right\}$$

$$\{\text{Definition of } \mathbf{bmh_2}\}$$

$$= \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} \exists\, ss_0 : State_\perp \bullet \\ \left(\begin{array}{l} (s, ss_0) \in \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array}\right\} \\ \lor \\ (s, ss_0 \cup \{\perp\}) \in \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array}\right\} \end{array}\right) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array}\right. \end{array}\right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \\ \vee \\ ((s, ss_0 \cup \{\perp\}) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \end{array} \right) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\ \wedge\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

□

In the following Lemma 4.3.5 we prove that $\mathbf{bmh_{0,1,2}}$ is an idempotent function.

Lemma 4.3.5 ($\mathbf{bmh_{0,1,2}}$-idempotent)

$$\mathbf{bmh_{0,1,2}} \circ \mathbf{bmh_{0,1,2}}(B) = \mathbf{bmh_{0,1,2}}(B)$$

Proof.

$\mathbf{bmh_{0,1,2}} \circ \mathbf{bmh_{0,1,2}}(B)$ {Definition of $\mathbf{bmh_{0,1,2}}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in \mathbf{bmh_{0,1,2}}(B) \vee (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B)) \\ \wedge\, ((s, \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) \Leftrightarrow (s, \emptyset) \in \mathbf{bmh_{0,1,2}}(B)) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Law B.2.6, Law B.2.5 and predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in \mathbf{bmh_{0,1,2}}(B) \vee (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B)) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in \mathbf{bmh_{0,1,2}}(B) \wedge ss_0 \subseteq ss \\ \wedge\, (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) \wedge ss_0 \subseteq ss \\ \wedge\, (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Law B.2.4}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, ss_1 \bullet \left(\begin{array}{l} ((s, ss_1) \in B \lor (s, ss_1 \cup \{\perp\}) \in B) \\ \land\, ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right) \\ \lor \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, ss_1 \bullet \left(\begin{array}{l} ((s, ss_1) \in B \lor (s, ss_1 \cup \{\perp\} \cup \{\perp\}) \in B) \\ \land\, ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

$$
\{\text{Property of sets and predicate calculus}\}
$$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \exists\, ss_1 \bullet ((s, ss_1) \in B \lor (s, ss_1 \cup \{\perp\}) \in B) \land ss_1 \subseteq ss \\ \land\, (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

$$
\{\text{Definition of } \mathbf{bmh_{0,1,2}}\}
$$

$$
= \mathbf{bmh_{0,1,2}}(B)
$$

□

The particular order of the functional composition is justified by Theorem 4.3.1.

Theorem 4.3.1

$$
\mathbf{BMH0} \land \mathbf{BMH1} \land \mathbf{BMH2} \Leftrightarrow \mathbf{bmh_{0,1,2}}(B) = B
$$

Proof. Follows from Lemmas 4.3.6 to 4.3.8 and Lemma 4.3.9. □

This theorem, together with the respective lemmas enumerated in the following paragraphs, establishes that $\mathbf{bmh_{0,1,2}}$ is a suitable function for characterising **BMH0-BMH2**-healthy relations. Appendix B.1 provides some reasoning as to why other orders of application are not desirable. For example, not all functions are necessarily commutative.

From $\mathbf{bmh_{0,1,2}}$ to BMH0-BMH2

In the following laws we prove that the fixed points of $\mathbf{bmh_{0,1,2}}$ satisfy each of the predicative healthiness conditions **BMH0**, **BMH1** and **BMH2**.

Lemma 4.3.6

$$(\mathbf{bmh_{0,1,2}}(B) = B) \Rightarrow \mathbf{BMH0}$$

Proof.

$$
\begin{aligned}
&\mathbf{BMH0} && \{\text{Definition of } \mathbf{BMH0}\}\\
&= \left(\begin{array}{l} \forall\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)) \Rightarrow (s_0, ss_1) \in B \end{array} \right) && \{\text{Predicate calculus: quantifier scope}\}\\
&= \left(\begin{array}{l} \forall\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet \\ (\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)) \\ \Rightarrow \\ (s_0, ss_1) \in B \end{array} \right) && \{\text{Assumption: } \mathbf{bmh_{0,1,2}}(B) = B\}\\
&= \left(\begin{array}{l} \forall\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} (s_0, ss_0) \in \mathbf{bmh_{0,1,2}}(B) \land ss_0 \subseteq ss_1 \\ \land\, (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right) \\ \Rightarrow \\ (s_0, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right) && \{\text{Law B.2.4}\}\\
&= \left(\begin{array}{l} \forall\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right) \end{array} \right) \\ \Rightarrow \\ (s_0, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right) && \{\text{Law B.2.3}\}\\
&= \left(\begin{array}{l} \forall\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet \\ (s_0, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \Rightarrow (s_0, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right) && \{\text{Predicate calculus}\}\\
&= true
\end{aligned}
$$

□

Lemma 4.3.7

$$(\mathbf{bmh_{0,1,2}}(B) = B) \Rightarrow \mathbf{BMH1}$$

Proof.

BMH1 $\{\text{Lemma 4.2.1}\}$

$$= \forall s : State, ss : \mathbb{P}\, State_\perp \bullet ((s, ss \cup \{\perp\}) \in B \land \perp \notin ss) \Rightarrow (s, ss) \in B$$

$\{\text{Assumption: } \mathbf{bmh_{0,1,2}}(B) = B\}$

$$= \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\ ((s, ss \cup \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) \land \perp \notin ss) \Rightarrow (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array}\right)$$

$\{\text{Law B.2.3}\}$

$$= \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ss_0 \subseteq (ss \cup \{\perp\}) \land (\perp \in ss_0 \Leftrightarrow \perp \in (ss \cup \{\perp\})) \end{array}\right) \end{array}\right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array}\right)$$

$\{\text{Property of sets}\}$

$$= \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ss_0 \subseteq (ss \cup \{\perp\}) \land \perp \in ss_0 \end{array}\right) \end{array}\right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array}\right)$$

$\{\text{Predicate calculus and property of sets}\}$

$$= \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land (ss_0 \setminus \{\perp\}) \subseteq ss \land \perp \in ss_0 \end{array}\right) \end{array}\right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array}\right)$$

$\{\text{Introduce fresh variable}\}$

$$= \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists\, ss_0, t : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\ t \subseteq ss \land \perp \in ss_0 \\ \land\ t = (ss_0 \setminus \{\perp\}) \end{array} \right) \end{array} \right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right)$$

{Lemma B.3.2}

$$= \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists\, ss_0, t : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\ t \subseteq ss \land \perp \notin t \\ \land\ t \cup \{\perp\} = ss_0 \end{array} \right) \end{array} \right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right)$$

{One-point rule}

$$= \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land \perp \notin ss \\ \land \\ \exists\, t : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} ((s, t \cup \{\perp\}) \in B \lor (s, t \cup \{\perp\} \cup \{\perp\}) \in B) \\ \land\ t \subseteq ss \land \perp \notin t \end{array} \right) \end{array} \right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right)$$

{Property of sets and predicate calculus}

$$= \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, t : \mathbb{P}\, State_\perp \bullet (s, t \cup \{\perp\}) \in B \land t \subseteq ss \land \perp \notin t \land \perp \notin ss \end{array} \right) \\ \Rightarrow \\ (s, ss) \in \mathbf{bmh_{0,1,2}}(B) \end{array} \right)$$

{Law B.2.3}

$$
= \left(\begin{array}{l}
\forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\exists t : \mathbb{P}\, State_\perp \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss \wedge \perp \notin t \wedge \perp \notin ss
\end{array}\right) \\
\Rightarrow \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\exists ss_0 : State_\perp \bullet \left(\begin{array}{l}
((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\
\wedge\ ss_0 \subseteq ss_1 \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)
\end{array}\right)
\end{array}\right)
\end{array}\right)
$$

{Predicate calculus}

$$
= \left(\begin{array}{l}
\forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\exists t : \mathbb{P}\, State_\perp \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss \wedge \perp \notin t \wedge \perp \notin ss
\end{array}\right) \\
\Rightarrow \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\left(\begin{array}{l}
\exists ss_0 : State_\perp \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss_1 \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \\
\vee \\
\exists ss_0 : State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ss_1 \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss_1)
\end{array}\right)
\end{array}\right)
\end{array}\right)
$$

{Predicate calculus}

$$
= \left(\begin{array}{l}
\forall s : State, ss : \mathbb{P}\, State_\perp \bullet \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\exists t : \mathbb{P}\, State_\perp \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss \wedge \perp \notin t \wedge \perp \notin ss
\end{array}\right) \\
\Rightarrow \\
\left(\begin{array}{l}
((s,\{\perp\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\
\wedge \\
\left(\begin{array}{l}
\exists ss_0 : State_\perp \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss_1 \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \\
\vee \\
\exists ss_0 : State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ss_1 \wedge \perp \in ss_0 \wedge \perp \in ss_1 \\
\vee \\
\exists ss_0 : State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ss_1 \wedge \perp \notin ss_0 \wedge \perp \notin ss_1
\end{array}\right)
\end{array}\right)
\end{array}\right)
$$

{Variable renaming and predicate calculus}

$= true$

□

Lemma 4.3.8

$$(\mathbf{bmh_{0,1,2}}(B) = B) \Rightarrow \mathbf{BMH2}$$

Proof.

$$
\begin{aligned}
&\mathbf{BMH2} && \{\text{Definition of } \mathbf{BMH2}\}\\
&= \forall\, s : \mathit{State} \bullet (s, \emptyset) \in B \Leftrightarrow (s, \{\perp\}) \in B && \{\text{Assumption: } \mathbf{bmh_{0,1,2}}(B) = B\}\\
&= \forall\, s : \mathit{State} \bullet (s, \emptyset) \in \mathbf{bmh_{0,1,2}}(B) \Leftrightarrow (s, \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) && \{\text{Law B.2.5 and Law B.2.6}\}\\
&= \forall\, s : \mathit{State} \bullet ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \Leftrightarrow ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) && \{\text{Predicate calculus}\}\\
&= \mathit{true}
\end{aligned}
$$

□

These laws confirm that a fixed point of $\mathbf{bmh_{0,1,2}}$ satisfies each of the predicative healthiness conditions **BMH0**-**BMH2**. In the following laws we prove the reverse implication of Theorem 4.3.1.

From BMH0-BMH2 to $\mathbf{bmh_{0,1,2}}$

A binary multirelation that is **BMH0**, **BMH1** and **BMH2**-healthy is a fixed point of $\mathbf{bmh_{0,1,2}}$.

Lemma 4.3.9 *Provided B is* **BMH0** – **BMH2**-*healthy.*

$$\mathbf{bmh_{0,1,2}}(B) = B$$

Proof.

$$
\begin{aligned}
&\mathbf{bmh_{0,1,2}}(B) = B && \{\text{Definition of } \mathbf{bmh_{0,1,2}}\}\\
&\Leftrightarrow \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\perp \\ \mid \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \quad \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \quad \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} = B && \{\text{Assumption: } B \text{ is } \mathbf{BMH2}\text{-healthy}\}
\end{aligned}
$$

$$\Leftrightarrow \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \mid \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} = B$$

{Assumption: B is **BMH1**-healthy and predicate calculus}

$$\Leftrightarrow \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet ((s, ss_0) \in B \lor ((s, ss_0 \cup \{\perp\}) \in B \land (s, ss_0) \in B) \\ \mid \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} = B$$

{Predicate calculus: absorption law}

$$\Leftrightarrow \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} = B$$

{Assumption: B is **BMH0**-healthy}

$$\Leftrightarrow \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \land (s, ss) \in B \end{array} \right\} = B$$

{Instantiation of existential quantifier for $ss_0 = ss$}

$$\Leftrightarrow \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \left(\begin{array}{l} (\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \\ \lor \\ (s, ss) \in B \end{array} \right) \\ \mid \land\, (s, ss) \in B \end{array} \right\} = B$$

{Predicate calculus: absorption law}

$\Leftrightarrow \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss) \in B\} = B$ {Property of sets}

$\Leftrightarrow true$

□

These proofs conclude our discussion of the healthiness conditions of the new theory of binary multirelations. These relations can be characterised either by the predicates **BMH0**-**BMH2** or as fixed points of $\mathbf{bmh_{0,1,2}}$. In the following section we focus our attention on the subset of the theory that is in addition **BMH3**-healthy.

4.3.6 BMH0-BMH3 as a fixed point ($\mathbf{bmh_{0,1,3,2}}$)

The relations that are **BMH0**, **BMH1**, **BMH2** and **BMH3**-healthy can be characterised as fixed points of the following function.

Definition 39

$$\mathbf{bmh_{0,1,3,2}}(B)$$
$$=$$
$$\left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array}\right) \end{array}\right. \end{array}\right\}$$

This definition is justified by the following Lemma 4.3.10.

Lemma 4.3.10

$$\mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_3} \circ \mathbf{bmh_2}(B)$$
$$=$$
$$\left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array}\right) \end{array}\right. \end{array}\right\}$$

Proof.

$$\mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_3} \circ \mathbf{bmh_2}(B) \qquad \{\text{Law B.2.8}\}$$
$$= \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left|\begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in \mathbf{bmh_2}(B) \lor (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_2}(B)) \\ \land \\ (s, \emptyset) \in \mathbf{bmh_2}(B) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in \mathbf{bmh_2}(B) \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array}\right. \end{array}\right\}$$
$$\{\text{Definition of } \mathbf{bmh_2}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in B \\ \wedge \\ ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right. \end{array} \right\} \\ \vee \\ (s, ss_0 \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in B \\ \wedge \\ ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right. \end{array} \right\} \end{array} \right) \\ \wedge \\ (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} \left(\begin{array}{l} ((s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \\ \vee \\ ((s, ss_0 \cup \{\perp\}) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \end{array} \right) \\ \wedge \\ ((s, \emptyset) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\ \wedge \\ (s, \emptyset) \in B \wedge (s, \{\perp\}) \in B \\ \wedge \\ ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} (s, \emptyset) \in B \wedge (s, \{\perp\}) \in B \\ \wedge \\ \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\ \wedge \\ ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right) \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Law B.2.9}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} (s, \emptyset) \in B \wedge (s, \{\perp\}) \in B \\ \wedge \\ \left(\begin{array}{l} (s, \{\perp\}) \in B \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\ \wedge \\ ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right) \end{array} \right) \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \in B \land (s, \emptyset) \in B \\ \land \\ \exists\, ss_0 \bullet \left((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \right) \end{array} \right) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

□

In Lemma B.1.13 we prove that $\mathbf{bmh_{0,1,3,2}}$ is idempotent. This also follows directly from idempotency of the respective functions $\mathbf{bmh_0}$-$\mathbf{bmh_3}$.

The following Theorem 4.3.2, together with the respective lemmas that we discuss in the following sections, establishes that the fixed points of $\mathbf{bmh_{0,1,3,2}}$ correspond to the conjunction of the predicative healthiness conditions **BMH0**-**BMH3**.

Theorem 4.3.2

$$\mathbf{BMH0} \land \mathbf{BMH1} \land \mathbf{BMH2} \land \mathbf{BMH3} \Leftrightarrow \mathbf{bmh_{0,1,3,2}}(B) = B$$

Proof. The implication follows from Lemma 4.3.11. While the reverse implication follows from the fact that $\mathbf{bmh_{0,1,3,2}}$ is a fixed point of $\mathbf{bmh}_{0,1,2}$

(Lemma B.1.14) and Lemmas 4.3.6 to 4.3.8 and Law 4.3.2. $\square$

In the following sections we prove the auxiliary results pertaining to Theorem 4.3.2. First, we consider the lemmas needed to prove the implication. This is followed by lemmas supporting the proof of the reverse implication.

From $\mathbf{bmh_{0,1,3,2}}$ to BMH0-BMH3

Since the model of **BMH0-BMH3** is a subset of the more general model of **BMH0-BMH2**, every fixed point of $\mathbf{bmh_{0,1,3,2}}$ is also a fixed point of $\mathbf{bmh_{0,1,2}}$. This result is established in Lemma B.1.14. Together with those results established in Section 4.3.5, this allows us to ascertain that any fixed point of $\mathbf{bmh_{0,1,3,2}}$ also satisfies **BMH0-BMH2**.

Finally, the following Law 4.3.2 establishes that every fixed point of $\mathbf{bmh_{0,1,3,2}}$ satisfies the predicative healthiness condition **BMH3**.

Law 4.3.2

$$(\mathbf{bmh_{0,1,3,2}}(B) = B) \Rightarrow \mathbf{BMH3}$$

Proof.

$$
\begin{aligned}
&\mathbf{BMH3} && \{\text{Definition of } \mathbf{BMH3}\}\\
&= \forall s_0 : State \bullet \left(\begin{array}{l} ((s_0, \emptyset) \notin B) \\ \Rightarrow \\ (\forall ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \Rightarrow \perp \notin ss_0) \end{array}\right) && \{\text{Predicate calculus}\}\\
&= \forall s_0 : State \bullet \left(\begin{array}{l} (\exists ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in B \land \perp \in ss_0) \\ \Rightarrow \\ ((s_0, \emptyset) \in B) \end{array}\right) && \{\text{Assumption: } \mathbf{bmh_{0,1,3,2}}(B) = B\}\\
&= \forall s_0 : State \bullet \left(\begin{array}{l} (\exists ss_0 : \mathbb{P}\, State_\perp \bullet (s_0, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \land \perp \in ss_0) \\ \Rightarrow \\ ((s_0, \emptyset) \in \mathbf{bmh_{0,1,3,2}}(B)) \end{array}\right) && \{\text{Law B.2.10 and Law B.2.13}\}
\end{aligned}
$$

$$= \forall s_0 : State \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists ss_0 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in B \land ss_0 \subseteq ss \\ \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right) \\ \land \\ \perp \in ss_0 \end{array} \right) \\ \Rightarrow \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \end{array} \right)$$

$\{$Predicate calculus$\}$

$$= \forall s_0 : State \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists ss_0 : \mathbb{P}\, State_\perp \bullet \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \in ss_0 \end{array} \right) \\ \Rightarrow \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \end{array} \right)$$

$\{$Case analysis on $ss_0\}$

$$= \forall s_0 : State \bullet \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \Rightarrow \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \end{array} \right) \qquad \{\text{Predicate calculus}\}$$

$= true$

$\square$

Having established the proof for the implication of Theorem 4.3.2, in the following section we focus on the reverse implication.

From BMH0-BMH3 to $\mathbf{bmh_{0,1,3,2}}$

Finally, the Lemma 4.3.11 establishes the proof with respect to the reverse implication of Theorem 4.3.2.

Lemma 4.3.11

$$\mathbf{BMH0} \land \mathbf{BMH1} \land \mathbf{BMH2} \land \mathbf{BMH3} \Rightarrow \mathbf{bmh_{0,1,3,2}}(B) = B$$

Proof.

$\mathbf{bmh_{0,1,3,2}}(B)$ $\{$Definition of $\mathbf{bmh_{0,1,3,2}}\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \big(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \big) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \lor ((s, \{\perp\}) \notin B \land (s, \emptyset) \notin B)) \\ \land \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ (\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \Leftrightarrow (s, \{\perp\}) \in B) \\ \land \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ (\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Assumption: B is **BMH2**-healthy}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ (\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Assumption: B is **BMH0**-healthy}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \\ \land \\ (s, ss) \in B \land \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: instatiation of existential quantifier for $ss_0 = ss$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0) \\ \lor \\ ((s, ss) \in B \land \perp \notin ss) \end{array} \right) \\ \land \\ ((s, ss) \in B \land \perp \notin ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ ((s, ss) \in B \land \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Assumption: B is **BMH2**-healthy}

$$= \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, \emptyset) \in B \lor ((s, ss) \in B \land \perp \notin ss)\}$$

{Assumption: B is **BMH0**, **BMH2** and **BMH3**-healthy and Law B.2.15}

$$= B$$

□

These results establish that there are suitable functions whose fixed points characterise the theories of interest. The more general theory, that can encode sets of final states where termination is not guaranteed is characterised by $\mathbf{bmh_{0,1,2}}$. The function $\mathbf{bmh_{0,1,3,2}}$ characterises the subset that corresponds to the original theory of binary multirelations. The relationship with the original theory of binary multirelations is explored in Section 4.6.

4.4 Refinement ordering

The refinement order for the new binary multirelation model is defined exactly as in the original theory of binary multirelations [15].

Definition 40 (Refinement)

$$B_1 \sqsubseteq_{BM_\perp} B_0 \mathrel{\widehat=} B_1 \supseteq B_0$$

It is defined as reverse subset inclusion, such that a program B_0 refines B_1 if and only if B_0 is a subset of B_1.

The extreme points of the theory follow from the subset ordering. As expected of a theory of designs, they are the everywhere miraculous program and abort. Their definition is presented below.

Definition 41 (Miracle)

$$\top_{BM_\perp} \mathrel{\widehat=} \emptyset$$

As in the original theory, miracle is denoted by the absence of any relationship between any input state and any set of final states, that is, the program cannot possibly be executed.

Definition 42 (Abort)

$$\bot_{BM_\perp} \mathrel{\widehat=} State \times \mathbb{P}\, State_\perp$$

On the other hand, abort is characterised by the universal relation similarly to the original theory [15], such that every initial state is related to every possible set of final states.

4.5 Operators

In this section the operators of the theory are defined. In Sections 4.5.1 to 4.5.3 the main operators are defined, namely, assignment, angelic choice and demonic choice. In Section 4.5.4 the definition of sequential composition in the new model is presented.

In Chapter 5 we establish that the operators defined here are in correspondence with those of the new theory of designs with angelic nondeterminism. There we prove that the operators are closed. Together with the respective isomorphism establish between the theories, these results are sufficient to establish closure of the operators under **BMH0-BMH2**. The proof of closure using only the assumptions of this model is left as future work.

4.5.1 Assignment

In the new model there is in fact the possibility to define two distinct assignment operators. The first one behaves exactly as in the original theory of binary multirelations ($x :=_{BM} e$). It specifies the assignment of the value of expression e to the program variable x; is guaranteed to terminate. This operator does not need to be redefined, since $BM \subseteq BM_\perp$.

The new operator that we define below, however, behaves rather differently, in that the sets of final states may or may not be terminating.

Definition 43

$$(x :=_{BM_\perp} e) \mathrel{\widehat=} \{s : State, ss : \mathbb{P}\, State_\perp \mid s \oplus (x \mapsto e) \in ss\}$$

This assignment guarantees that for every initial state s, there is some set of final states available for angelic choice where x has the value of expression e. However, termination is not guaranteed. While the angel can choose the final value of x it cannot possibly guarantee termination in this case.

4.5.2 Angelic choice

The definition of angelic choice is the same as in the original theory of binary multirelations.

Definition 44

$$B_0 \sqcup_{BM_\perp} B_1 \mathrel{\widehat=} B_0 \cap B_1$$

It is defined by set intersection, such that for every set of final states available for demonic choice in B_0 and B_1 when started from a particular initial state, only those that can be chosen both in B_0 and B_1 are available.

In the following paragraphs we explore some of the properties observed by the angelic choice operator.

Properties

An interesting property of angelic choice that is observed in this model is illustrated by the following Law 4.5.1. It considers the angelic choice between two assignments of the same value, yet only one is guaranteed to terminate.

Law 4.5.1

$$(x :=_{BM_\perp} e) \sqcup_{BM_\perp} (x :=_{BM} e) = (x :=_{BM} e)$$

Proof.

$$
\begin{aligned}
&(x :=_{BM_\perp} e) \sqcup_{BM_\perp} (x :=_{BM} e) && \{\text{Definition of } :=_{BM_\perp}, :=_{BM} \text{ and } \sqcup_{BM_\perp}\}\\
&= \left(\begin{array}{l} \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid s \oplus (x \mapsto e) \in ss\} \\ \cap \\ \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State} \mid s \oplus (x \mapsto e) \in ss\} \end{array} \right) && \{\text{Type: } \perp \notin \mathbb{P}\,\mathit{State}\}\\
&= \left(\begin{array}{l} \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid s \oplus (x \mapsto e) \in ss\} \\ \cap \\ \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid s \oplus (x \mapsto e) \in ss \land \perp \notin ss\} \end{array} \right)\\
& && \{\text{Property of sets and predicate calculus}\}\\
&= \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \mid s \oplus (x \mapsto e) \in ss \land \perp \notin ss\}\\
& && \{\text{Type: } \perp \notin \mathbb{P}\,\mathit{State}\}\\
&= \{s : \mathit{State}, ss : \mathbb{P}\,\mathit{State} \mid s \oplus (x \mapsto e) \in ss\} && \{\text{Definition of } :=_{BM}\}\\
&= (x :=_{BM} e)
\end{aligned}
$$

□

This result can be interpreted as follows: given an assignment which is guaranteed to terminate, adding an equivalent angelic choice which is potentially non-terminating does not in fact introduce any new choices. Termination can still be enforced.

In general, and as expected from the original model of binary multirelations, the angelic choice operator observes the following properties with respect to the extreme points.

Law 4.5.2

$$\top_{BM_\perp} \sqcup_{BM_\perp} B = \top_{BM_\perp}$$

Proof.

$$
\begin{aligned}
&\top_{BM_\perp} \sqcup_{BM_\perp} B && \{\text{Definition of } \top_{BM_\perp} \text{ and } \sqcup_{BM_\perp}\}\\
&= \emptyset \cap B && \{\text{Property of sets}\}\\
&= \emptyset && \{\text{Definition of } \top_{BM_\perp}\}\\
&= \top_{BM_\perp}
\end{aligned}
$$

□

The angelic choice between an everywhere miraculous program and any other program is still miraculous.

Law 4.5.3

$$\perp_{BM_\perp} \sqcup_{BM_\perp} B = B$$

Proof.

$$\begin{aligned}
&\perp_{BM_\perp} \sqcup_{BM_\perp} B && \{\text{Definition of } \perp_{BM_\perp} \text{ and } \sqcup_{BM_\perp}\}\\
&= (State \times \mathbb{P}\, State_\perp) \cap B && \{\text{Property of sets}\}\\
&= B
\end{aligned}$$

□

On the other hand, the angelic choice between abort and any other program B is the same as B. That is, the angel will avoid choosing an aborting program if possible.

4.5.3 Demonic choice

The demonic choice operator is defined by set union, exactly as in the original theory of binary multirelations.

Definition 45

$$B_0 \sqcap_{BM_\perp} B_1 \mathrel{\widehat{=}} B_0 \cup B_1$$

For every initial state, a corresponding set of final states available for demonic choice in either, or both, of B_0 and B_1, is included in the result.

In the following paragraphs we present some results regarding the demonic choice operator.

Properties

Similar to the angelic choice operator, there is a general result regarding the demonic choice over the two assignment operators, terminating and not necessarily terminating. This is shown in the following Law 4.5.4.

Law 4.5.4

$$(x :=_{BM}\ e) \sqcap_{BM_\perp} (x :=_{BM_\perp}\ e) = (x :=_{BM_\perp}\ e)$$

Proof.

$$
\begin{array}{ll}
(x :=_{BM}\ e) \sqcap_{BM_\perp} (x :=_{BM_\perp}\ e) & \{\text{Definition of } :=_{BM},\ :=_{BM_\perp} \text{ and } \sqcap_{BM_\perp}\} \\
= \left(\begin{array}{l} \{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto e) \in ss\} \\ \cup \\ \{s : State, ss : \mathbb{P}\, State_\perp \mid s \oplus (x \mapsto e) \in ss\} \end{array} \right) & \{\text{Type: } \perp \notin \mathbb{P}\, State\} \\
= \left(\begin{array}{l} \{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto e) \in ss \land \perp \notin ss\} \\ \cup \\ \{s : State, ss : \mathbb{P}\, State_\perp \mid s \oplus (x \mapsto e) \in ss\} \end{array} \right) & \\
 & \{\text{Property of sets}\} \\
= \{s : State, ss : \mathbb{P}\, State \mid (s \oplus (x \mapsto e) \in ss \land \perp \notin ss) \lor s \oplus (x \mapsto e) \in ss\} & \\
 & \{\text{Predicate calculus: absorption law}\} \\
= \{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto e) \in ss\} & \{\text{Definition of } :=_{BM_\perp}\} \\
= (x :=_{BM_\perp}\ e) &
\end{array}
$$

□

This result can be interpreted as follows: if there is an assignment for which termination is not guaranteed, then the demonic choice over this assignment and an equivalent one that is guaranteed to terminate is the same as the assignment that does not require termination. In other words, if it is possible for the demon to choose between two similar sets of final states, one that is possibly non-terminating and one that terminates, then the one for which termination is not guaranteed dominates the choice.

The following two laws show how the demonic choice operator behaves with respect to the extreme points of the theory.

Law 4.5.5

$$\perp_{BM_\perp} \sqcap_{BM_\perp} B = \perp_{BM_\perp}$$

Proof.

$$\perp_{BM_\perp} \sqcap_{BM_\perp} B \qquad \{\text{Definition of } \perp_{BM_\perp} \text{ and } \sqcap_{BM_\perp}\}$$

$= (State \times \mathbb{P}\, State_{\perp}) \cup B$ $\{$Property of sets$\}$
$= (State \times \mathbb{P}\, State_{\perp})$ $\{$Definition of $\perp_{BM_{\perp}}\}$
$= \perp_{BM_{\perp}}$

$\square$

Law 4.5.6

$$\top_{BM_{\perp}} \sqcap_{BM_{\perp}} B = B$$

Proof.

$\top_{BM_{\perp}} \sqcap_{BM_{\perp}} B$ $\{$Definition of $\top_{BM_{\perp}}$ and $\sqcap_{BM_{\perp}}\}$
$= \emptyset \cup B$ $\{$Property of sets$\}$
$= B$

$\square$

As expected, the demonic choice between abort and some other program is abort. In the case of a miracle, the demon will avoid choosing it if possible.

Since the angelic and demonic choice operators are defined as set intersection and union, respectively, they also distribute through each other. This is exactly the same property as in the original theory of binary multirelations.

4.5.4 Sequential composition

The definition of sequential composition is not immediately obvious. In fact, one of the main reasons for developing a new binary multirelational model is that it provides a more intuitive approach to the definition of sequential composition. Consider the following example from the theory of designs.

Example 10

$(x' = 1 \vdash true) \ ;_{\mathcal{D}} I\!I_{\mathcal{D}}$ $\{$Definition of $I\!I_{\mathcal{D}}\}$
$= (x' = 1 \vdash true) \ ;_{\mathcal{D}} (true \vdash x' = x)$
$\{$Definition of sequential composition for designs$\}$
$= (\neg\, (x' \neq 1 \ ; true) \land \neg\, (true \ ; false) \vdash true \ ; x' = x)$
$\{$Definition of sequential composition$\}$

$$= (\neg (\exists x_0 \bullet x_0 \neq 1 \land true) \land \neg (\exists x_0 \bullet true \land false) \vdash \exists x_0 \bullet true \land x' = x_0)$$ {Predicate calculus and one-point rule}

$$= (\neg true \land \neg false \vdash true)$$ {Predicate calculus}

$$= (false \vdash true)$$ {Property of designs and predicate calculus}

$$= true$$ {Definition of $\bot_D$}

$$= \bot_D$$

In this case, a non-**H3**-design is sequentially composed with II_D, the Skip of the theory. The result is an aborting program. In fact, this result can be generalised for the sequential composition of any non-**H3**-design.

The behaviour just described provides the motivation for the definition of sequential composition in the new binary multirelational model.

Definition 46

$$\begin{array}{l} B_0 \ ;_{BM_\bot} B_1 \\ \widehat{=} \\ \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \exists ss : \mathbb{P}\, State_\bot \bullet (s_0, ss) \in B_0 \land \\ \left(\begin{array}{l} \bot \in ss \\ \lor \\ (\bot \notin ss \land ss \subseteq \{ s_1 : State \mid (s_1, ss_0) \in B_1 \}) \end{array} \right) \end{array} \right. \end{array} \right\} \end{array}$$

This definition is similar to the one for binary multirelations, except for the case where B_0 may lead to sets of final states where termination is not guaranteed. For sets of final states where termination is guaranteed, that is, $\bot$ is not in the set of intermediate states ss, then the definition matches that of the original theory of binary multirelations. If $\bot$ is in ss, and hence termination is not guaranteed, then the result of the sequential composition is arbitrary as it can include any set of final states.

If we assume that B_0 is **BMH0**-healthy, then the definition of sequential composition can be split into the set union of two sets as shown in Law 4.5.7.

Law 4.5.7 *Provided B_0 is* **BMH0***-healthy.*

$$\begin{array}{l} B_0 \ ;_{BM_\bot} B_1 \\ = \end{array}$$

$$\left(\begin{array}{l}\{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B_0\}\\ \cup\\ \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_1\}) \in B_0\}\end{array}\right)$$

Proof.

$B_0 \;\;_{BM_\perp} B_1$ $\quad$ {Definition of $\;_{BM_\perp}$}

$$= \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \wedge \\ \left(\begin{array}{l} \perp \in ss \\ \vee \\ (\perp \notin ss \wedge ss \subseteq \{s_1 : State \mid (s_1, ss_0) \in B_1\}) \end{array}\right) \end{array}\right. \end{array}\right\}$$

{Predicate calculus and property of sets}

$$= \left(\begin{array}{l} \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \wedge \perp \in ss \end{array}\right\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \\ \wedge (\perp \notin ss \wedge ss \subseteq \{s_1 : State \mid (s_1, ss_0) \in B_1\}) \end{array}\right. \end{array}\right\} \end{array}\right)$$

{Propositional calculus and property of sets}

$$= \left(\begin{array}{l} \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \\ \wedge \perp \in ss \wedge ss \subseteq State_\perp \end{array}\right. \end{array}\right\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \\ \wedge \perp \notin ss \wedge ss \subseteq \{s_1 : State \mid (s_1, ss_0) \in B_1\} \end{array}\right. \end{array}\right\} \end{array}\right)$$

{$\perp$ in $State_\perp$ and $\perp$ not in $State$}

$$= \left(\begin{array}{l} \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \\ \wedge \perp \in ss \wedge ss \subseteq State_\perp \wedge \perp \in State_\perp \end{array}\right. \end{array}\right\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss : \mathbb{P}\, State_\perp \bullet (s_0, ss) \in B_0 \\ \wedge \perp \notin ss \wedge ss \subseteq \{s_1 : State \mid (s_1, ss_0) \in B_1\} \\ \wedge \perp \notin \{s_1 : State \mid (s_1, ss_0) \in B_1\} \end{array}\right. \end{array}\right\} \end{array}\right)$$

$$\{\text{Assumption: } B_0 \text{ is } \mathbf{BMH0}\text{-healthy and Law B.2.1}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B_0\} \\ \cup \\ \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_1\}) \in B_0\} \end{array}\right)$$

$\square$

The first set considers the case when B_0 leads to sets of final states where termination is not required ($State_\perp$). The second set considers the case where termination is required.

For a similar example to Example 10 expressed in the new theory, we consider the following example, where a non-terminating assignment is followed by the assignment that requires termination, but does not change the value of x.

Example 11

$$(x :=_{BM_\perp} e) \ ;_{BM_\perp} (x :=_{BM} x) \qquad \{\text{Definition of } ;_{BM_\perp} \text{ (Law 4.5.7)}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in (x :=_{BM_\perp} e)\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in (x :=_{BM} x)\}) \in (x :=_{BM_\perp} e) \end{array}\right\} \end{array}\right)$$

$$\{\text{Definition of } :=_{BM} \text{ and } :=_{BM_\perp}\}$$

$$= \left(\begin{array}{l} \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, State_\perp) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid s \oplus (x \mapsto e) \in ss\} \end{array}\right\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s_0, \{s_1 : State \mid (s_1, ss_0) \in (x :=_{BM} x)\}) \\ \in \\ \{s : State, ss : \mathbb{P}\, State \mid s \oplus (x \mapsto e) \in ss\} \end{array}\right. \end{array}\right\} \end{array}\right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid s_0 \oplus (x \mapsto e) \in State_\perp\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid s_0 \oplus (x \mapsto e) \in \{s_1 : State \mid (s_1, ss_0) \in (x :=_{BM} x)\} \end{array}\right\} \end{array}\right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid true\} \\ \cup \\ \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid s_0 \oplus (x \mapsto e) \in \{s_1 : State \mid (s_1, ss_0) \in (x :=_{BM} x)\} \end{array} \right\} \end{array} \right)$$

$$\{\text{Property of sets and definition of } \perp_{BM_\perp}\}$$

$$= \perp_{BM_\perp}$$

The result of this sequential composition is an aborting program. If it is possible for the first program not to terminate, then the sequential composition cannot provide any guarantees either. The properties observed by the sequential composition operator are explored in what follows.

Properties

The first property of interest considers the sequential composition of $\top_{BM_\perp}$ followed by some program B. The result is also a miraculous program as shown in the following Law 4.5.8

Law 4.5.8

$$\top_{BM_\perp} \; ;_{BM_\perp} B = \top_{BM_\perp}$$

Proof.

$$\top_{BM_\perp} \; ;_{BM_\perp} B \qquad \{\text{Definition of } \top_{BM_\perp}\}$$

$$= \emptyset \; ;_{BM_\perp} B$$

$$\{\text{Definition of } ;_{BM_\perp} \text{ (Law 4.5.7 as } \top_{BM_\perp} \text{ is } \textbf{BMH0}\text{-healthy)}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in \emptyset\} \\ \cup \\ \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_1\}) \in \emptyset\} \end{array} \right)$$

$$\{\text{Property of sets}\}$$

$$= \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid false\} \cup \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid false\}$$

$$\{\text{Property of sets}\}$$

$$= \emptyset \cup \emptyset \qquad \{\text{Property of sets and definition of } \top_{BM_\perp}\}$$

$$= \top_{BM_\perp}$$

□

The following law expresses that the sequential composition of abort with another program is also abort.

Law 4.5.9

$$\perp_{BM_\perp} \ ;_{BM_\perp} B = \perp_{BM_\perp}$$

Proof.

$$
\begin{aligned}
&\perp_{BM_\perp} \ ;_{BM_\perp} B && \{\text{Definition of } \perp_{BM_\perp}\} \\
&= (State \times \mathbb{P}\, State_\perp) \ ;_{BM_\perp} B \\
& && \{\text{Definition of } ;_{BM_\perp} \text{ (Law 4.5.7 as } \perp_{BM_\perp} \text{ is } \mathbf{BMH0}\text{-healthy)}\} \\
&= \left(\begin{array}{l}
\{\ s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in (State \times \mathbb{P}\, State_\perp)\ \} \\
\cup \\
\left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B\}) \in (State \times \mathbb{P}\, State_\perp) \end{array}\right\}
\end{array}\right) \\
& && \{\text{Property of sets}\} \\
&= \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid true\} \cup \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid true\} \\
& && \{\text{Property of sets}\} \\
&= (State \times \mathbb{P}\, State_\perp) && \{\text{Definition of } \perp_{BM_\perp}\} \\
&= \perp_{BM_\perp}
\end{aligned}
$$

□

In the following paragraphs we explore some examples with respect to the extreme points of the theory.

Examples

The following example describes the general behaviour of some program B that is **BMH0**-healthy sequentially composed with a miraculous program.

Example 12

$$
\begin{aligned}
&B \ ;_{BM_\perp} \top_{BM_\perp} && \{\text{Definition of } \top_{BM_\perp} \text{ and } ;_{BM_\perp} \text{ (Law 4.5.7)}\} \\
&= \left(\begin{array}{l}
\{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B\} \\
\cup \\
\{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in \emptyset\}) \in B\}
\end{array}\right) \\
& && \{\text{Property of sets}\}
\end{aligned}
$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B\} \\ \cup \\ \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \emptyset) \in B\} \end{array} \right)$$

If B may not terminate for some set of final states, and it is **BMH0**-healthy, then the result of the sequential composition is also abort, as $State_\perp$ is in B. If B aborts for some particular initial state s_0, then that state is related to the empty set in B and the result of the sequential composition is also abort. Otherwise, the result is miraculous as the union of both sets if the empty set.

The following example describes the behaviour of a program B sequentially composed with abort.

Example 13

$$B \;\, ;_{BM_\perp} \perp_{BM_\perp} \qquad \{\text{Definition of } \perp_{BM_\perp} \text{ and } ;_{BM_\perp} \text{ (Law 4.5.7)}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B\} \\ \cup \\ \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in (State \times \mathbb{P}\, State_\perp)\}) \in B \end{array} \right\} \end{array} \right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B\} \\ \cup \\ \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, \{s_1 : State \mid true\}) \in B\} \end{array} \right)$$

$$\{\text{Property of sets}\}$$

$$= \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B \lor (s_0, State) \in B\}$$

Because B is upward closed, if it definitely terminates then $State$ is a superset of all sets of final states and is in B. If B may or may not terminate for some particular set of final states, then $State_\perp$ is also in B due to the upward closure guaranteed by **BMH0**. In either case, the sequential composition behaves as abort. If B is miraculous, then so is the sequential composition.

4.6 Relationship with binary multirelations

In this section we focus our attention on the relationship between the subset of the theory that is **BMH3**-healthy and the original theory of binary multirelations [15]. In the following Sections 4.6.1 and 4.6.2 we define the linking functions that relate both models. Finally in Section 4.6.3 we prove that the linking functions form a bijection under the respective healthiness conditions of each theory.

4.6.1 *bmb2bm*

The function *bmb2bm* maps binary multirelations in the new model, of type $BM_\perp$, to those in the original model. It is defined by considering every pair in (s, ss) in B such that $\perp$ is not in ss.

Definition 47 (*bmb2bm*)

$$\begin{array}{l} bmb2bm : BM_\perp \nrightarrow BM \\ bmb2bm(B) \mathrel{\widehat=} \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, ss) \in B \land \perp \notin ss)\} \end{array}$$

In order to show that *bmb2bm* yields a binary multirelation that is **BMH**-healthy, we first calculate the result of applying *bmb2bm* to a relation that is **BMH0**-**BMH3**-healthy in Lemma 4.6.1. Finally in Theorem 4.6.1 we prove that *bmb2bm* yields a **BMH**-healthy binary multirelation.

Lemma 4.6.1

$$\begin{array}{l} bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) \\ = \\ \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin ss \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet (\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ ) \end{array} \right) \end{array} \right. \end{array} \right\} \end{array}$$

Proof.

$$\begin{array}{ll} bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) & \{\text{Definition of } bmb2bm\} \\ = \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, ss) \in \mathbf{bmh_{0,1,3,2}}(B) \land \perp \notin ss)\} & \\ & \{\text{Definition of } \mathbf{bmh_{0,1,3,2}}\} \end{array}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \land \perp \notin ss \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right) \\ \land \perp \notin ss \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin ss \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

□

Theorem 4.6.1 ($\mathbf{bmb2bm\text{-}is\text{-}bmh_{upclosed}}$)

$$\mathbf{bmh_{upclosed}} \circ bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) = bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$$

Proof.

$\mathbf{bmh_{upclosed}} \circ bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$ {Definition of $\mathbf{bmh_{upclosed}}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right\}$$

{Law B.2.17}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\bot \\
\left| \begin{array}{l}
\exists ss_0 \bullet (s, ss_0) \in \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\bot \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \land \bot \notin ss \\
\lor \\
\left(\begin{array}{l}
(s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss\ \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\} \\
\land \\
\bot \notin ss_0 \land ss_0 \subseteq ss \land \bot \notin ss
\end{array} \right.
\end{array} \right\}
$$

{Variable renaming and property of sets}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\bot \\
\left| \begin{array}{l}
\exists ss_0 \bullet \left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \land \bot \notin ss_0 \\
\lor \\
\left(\begin{array}{l}
(s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists ss_1 \bullet \left(\ (s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \bot \notin ss_1 \land \bot \notin ss_0\ \right)
\end{array} \right)
\end{array} \right) \\
\land \\
\bot \notin ss_0 \land ss_0 \subseteq ss \land \bot \notin ss
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus: distributivity and quantifier scope}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\bot \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \land \bot \notin ss \land \exists ss_0 \bullet \bot \notin ss_0 \land ss_0 \subseteq ss \\
\lor \\
\left(\begin{array}{l}
(s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists ss_1, ss_0 \bullet \left(\begin{array}{l}
(s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \bot \notin ss_1 \land \bot \notin ss_0 \\
\land \\
\bot \notin ss_0 \land ss_0 \subseteq ss \land \bot \notin ss
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus: case-analysis on ss_0}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\bot \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \land \bot \notin ss \\
\lor \\
\left(\begin{array}{l}
(s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists ss_1, ss_0 \bullet \left(\begin{array}{l}
(s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \bot \notin ss_1 \land \bot \notin ss_0 \\
\land \\
\bot \notin ss_0 \land ss_0 \subseteq ss \land \bot \notin ss
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin ss \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_1 \bullet \left(\; (s, ss_1) \in B \land ss_1 \subseteq ss \land \perp \notin ss_1 \land \perp \notin ss \; \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Lemma 4.6.1}

$$= bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$$

□

This result establishes that for **BMH0**-**BMH3**-healthy relations $bmb2bm(B)$ yields relations that are in the original theory.

4.6.2 $bm2bmb$

The function that maps from relations in the original model, of type BM, into the new model is $bm2bmb$ and its definition is presented below.

Definition 48 ($bm2bmb$**)**

$$bm2bmb : BM \to BM_\perp$$

$$bm2bmb(B) \mathrel{\widehat=} \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, ss) \in B \land \perp \notin ss) \lor (s, \emptyset) \in B \end{array} \right\}$$

It considers every pair (s, ss) in B where $\perp$ is not in the set of final states ss, or if B is aborting for a particular initial state s, then the result is the universal relation of type $BM_\perp$.

In the following Lemma 4.6.2 we calculate the result of applying $bm2bmb$ to a relation that is **BMH**-healthy. Finally, Theorem 4.6.2 establishes that $bm2bmb$ yields relations that are **BMH0**-**BMH3**-healthy.

Lemma 4.6.2

$$bm2bmb(\mathbf{bmh_{upclosed}}(B))$$
$$=$$
$$\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\}$$

Proof.

$bm2bmb(\mathbf{bmh_{upclosed}}(B))$ $\{$Definition of $bm2bmb\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, ss) \in \mathbf{bmh_{upclosed}}(B) \land \perp \notin ss) \lor (s, \emptyset) \in \mathbf{bmh_{upclosed}}(B) \end{array} \right\}$$

$\{$Definition of $\mathbf{bmh_{upclosed}}\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left((s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right\} \land \perp \notin ss \right) \\ \lor \\ (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right\} \end{array} \right. \end{array} \right\}$$

$\{$Property of sets and predicate calculus$\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq \emptyset \end{array} \right. \end{array} \right\}$$

$\{$Case-analysis on ss_0 and one-point rule$\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\}$$

□

Theorem 4.6.2

$$\mathbf{bmh_{0,1,3,2}} \circ bm2bmb(\mathbf{bmh_{upclosed}}(B)) = bm2bmb(\mathbf{bmh_{upclosed}}(B))$$

Proof.

$\mathbf{bmh_{0,1,3,2}} \circ bm2bmb(\mathbf{bmh_{upclosed}}(B))$ $\{$Definition of $\mathbf{bmh_{0,1,3,2}}\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \land (s,\{\perp\}) \in bm2bmb(\mathbf{bmh_{upclosed}}(B))) \\ \lor \\ \left(\begin{array}{l} (s,\{\perp\}) \notin bm2bmb(\mathbf{bmh_{upclosed}}(B)) \land (s,\emptyset) \notin bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Law B.2.19 and Law B.2.18}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in B \land (s,\emptyset) \in B) \\ \lor \\ \left(\begin{array}{l} ((s,\emptyset) \notin B \land (s,\emptyset) \notin B) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus and definition of $bm2bmb(\mathbf{bmh_{upclosed}}(B))$ (Law B.2.16)}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s,\emptyset) \in B \\ \lor \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s,\emptyset) \in B \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Variable renaming and property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s,\emptyset) \in B \\ \lor \\ \exists\, ss_0 \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_1 \bullet (s, ss_1) \in B \land \perp \notin ss_1 \land ss_1 \subseteq ss_0 \land \perp \notin ss_0 \\ \lor \\ (s,\emptyset) \in B \end{array} \right) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_0 \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\exists\, ss_0 \bullet (s, \emptyset) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ (\exists\, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ ((s, \emptyset) \in B \wedge \exists\, ss_0 \bullet ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ (\exists\, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss \wedge \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Law B.2.16}

$$= bm2bmb(\mathbf{bmh_{upclosed}}(B))$$

□

These results complete the proofs for healthiness regarding both linking functions. In the following section we discuss the isomorphism.

4.6.3 *bm2bmb* and *bmb2bm*

Using the results from the previous section we establish that *bm2bmb* and *bmb2bm* form a bijection for healthy relations. Theorem 4.6.3 establishes this for relations that are **BMH0**-**BMH3**-healthy, while Theorem 4.6.4 establishes the bijection for relations that are **BMH**-healthy.

Theorem 4.6.3 *Provided B is* **BMH0**-**BMH3**-*healthy.*

$$bm2bmb \circ bmb2bm(B) = B$$

Proof.

$$
\begin{array}{ll}
bm2bmb \circ bmb2bm(B) & \{\text{Assumption: } B \text{ is } \mathbf{BMH0}\text{-}\mathbf{BMH3}\text{-healthy}\} \\
= bm2bmb \circ bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) & \{\text{Definition of } bm2bmb\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, ss) \in bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) \land \perp \notin ss) \\ \lor \\ (s, \emptyset) \in bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) \end{array} \right. \end{array} \right\} & \{\text{Law B.2.17}\}
\end{array}
$$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B \land \perp \notin ss) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 \bullet \left((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \right) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \land \perp \notin ss \end{array} \right) \\ \lor \\ (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B \land \perp \notin ss) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 \bullet \left((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \right) \end{array} \right) \end{array} \right. \end{array} \right\} \end{array} \right. \end{array} \right\}
$$

$$\{\text{Property of sets}\}$$

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\perp \\
\left| \begin{array}{l}
\left(\begin{array}{l}
\left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B \land \perp \notin ss) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right)
\end{array} \right)
\end{array} \right) \\
\land \perp \notin ss
\end{array} \right) \\
\lor \\
\left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin \emptyset \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq \emptyset \land \perp \notin ss_0 \land \perp \notin \emptyset\ \right)
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Property of sets, predicate calculus and one-point rule}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\perp \\
\left| \begin{array}{l}
\left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B \land \perp \notin ss) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right)
\end{array} \right)
\end{array} \right) \\
\lor \\
\left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
(s, \emptyset) \in B
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l}
s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\perp \\
\left| \begin{array}{l}
\left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B \land \perp \notin ss) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right)
\end{array} \right)
\end{array} \right) \\
\lor \\
((s, \emptyset) \in B \land (s, \{\perp\}) \in B)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Definition of $\mathbf{bmh_{0,1,3,2}}$}

$= \mathbf{bmh_{0,1,3,2}}(B)$ {Assumption: B is **BMH0**-**BMH3**-healthy}

$= B$

□

Theorem 4.6.4 *Provided B is* **BMH**-*healthy.*

$$bmb2bm \circ bm2bmb(B) = B$$

Proof.

$bmb2bm \circ bm2bmb(B)$ {Assumption: B is **BMH**-healthy}

$= bmb2bm \circ bm2bmb(\mathbf{bmh_{upclosed}}(B))$ {Definition of $bmb2bm$}

$= \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, ss) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \land \perp \notin ss)\}$

{Law B.2.16}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \left(\begin{array}{l} (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\} \\ \land \\ \perp \notin ss \end{array} \right) \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right) \\ \land \\ \perp \notin ss \end{array} \right) \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss) \\ \lor \\ ((s, \emptyset) \in B \land \perp \notin ss) \end{array} \right. \end{array} \right\}$$

$$\{\text{Instantiation: consider case where } ss_0 = \emptyset\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right\}$$

$$\{\text{Definition of } \mathbf{bmh_{upclosed}}\}$$

$$= \mathbf{bmh_{upclosed}}(B) \qquad \{\text{Assumption: } B \text{ is } \mathbf{BMH}\text{-healthy}\}$$

$$= B$$

□

These results show that the subset of the theory that is **BMH3**-healthy is isomorphic to the original theory of binary multirelations [15]. This confirms that while our model is more expressive, it is still possible to express every program that could be specified using the original model.

4.7 Final considerations

In this section we have introduced a new binary multirelational model that allows specifying sets of final states for which termination is not required. This model extends that of [15] by using the symbol $\perp$ to denote the possibility for non-termination. The healthiness conditions have been introduced as predicates and subsequently characterised as fixed points of idempotent functions. These functions have been studied at length and their functional composition has been justified.

The operators of the theory have been introduced and their properties studied. The definition of sequential composition is the most unexpected. Its intuition comes from the theory of designs. The full justification for the definition of some of the operators and of the refinement order, is deferred until the study of the equivalent predicative model in the following Chapter 5.

Chapter 5

Designs with angelic nondeterminism

In this chapter we introduce a new UTP theory of designs that embodies the notion of angelic nondeterminism. The starting points for this predicative model are the theory of [14] and the binary multirelational model presented in Chapter 4. For this reason we begin this chapter by discussing the choice of alphabet in Section 5.1 and its relationship with that of [14].

In Section 5.2 the healthiness conditions of the theory are defined. These are specified by idempotent and monotonic functions whose fixed points are the designs of interest.

Since this theory is a predicative account of the model of Chapter 4, in Section 5.3 we establish that these models are isomorphic. This is achieved by defining a pair of linking functions and subsequently proving that they form a bijection. This result enables us, for example, to establish the correspondence between the healthiness conditions and operators of both models.

In Section 5.4 we justify that the theory of designs that we propose is a complete lattice. The definition of refinement adopted is the same as in the original theory of designs. Furthermore, we prove that this corresponds exactly to the refinement ordering of the binary multirelational model of Chapter 4, which is defined as subset ordering.

Section 5.5 discusses the main operators of the theory, including assignment and sequential composition. The entire Section 5.6 is dedicated to the main focus of this theory: angelic and demonic nondeterminism. Finally, in Section 5.7 we show that the subset of **H3**-designs of our theory is isomorphic to the UTP model of [14].

5.1 Alphabet

The result in [14] establishes that demonic and angelic nondeterminism cannot be both directly modelled in the relational setting of the UTP. To address that, Cavalcanti et al. [14] propose a non-homogeneous theory that can encode demonic and angelic choices. Our aim is to build on that model, which is isomorphic to the monotonic predicate transformers [14], and define a theory of designs (that includes the observational variables ok and ok' and can describe both demonic and angelic nondeterminism). In order to put our choice of alphabet into perspective, we first explain the reasoning behind the alphabet used in [14].

The work of Cavalcanti et al. [14] considers an alphabet that includes the undashed program variables and, as the only dashed variable, ac'. This sole dashed variable represents the set of final states that can be chosen by the angel. A state is a record whose components represent program variables. For example, if we specify a program that uses the program variable x, then each state in ac' must contain a component of name x', whose value is one of the possible final values of x'.

The non-homogeneous relations can be understood as establishing the relationship between an initial state and a set of possible final states corresponding to the choices available to the angel. For example, in the case of the program specified by $x := 1 \sqcup x := 2$, where $\sqcup$ is the angelic choice operator, the set of outcomes ac' includes at least two states whose component x' is set to the possible final values of x, 1 and 2, respectively.

Perhaps, the most surprising observation we can make about the theory in [14] is the absence of variables such as ok and ok', although it captures termination. In particular, the healthiness conditions of that theory correspond to **H1**, **H2** and in fact **H3** as well. However, for our purposes, it is essential to use the variables ok and ok' as other theories of interest, namely the theory of reactive processes [7], make use of these. Furthermore, as mentioned before, it is absolutely vital that we can consider non-**H3** designs.

The theory that we propose is, therefore, a theory of designs: we consider an alphabet that includes ok and ok'. In addition, we introduce two variables s and ac' as shown below.

Definition 49 (Alphabet)

$$s : State$$

$ac' : \mathbb{P}\, State$
$ok, ok' : \{true, false\}$

We observe that as mentioned in Chapter 2, it is possible to define a non-homogeneous theory of designs with ok and ok'.

The variable s encapsulates the initial values of program variables as record components: each component corresponds to an undashed program variable. The set of final states ac' is similar to that of [14] with the notable difference that we do not dash the variables in the record components, instead we only consider these as undashed. This simplifies reasoning and proofs. We observe that we still make an explicit distinction between the initial state, which are encoded by s, and the final states, which are encoded in the set defined by ac'. It is possible to relate the two sets through the following pair of functions.

Definition 50 (acdash-to-ac)

$$\begin{aligned} acdash2ac(ss) &= \left\{ \begin{array}{l} s_0 : S_{in\alpha}, s_1 : S_{out\alpha} \\ \mid s_1 \in ss \land (\bigwedge x : \alpha P \bullet s_0.x = s_1.(x')) \bullet s_0 \end{array} \right\} \\ ac2acdash(zz) &= \left\{ \begin{array}{l} z_0 : S_{in\alpha}, z_1 : S_{out\alpha} \\ \mid z_0 \in ss \land (\bigwedge x : \alpha P \bullet z_0.x = z_1.(x')) \bullet z_1 \end{array} \right\} \end{aligned}$$

The function $acdash2ac$ maps a set ss of angelic choices whose record components are dashed variables into a set whose record components are undashed. This is achieved by considering every state s_1 in ss and every state s_0, such that s_0 is a state on the undashed variables of P and whose components are exactly the same as those in s_1, except that those in s_1 are dashed. Each state is characterised by its alphabet, so in the case of the dashed sate $s_1 : S_{out\alpha}$ this corresponds to a state whose record components are those in the output alphabet, $out\alpha$, for some program.

These two functions are important in the definition of a link between the theories as explained later in Section 5.7. In the following section we introduce the healthiness conditions.

5.2 Healthiness conditions

The theory we propose is a theory of designs. Therefore, predicates at the very are fixed points of **H1** and **H2**. Furthermore, since we seek to integrate designs with a model similar to [14], a consistent notion of termination must be established. In that model, non-termination is possible if the set of angelic choices can be empty [14]. However, explicit non-termination cannot be required since the theory adopts **H2** as a healthiness condition.

In addition to characterising termination appropriately, we also need to ensure that the set of final choices ac' is upward closed. The reason behind this is further explained in Section 5.2.2. These two concerns are addressed separately by the healthiness conditions **A0** and **A1**, respectively. We introduce **A0** in Section 5.2.1 and **A1** in Section 5.2.2. Finally in Section 5.2.3 both functions are composed together and their combined properties explored.

5.2.1 A0

The first healthiness condition provides a consistent treatment of termination between the auxiliary variable ok' and the value of ac' in the theory. It is defined as follows.

Definition 51 (A0) *If ok' holds, then ac' cannot be empty. Otherwise any value for ac' is allowed.*

$$\mathbf{A0}(P) \mathrel{\widehat=} P \land ((ok \land \neg P^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset))$$

A0 states that when a design P terminates, that is ok' is *true*, then it must also be the case that ac' is not empty. In other words, there must be at least one final state in ac' available to the angel. If the precondition $\neg P^f$ is not satisfied then the design aborts and there are no guarantees on the outcome. This embodies the notion of termination as found in [14] and related models, such as binary multirelations [15]. This particular definition ensures that **H1** and **H2** are preserved as shown in the following section.

Properties

In the following laws we show that **A0** is closed with respect to designs, idempotent, and monotonic with respect to the refinement ordering.

Law 5.2.1 (A0-design) *If P is a design so is $\mathbf{A0}(P)$.*

$$\mathbf{A0}(P) = (\neg P^f \vdash P^t \land ac' \neq \emptyset)$$

Proof.

$$
\begin{aligned}
&\mathbf{A0}(P) && \{\text{Definition of design and } \mathbf{A0}\}\\
&= (\neg\, P^f \vdash P^t) \wedge ((ok \wedge \neg\, P^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset)) \\
& && \{\text{Definition of design and propositional calculus}\}\\
&= (ok \wedge \neg\, P^f) \Rightarrow (P^t \wedge ok' \wedge (ok' \Rightarrow ac' \neq \emptyset)) && \{\text{Propositional calculus}\}\\
&= (ok \wedge \neg\, P^f) \Rightarrow (P^t \wedge ok' \wedge ac' \neq \emptyset) && \{\text{Definition of design}\}\\
&= (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset)
\end{aligned}
$$

□

Law 5.2.1 establishes that a design in our theory can be stated in the usual manner, with a precondition and a postcondition, but the postcondition must guarantee that ac' is not equal to the empty set. In other words, once its precondition is satisfied, it establishes the postcondition and terminates.

Law 5.2.2 (A0-idempotent)

$$\mathbf{A0} \circ \mathbf{A0}(P) = \mathbf{A0}(P)$$

Proof.

$$
\begin{aligned}
&\mathbf{A0} \circ \mathbf{A0}(P) && \{\text{Law 5.2.1}\}\\
&= \mathbf{A0}(\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset) && \{\text{Law 5.2.1}\}\\
&= (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset \wedge ac' \neq \emptyset) && \{\text{Propositional calculus}\}\\
&= (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset) && \{\text{Definition of } \mathbf{A0}\}\\
&= \mathbf{A0}(P)
\end{aligned}
$$

□

Law 5.2.3 (A0-monotonic)

$$(P \sqsubseteq Q) \Rightarrow (\mathbf{A0}(P) \sqsubseteq \mathbf{A0}(Q))$$

Proof.

$$
\begin{aligned}
&\mathbf{A0}(Q) && \{\text{Definition of } \mathbf{A0}\}\\
&= Q \wedge ((ok \wedge \neg\, Q^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset)) \\
& && \{\text{Assumption: } [Q \Rightarrow P] \Leftrightarrow [\neg\, P \Rightarrow \neg\, Q]\}\\
&\Rightarrow P \wedge ((ok \wedge \neg\, P^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset)) && \{\text{Definition of } \mathbf{A0}\}\\
&= \mathbf{A0}(P)
\end{aligned}
$$

$\square$

Law 5.2.2 establishes that the function **A0** is idempotent, and Law 5.2.3 establishes that it is monotonic. These results confirm the suitability of **A0** as a healthiness condition. In the following section we explore the closure properties of **A0**.

Closure properties

In the following laws we show that **A0** is closed with respect to disjunction and conjunction.

Law 5.2.4 (A0-conjunction-closure) *Provided P and Q are* **A0***-healthy.*

$$\mathbf{A0}(P \wedge Q) = P \wedge Q$$

Proof.

$$
\begin{aligned}
& P \wedge Q && \{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A0}\text{-healthy}\} \\
& = \mathbf{A0}(P) \wedge \mathbf{A0}(Q) && \{\text{Definition of } \mathbf{A0}\} \\
& = \left(\begin{array}{l} (P \wedge ((ok \wedge \neg P^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset))) \\ \wedge \\ (Q \wedge ((ok \wedge \neg Q^f) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset))) \end{array}\right) && \{\text{Propositional calculus}\} \\
& = (P \wedge Q) \wedge (((ok \wedge \neg P^f) \vee (ok \wedge \neg Q^f)) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset)) \\
& && \{\text{Propositional calculus}\} \\
& = (P \wedge Q) \wedge ((ok \wedge \neg (P^f \wedge Q^f)) \Rightarrow (ok' \Rightarrow ac' \neq \emptyset)) \\
& && \{\text{Definition of } \mathbf{A0}\} \\
& = \mathbf{A0}(P \wedge Q)
\end{aligned}
$$

$\square$

Law 5.2.5 (A0-disjunction-closure) *Provided P and Q are* **A0***-healthy.*

$$\mathbf{A0}(P \vee Q) = P \vee Q$$

Proof.

$$
\begin{aligned}
& P \vee Q && \{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A0}\text{-healthy}\} \\
& = \mathbf{A0}(P) \vee \mathbf{A0}(Q) && \{\text{Definition of } \mathbf{A0}\}
\end{aligned}
$$

$$
\begin{aligned}
&= (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset) \vee (\neg\, Q^f \vdash Q^t \wedge ac' \neq \emptyset) \\
&\hspace{20em} \{\text{Disjunction of designs}\} \\
&= (\neg\, P^f \wedge \neg\, Q^f \vdash (P^t \wedge ac' \neq \emptyset) \vee (Q^t \wedge ac' \neq \emptyset)) \\
&\hspace{20em} \{\text{Propositional calculus}\} \\
&= (\neg\, (P^f \vee Q^f) \vdash (P^t \vee Q^t) \wedge ac' \neq \emptyset) \qquad \{\text{Property of substitution}\} \\
&= (\neg\, (P \vee Q)^f \vdash (P \vee Q)^t \wedge ac' \neq \emptyset) \qquad \{\text{Definition of } \mathbf{A0}\} \\
&= \mathbf{A0}(P \vee Q)
\end{aligned}
$$

□

We observe that the proofs for both Law 5.2.4 and Law 5.2.5 also show that **A0** distributes over conjunction and disjunction, irrespective of satisfying their provisos. This concludes our discussion of the basic properties of **A0**.

5.2.2 A1

In addition to requiring a consistent treatment of termination, our theory of designs requires that both pre and postcondition observe the upward closure of the set of final states, ac'. When this requirement is applied on simple predicates, this corresponds exactly to the healthiness condition **PBMH** of [14]. The definition is reproduced below.

Definition 52 (PBMH)

$$\mathbf{PBMH}(P) \mathrel{\widehat=} P \,;\, ac \subseteq ac'$$

For every fixed point P of **PBMH**, the value of ac' must be upward closed. We observe that the function **PBMH** is idempotent. Other properties of interest are established in Appendix D.

The requirement upon our theory of designs regarding upward closure concerns both pre and postcondition. This is specified by the following healthiness condition **A1**.

Definition 53 (A1)

$$\mathbf{A1}(P_0 \vdash P_1) \mathrel{\widehat=} (\neg\, \mathbf{PBMH}(\neg\, P_0) \vdash \mathbf{PBMH}(P_1))$$

The upward closure of ac' in the postcondition is enforced exactly as in [14]. However, the precondition is treated differently. In this case we ensure that it is the negation of the precondition that is upward closed, since it is the negation that actually establishes the value of ac' for designs that do not require termination. This can be illustrated by the following Lemma 5.2.1.

Lemma 5.2.1

$$\mathbf{A1}(P_0 \vdash P_1) = ok \Rightarrow (((P_1 \mathbin{;} ac \subseteq ac') \wedge ok') \vee (\neg\, P_0 \mathbin{;} ac \subseteq ac'))$$

Proof.

$$\begin{aligned}
&\mathbf{A1}(P_0 \vdash P_1) && \{\text{Definition of } \mathbf{A1}\}\\
&= (\neg\,(\neg\, P_0 \mathbin{;} ac \subseteq ac') \vdash P_1 \mathbin{;} ac \subseteq ac') && \{\text{Definition of designs}\}\\
&= (ok \wedge \neg\,(\neg\, P_0 \mathbin{;} ac \subseteq ac')) \Rightarrow ((P_1 \mathbin{;} ac \subseteq ac') \wedge ok') &&\\
& && \{\text{Predicate calculus}\}\\
&= ok \Rightarrow (((P_1 \mathbin{;} ac \subseteq ac') \wedge ok') \vee (\neg\, P_0 \mathbin{;} ac \subseteq ac')) &&
\end{aligned}$$

□

When the program is started it can either terminate, in which case ok' is *true* and P_1 is established, or $\neg\, P_0$ is established and termination is then not required. In either case we enforce the upward closure of ac'.

This concludes our discussion of the definition of **A1**. In the sequel we show how it satisfies some basic properties.

Properties

In the following Laws 5.2.6 and 5.2.7 we establish that **A1** is an idempotent and monotonic function.

Law 5.2.6 (A1-idempotent)

$$\mathbf{A1} \circ \mathbf{A1}(P_0 \vdash P_1)$$

Proof.

$$\begin{aligned}
&\mathbf{A1} \circ \mathbf{A1}(P_0 \vdash P_1) && \{\text{Definition of } \mathbf{A1}\}\\
&= \mathbf{A1} \circ (\neg\, \mathbf{PBMH}(\neg\, P_0) \vdash \mathbf{PBMH}(P_1)) && \{\text{Definition of } \mathbf{A1}\}\\
&= (\neg\,(\mathbf{PBMH}(\neg\,\neg\, \mathbf{PBMH}(\neg\, P_0))) \vdash \mathbf{PBMH} \circ \mathbf{PBMH}(P_1)) &&\\
& && \{\text{Propositional calculus}\}
\end{aligned}$$

$$= (\neg\,(\mathbf{PBMH} \circ \mathbf{PBMH}(\neg\, P_0)) \vdash \mathbf{PBMH} \circ \mathbf{PBMH}(P_1)) \qquad \{\text{Law D.1.1}\}$$
$$= (\neg\,(\mathbf{PBMH}(\neg\, P_0)) \vdash \mathbf{PBMH}(P_1)) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= \mathbf{A1}(P_0 \vdash P_1)$$

□

Law 5.2.7 (A1-monotonic)

$$(P \sqsubseteq Q) \Rightarrow \mathbf{A1}(P) \sqsubseteq \mathbf{A1}(Q)$$

Proof.

$$\mathbf{A1}(Q) \qquad \{\text{Definition of design}\}$$
$$= \mathbf{A1}(\neg\, Q^f \vdash Q^t) \qquad \{\text{Definition of design and propositional calculus}\}$$
$$= \mathbf{A1}((\neg\, ok \vee Q^f) \vee (Q^t \wedge ok')) \qquad \{\text{Assumption: } [Q \Rightarrow P] \text{ holds}\}$$
$$= \mathbf{A1}((\neg\, ok \vee (Q^f \wedge (Q^f \Rightarrow P^f))) \vee (Q^t \wedge (Q^t \Rightarrow P^t) \wedge ok'))$$
$$\{\text{Predicate calculus and definition of design}\}$$
$$= \mathbf{A1}(\neg\,(Q^f \wedge P^f) \vdash Q^t \wedge P^t) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= (\neg\, \mathbf{PBMH}(Q^f \wedge P^f) \vdash \mathbf{PBMH}(Q^t \wedge P^t)) \qquad \{\text{Definition of } \mathbf{PBMH}\}$$
$$= (\neg\, \mathbf{PBMH}(Q^f \wedge P^f) \vdash \mathbf{PBMH}(Q^t \wedge P^t))$$
$$\{\text{Definition of sequential composition}\}$$
$$= (\neg\, \exists\, ac_0 \bullet Q^f[ac_0/ac'] \wedge P^f[ac_0/ac'] \wedge ac_0 \subseteq ac' \vdash (Q^t \wedge P^t)\ ;\ ac \subseteq ac')$$
$$\{\text{Predicate calculus}\}$$
$$= \left(\begin{array}{l} \forall\, ac_0 \bullet \neg\, Q^f[ac_0/ac'] \vee \neg\, P^f[ac_0/ac'] \vee \neg\,(ac_0 \subseteq ac') \\ \vdash \\ (Q^t \wedge P^t)\ ;\ ac \subseteq ac' \end{array}\right)$$
$$\{\text{Predicate calculus}\}$$
$$= \left(\begin{array}{l} \forall\, ac_0 \bullet \left(\begin{array}{l} (\neg\, Q^f[ac_0/ac'] \vee \neg\,(ac_0 \subseteq ac')) \\ \vee \\ (\neg\, P^f[ac_0/ac'] \vee \neg\,(ac_0 \subseteq ac')) \end{array}\right) \\ \vdash \\ (Q^t \wedge P^t)\ ;\ ac \subseteq ac' \end{array}\right)$$
$$\{\text{Weaken precondition}\}$$

$$\sqsupseteq \left(\begin{array}{l} \forall\, ac_0 \bullet (\neg\, Q^f[ac_0/ac'] \vee \neg\, (ac_0 \subseteq ac')) \\ \vee \\ \forall\, ac_0 \bullet (\neg\, P^f[ac_0/ac'] \vee \neg\, (ac_0 \subseteq ac')) \\ \vdash \\ (Q^t \wedge P^t) \;;\; ac \subseteq ac' \end{array}\right) \qquad \{\text{Weaken precondition}\}$$

$$\sqsupseteq (\forall\, ac_0 \bullet (\neg\, P^f[ac_0/ac'] \vee \neg\, (ac_0 \subseteq ac')) \vdash (Q^t \wedge P^t) \;;\; ac \subseteq ac')$$
$$\{\text{Predicate calculus}\}$$
$$= (\neg\, \exists\, ac_0 \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac' \vdash (Q^t \wedge P^t) \;;\; ac \subseteq ac')$$
$$\{\text{Definition of sequential composition}\}$$
$$= (\neg\, (P^f \;;\; ac \subseteq ac') \vdash (Q^t \wedge P^t) \;;\; ac \subseteq ac') \qquad \{\text{Strengthen postcondition}\}$$
$$\sqsupseteq (\neg\, (P^f \;;\; ac \subseteq ac') \vdash P^t \;;\; ac \subseteq ac') \qquad \{\text{Definition of } \mathbf{PBMH}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t)) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= \mathbf{A1}(\neg\, P^f \vdash P^t) \qquad \{\text{Definition of designs}\}$$
$$= \mathbf{A1}(P)$$

□

These results establish the suitability of **A1** as a healthiness condition. We tackle the commutativity of **A1** and **A0** in Section 5.2.3, where we define **A**. In the following section we show the closure properties satisfied by **A1**.

Closure properties

The function **A1** is closed with respect to disjunction. In fact it also distributes through disjunction. This is expected as **PBMH** is defined by the standard sequential composition operator that distributes over disjunction [1].

Law 5.2.8 (A1-distribute-disjunction)

$$\mathbf{A1}(P \vee Q) = \mathbf{A1}(P) \vee \mathbf{A1}(Q)$$

Proof.

$$\mathbf{A1}(P) \vee \mathbf{A1}(Q) \qquad \{\text{Definition of design}\}$$
$$= \mathbf{A1}(\neg\, P^f \vdash P^t) \vee \mathbf{A1}(\neg\, Q^f \vdash Q^t) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t)) \vee (\neg\, \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(Q^t))$$
$$\{\text{Disjunction of designs}\}$$

$$
\begin{aligned}
&= (\neg\, \mathbf{PBMH}(P^f) \land \neg\, \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(P^t) \lor \mathbf{PBMH}(Q^t)) \\
&\hfill \{\text{Propositional calculus}\} \\
&= (\neg\, (\mathbf{PBMH}(P^f) \lor \mathbf{PBMH}(Q^f)) \vdash \mathbf{PBMH}(P^t) \lor \mathbf{PBMH}(Q^t)) \\
&\hfill \{\text{Disjunction closure of } \mathbf{PBMH} \text{ (Law D.3.1)}\} \\
&= (\neg\, (\mathbf{PBMH}(P^f \lor Q^f)) \vdash \mathbf{PBMH}(P^t \lor Q^t)) && \{\text{Definition of } \mathbf{A1}\} \\
&= \mathbf{A1}(\neg\, (P^f \lor Q^f) \vdash P^t \lor Q^t) && \{\text{Propositional calculus}\} \\
&= \mathbf{A1}(\neg\, P^f \land \neg\, Q^f \vdash P^t \lor Q^t) && \{\text{Disjunction of designs}\} \\
&= \mathbf{A1}(\neg\, P^f \vdash P^t) \lor (\neg\, Q^f \vdash Q^t) && \{\text{Definition of design}\} \\
&= \mathbf{A1}(P \lor Q)
\end{aligned}
$$

□

Law 5.2.9 (A-closure-disjunction) *Provided P and Q are* **A1***-healthy.*

$$\mathbf{A1}(P \lor Q) = P \lor Q$$

Proof.

$$
\begin{aligned}
&\mathbf{A1}(P \lor Q) = P \lor Q && \{\text{Law 5.2.8}\} \\
&= \mathbf{A1}(P) \lor \mathbf{A1}(Q) && \{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A1}\text{-healthy}\} \\
&= P \lor Q
\end{aligned}
$$

□

This concludes our discussion regarding the properties observed by **A1**. In the following section we discuss the functional composition of **A0** and **A1**.

5.2.3 A

The proposed theory of designs is characterised by the two healthiness conditions **A0** and **A1**. The order in which these functions are composed is important since they do not always necessarily commute. In order to see the reason behind this consider the following counter-example.

Counter-example 1

$$\mathbf{A0} \circ \mathbf{A1}(\mathit{true} \vdash ac' = \emptyset) \qquad \{\text{Definition of } \mathbf{A1}\}$$

$$= \mathbf{A0}(\neg\,(\mathit{false}\ ;\ ac \subseteq ac') \vdash ac' = \emptyset\ ;\ ac \subseteq ac')$$
$$\{\text{Definition of sequential composition}\}$$
$$= \mathbf{A0}(\neg\,(\mathit{false} \land \exists\, ac_0 \bullet ac_0 \subseteq ac') \vdash \exists\, ac_0 \bullet ac_0 = \emptyset \land ac_0 \subseteq ac')$$
$$\{\text{One-point rule and predicate calculus}\}$$
$$= \mathbf{A0}(\mathit{true} \vdash \mathit{true}) \qquad \{\text{Definition of } \mathbf{A0}\}$$
$$= \mathbf{A0}(\mathit{true} \vdash ac' \neq \emptyset)$$

$$\mathbf{A1} \circ \mathbf{A0}(\mathit{true} \vdash ac' = \emptyset) \qquad \{\text{Definition of } \mathbf{A0}\}$$
$$= \mathbf{A1}(\mathit{true} \vdash ac' = \emptyset \land ac' \neq \emptyset) \qquad \{\text{Predicate calculus}\}$$
$$= \mathbf{A1}(\mathit{true} \vdash \mathit{false}) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= (\neg\,(\mathit{false}\ ;\ ac \subseteq ac') \vdash \mathit{false}\ ;\ ac \subseteq ac')$$
$$\{\text{Definition of sequential composition}\}$$
$$= (\mathit{true} \vdash \mathit{false})$$

In this example we apply the healthiness conditions to an unhealthy design whose postcondition requires non-termination: $ac' = \emptyset$. In the first case **A1** changes the postcondition into *true*, followed by the application of **A0**. While in the second case, **A0** is applied in the first place, making the postcondition *false*, a predicate that satisfies **PBMH**. The resulting predicate conforms to the definition of **Miracle**. Thus the functions do not always commute.

If instead we consider healthy predicates, then we can ensure that **A0** and **A1** commute. The following Law 5.2.10 establishes this result for predicates that are **A1** healthy. In fact the only requirement is for the postcondition, P^t to satisfy **PBMH**.

Law 5.2.10 (A0-A1-commutative) *Provided P^t satisfies* **PBMH**.

$$\mathbf{A0} \circ \mathbf{A1}(P) = \mathbf{A1} \circ \mathbf{A0}(P)$$

Proof.

$$\mathbf{A0} \circ \mathbf{A1}(P) \qquad \{\text{Definition of design}\}$$
$$= \mathbf{A0} \circ \mathbf{A1}(\neg\, P^f \vdash P^t) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= \mathbf{A0}(\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t)) \qquad \{\text{Law 5.2.1}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$$
$$\{ac' \neq \emptyset \text{ satisfies } \mathbf{PBMH} \text{ (Lemma D.4.5)}\}$$

$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land \mathbf{PBMH}(ac' \neq \emptyset))$$
$$\{\text{Closure of } \mathbf{PBMH} \text{ w.r.t. conjunction (Law D.3.2)}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(ac' \neq \emptyset)))$$
$$\{ac' \neq \emptyset \text{ satisfies } \mathbf{PBMH} \text{ (Lemma D.4.5)}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(\mathbf{PBMH}(P^t) \land ac' \neq \emptyset))$$
$$\{\text{Assumption: } P^t \text{ satisfies } \mathbf{PBMH}\}$$
$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t \land ac' \neq \emptyset)) \qquad \{\text{Definition of } \mathbf{A1}\}$$
$$= \mathbf{A1}(\neg\, P^f \vdash P^t \land ac' \neq \emptyset) \qquad \{\text{Definition of } \mathbf{A0}\}$$
$$= \mathbf{A1} \circ \mathbf{A0}(\neg\, P^f \vdash P^t) \qquad \{\text{Definition of design}\}$$
$$= \mathbf{A1} \circ \mathbf{A0}(P)$$

□

Following this discussion it is safe to introduce the definition of **A** as the functional composition of **A1** followed by **A0**.

Definition 54 (A)

$$\mathbf{A}(P) \mathrel{\widehat=} \mathbf{A0} \circ \mathbf{A1}(P)$$

Law 5.2.10 establishes that once the postcondition of P satisfies **PBMH** then the functions commute. Therefore by functionally composing first **A1** we guarantee that this is always the case. In the following section we explore some of the basic properties of **A** as expected of a healthiness condition.

Properties

In the following laws we prove that **A** is idempotent, monotonic and that it commutes with **H1** ∘ **H2**. These results establish the suitability of **A** as a healthiness condition for a theory of designs.

Law 5.2.11 (A-idempotent)

$$\mathbf{A} \circ \mathbf{A}(P) = \mathbf{A}(P)$$

Proof.

$$\mathbf{A} \circ \mathbf{A}(P) \qquad \{\text{Definition of } \mathbf{A} \text{ twice}\}$$

$= \mathbf{A0} \circ \mathbf{A1} \circ \mathbf{A0} \circ \mathbf{A1}(P)$

$\{\text{Law 5.2.10 and } \mathbf{A1}(P) \text{ ensures } P^t \text{ satisfies } \mathbf{PBMH}\}$

$= \mathbf{A0} \circ \mathbf{A0} \circ \mathbf{A1} \circ \mathbf{A1}(P)$

$\{\mathbf{A0}\text{-idempotent (Law 5.2.2) and } \mathbf{A1}\text{-idempotent (Law 5.2.6)}\}$

$= \mathbf{A0} \circ \mathbf{A1}(P)$ $\{\text{Definition of } \mathbf{A}\}$

$= \mathbf{A}(P)$

□

The proof of Law 5.2.11 relies on the fact that once $\mathbf{A1}(P)$ is applied, then P^t is guaranteed to satisfy **PBMH**. In turn this means that **A0** and **A1** commute according to Law 5.2.10. Finally both idempotents allow us to establish that the result of applying **A** twice is indeed **A**.

Law 5.2.12 (A-monotonic)

$$P \sqsubseteq Q \Rightarrow \mathbf{A}(P) \sqsubseteq \mathbf{A}(Q)$$

Proof. Follows from **A0**-monotonic (Law 5.2.3) and **A1**-monotonic (Law 5.2.7).
□

As expected, the function **A** is monotonic as established by Law 5.2.12. This follows from the monotonicity of both **A0** and **A1**.

Law 5.2.13 (A-H-commutative)

$$\mathbf{H1} \circ \mathbf{H2} \circ \mathbf{A}(P) = \mathbf{A} \circ \mathbf{H1} \circ \mathbf{H2}(P)$$

Proof.

$\mathbf{H1} \circ \mathbf{H2} \circ \mathbf{A}$ $\{\text{Definition of } \mathbf{A}\}$

$= \mathbf{H1} \circ \mathbf{H2}(\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$

$\{\text{Property of designs}\}$

$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$ $\{\text{Definition of } \mathbf{A}\}$

$= \mathbf{A}(\neg\, P^f \vdash P^t)$ $\{\text{Definition of } \mathbf{H1} \circ \mathbf{H2}\}$

$= \mathbf{A} \circ \mathbf{H1} \circ \mathbf{H2}(P)$

□

The healthiness condition of our theory is $\mathbf{H} \circ \mathbf{A}$. Since $\mathbf{H}$ and $\mathbf{A}$ commute, and $\mathbf{H}$ and $\mathbf{A}$ are idempotents, so is $\mathbf{H} \circ \mathbf{A}$ [1]. Furthermore, monotonicity also follows from monotonicity of $\mathbf{H}$ and $\mathbf{A}$.

This concludes our discussion of the healthiness conditions of the theory of designs with angelic nondeterminism. The designs of interest are characterised as fixed points of $\mathbf{A}$, an idempotent and monotonic function.

5.3 Relationship with the binary multirelational model

In this section we prove that the predicative model of $\mathbf{A}$-healthy designs is isomorphic to the binary multirelational model presented in Chapter 4. As mentioned previously, this allows us to establish the correspondence of the healthiness conditions and operators of both models.

In order to do so, we define a pair of linking functions: $bmb2d$, that maps from binary multirelations to predicates, and $d2bmb$ that maps in the opposite direction. The latter is defined in the following Section 5.3.1 while the former is defined in Section 5.3.2. Finally, in Section 5.3.3 the isomorphism is established by proving that both functions form a bijection.

5.3.1 From designs to binary multirelations ($d2bmb$)

The first linking function of interest is $d2bmb$. It maps from $\mathbf{A}$-healthy designs into relations of type $BM_\perp$. It is defined as follows.

Definition 55 (d2bmb) *Provided P is a design.*

$$d2bmb : \mathbf{A} \nrightarrow BM_\perp$$

$$d2bmb(P) \mathrel{\widehat=} \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \quad \left| \begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'] \land \perp \notin ss) \\ \lor \\ (P^f[ss \setminus \{\perp\}/ac'] \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

For a given design $P = (\neg\, P^f \vdash P^t)$, the set construction of $d2bmb(P)$ is split into two disjuncts.

In the first disjunction we consider the case where P is guaranteed to terminate, with ok and ok' both being substituted for $true$. The resulting set of final states ss, for which termination is required ($\perp \notin ss$) is obtained by substituting ss for ac' in P.

The second disjunct considers the case where ok is also $true$, but ok' is $false$. This corresponds to the situation where P does not terminate. In this case, the set of final states is obtained by substituting $ss \setminus \{\perp\}$ for ac' and requiring $\perp$ to be in the set of final states ss.

As a consequence of P satisfying **H2**, we ensure that if there is some set of final states captured by the second disjunct with $\perp$, then there is also an equivalent set of final states without $\perp$ that is captured by the first disjunct.

In the following Theorem 5.3.1 we prove that the application of $d2bmb$ to **A**-healthy designs yields relations that are **BMH0**-**BMH2**-healthy.

Theorem 5.3.1

$$\mathbf{bmh_{0,1,2}} \circ d2bmb(\mathbf{A}(P)) = d2bmb(\mathbf{A}(P))$$

Proof.

$$\begin{aligned}
&\mathbf{bmh_{0,1,2}} \circ d2bmb(\mathbf{A}(P)) && \{\text{Definition of } \mathbf{bmh_{0,1,2}}\} \\
&= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in d2bmb(\mathbf{A}(P)) \lor (s, ss_0 \cup \{\perp\}) \in d2bmb(\mathbf{A}(P))) \\ \land\, ((s, \{\perp\}) \in d2bmb(\mathbf{A}(P)) \Leftrightarrow (s, \emptyset) \in d2bmb(\mathbf{A}(P))) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\} \\
& && \{\text{Lemma C.1.4}\} \\
&= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in d2bmb(\mathbf{A}(P)) \lor (s, ss_0 \cup \{\perp\}) \in d2bmb(\mathbf{A}(P))) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\} \\
& && \{\text{Predicate calculus}\} \\
&= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in d2bmb(\mathbf{A}(P)) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in d2bmb(\mathbf{A}(P)) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right. \end{array} \right\} \\
& && \{\text{Lemmas C.1.2 and C.1.3}\}
\end{aligned}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \perp \notin ss)) \land ac_0 \subseteq ss \end{array} \right) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus and Lemma C.1.1}

$$= d2bmb(\mathbf{A}(P))$$

□

This result, whose proof relies on a number of lemmas proved in Appendix C.1, establishes the suitability of $d2bmb$ as a linking function.

In order to understand the result of applying $d2bmb$ better, we consider the following Example 14. It specifies a program that either assigns the value 1 to the sole program variable x and successfully terminates, or assigns the value 2 to x, in which case termination is not required.

Example 14

$$d2bmb((x \mapsto 2) \notin ac' \vdash (x \mapsto 1) \in ac') \qquad \{\text{Definition of } d2bmb\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((x \mapsto 2) \notin ac' \Rightarrow (x \mapsto 1) \in ac')[ss/ac'] \land \perp \notin ss) \\ \lor \\ (((x \mapsto 2) \in ac')[ss \setminus \{\perp\}/ac'] \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus and substitution}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((x \mapsto 2) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 1) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 2) \in (ss \setminus \{\perp\}) \land \perp \in ss) \end{array} \right. \end{array} \right\} \qquad \{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((x \mapsto 2) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 1) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 2) \in ss \land (x \mapsto 2) \notin \{\perp\} \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((x \mapsto 2) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 1) \in ss \land \perp \notin ss) \\ \lor \\ ((x \mapsto 2) \in ss \land \perp \in ss) \end{array} \right. \end{array} \right\} \qquad \{\text{Predicate calculus}\}
$$

$$
= \{s : State, ss : \mathbb{P}\, State_\perp \mid (x \mapsto 2) \in ss \lor ((x \mapsto 1) \in ss \land \perp \notin ss)\}
$$

$$
\{\text{Definition of } \sqcap_{BM_\perp} \text{ and } :=_{BM_\perp} \text{ and } :=_{BM}\}
$$

$$
= (x :=_{BM_\perp} 2) \sqcap_{BM_\perp} (x :=_{BM} 1)
$$

As expected, the function $d2bmb$ yields a program with the same behaviour specified using the binary multirelational model. It is the demonic choice over two assignments, one requires termination while the other does not.

5.3.2 From binary multirelations to designs ($bmb2d$)

The second linking function of interest is $bmb2d$ that maps binary multirelations to **A**-healthy predicates. Its definition is presented below.

Definition 56

$$
\begin{array}{l} bmb2d : BM_\perp \nrightarrow \mathbf{A} \\ bmb2d(B) \mathrel{\widehat=} ((s, ac' \cup \{\perp\}) \notin B \vdash (s, ac') \in B) \end{array}
$$

It is defined as a design, such that for a particular initial state s, the precondition requires $(s, ac' \cup \{\perp\})$ not to be in B, while the postcondition establishes that (s, ac') is in B. This definition can be expanded into a more intuitive representation according to the following Lemma 5.3.1.

Lemma 5.3.1

$$
bmb2d(B) = ok \Rightarrow \left(\begin{array}{l} ((s, ac') \in B \land \perp \notin ac' \land ok') \\ \lor \\ ((s, ac' \cup \{\perp\}) \in B \end{array} \right)
$$

Proof. Follows from the definition of design and type restriction on ac'. □

The behaviour of $bmb2d$ is split into two disjuncts. The first one considers the case where B requires termination, and hence $\perp$ is not part of the set of final states of the pair in B. While the second disjunct considers sets of final states that do not require termination, in which case ok' can be either *true* or *false*.

The following Theorem 5.3.2 establishes that $bmb2d(B)$ yields **A**-healthy designs provided that B is **BMH0-BMH2**-healthy.

Theorem 5.3.2 *Provided B satisfies* $\mathbf{bmh_{0,1,2}}$.

$$\mathbf{A} \circ bmb2d(B) = bmb2d(B)$$

Proof.

$$\begin{aligned}
& \mathbf{A} \circ bmb2d(B) && \{\text{Assumption: } B = \mathbf{bmh_{0,1,2}}(B) \text{ and Lemma C.2.2}\} \\
&= \mathbf{A}\left(\begin{array}{l} \neg\,((s, ac' \cup \{\bot\}) \in B \,;\, ac \subseteq ac') \\ \vdash \\ ((s, ac') \in B \,;\, ac \subseteq ac') \land (s, \emptyset) \notin B \end{array}\right) && \{\text{Lemma C.2.1}\} \\
&= \mathbf{A}\left(\begin{array}{l} \neg\,((s, ac' \cup \{\bot\}) \in B \,;\, ac \subseteq ac') \\ \vdash \\ ((s, ac') \in B \,;\, ac \subseteq ac') \land ac' \neq \emptyset \land (s, \emptyset) \notin B \end{array}\right) \\
& && \{\text{Definition of } \mathbf{PBMH}\} \\
&= \mathbf{A}\left(\begin{array}{l} \neg\,\mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B) \\ \vdash \\ \mathbf{PBMH}((s, ac') \in B) \land ac' \neq \emptyset \land (s, \emptyset) \notin B \end{array}\right) && \{\text{Definition of } \mathbf{A}\} \\
&= \left(\begin{array}{l} \neg\,(\mathbf{PBMH} \circ \mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B)) \\ \vdash \\ \mathbf{PBMH}(\mathbf{PBMH}((s, ac') \in B) \land ac' \neq \emptyset \land (s, \emptyset) \notin B) \land ac' \neq \emptyset \end{array}\right) \\
& && \{(\mathbf{PBMH}\text{-idempotent) Law D.1.1}\} \\
&= \left(\begin{array}{l} \neg\,\mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B) \\ \vdash \\ \mathbf{PBMH}(\mathbf{PBMH}((s, ac') \in B) \land ac' \neq \emptyset \land (s, \emptyset) \notin B) \land ac' \neq \emptyset \end{array}\right) \\
& && \{\text{Lemma D.4.5 and Lemma D.4.6}\} \\
&= \left(\begin{array}{l} \neg\,\mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B) \\ \vdash \\ \mathbf{PBMH}\left(\begin{array}{l} \mathbf{PBMH}((s, ac') \in B) \land \mathbf{PBMH}(ac' \neq \emptyset) \\ \land\, \mathbf{PBMH}((s, \emptyset) \notin B) \end{array}\right) \land ac' \neq \emptyset \end{array}\right) \\
& && \{\text{Law D.2.2}\}
\end{aligned}$$

$$= \left(\begin{array}{l} \neg \mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B) \\ \vdash \\ \left(\begin{array}{l} \mathbf{PBMH}((s, ac') \in B) \land \mathbf{PBMH}(ac' \neq \emptyset) \\ \land \mathbf{PBMH}((s, \emptyset) \notin B) \land ac' \neq \emptyset \end{array} \right) \end{array} \right)$$

{Lemma D.4.5 and Lemma D.4.6 and predicate calculus}

$$= \left(\begin{array}{l} \neg \mathbf{PBMH}((s, ac' \cup \{\bot\}) \in B) \\ \vdash \\ \mathbf{PBMH}((s, ac') \in B) \land ac' \neq \emptyset \land (s, \emptyset) \notin B \end{array} \right)$$

{Definition of **PBMH** and Lemma C.2.1}

$$= \left(\begin{array}{l} \neg ((s, ac' \cup \{\bot\}) \in B \ ; \ ac \subseteq ac') \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

{Assumption: $B = \mathbf{bmh_{0,1,2}}(B)$ and Lemma C.2.2}

$= bmb2d(B)$

□

This result confirms that $bmb2d$ is closed with respect to **A** when applied to relations that are **BMH0**-**BMH2**-healthy. This concludes our discussion of $bmb2d$. In the following section we focus our attention on the isomorphism.

5.3.3 Isomorphism: $d2bmb$ and $bmb2d$

In this section we show that $d2bmb$ and $bmb2d$ form a bijection. The following Theorem 5.3.3 establishes that $d2bmb$ is the inverse function of $bmb2d$ for relations that are **BMH0**-**BMH2**-healthy. While Theorem 5.3.4 establishes that $bmb2d$ is the inverse function of $d2bmb$ for designs that are **A**-healthy. Together these results establish that the models are isomorphic.

Theorem 5.3.3 *Provided B is* **BMH0**-**BMH2**-*healthy.*

$$d2bmb \circ bmb2d(B) = B$$

Proof.

$d2bmb \circ bmb2d(B)$ {Assumption: B is **BMH0**-**BMH2**-healthy}

$= d2bmb \circ bmb2d(\mathbf{bmh_{0,1,2}}(B))$ {Lemma C.2.2}

$$
= d2bmb \left(\begin{array}{l} \left(\begin{array}{l} \neg \, ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \land \\ \neg \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \in B \; ; \; ac \subseteq ac') \\ \land \\ (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right) \end{array} \right) \\ \vdash \\ ((s, ac') \in B \; ; \; ac \subseteq ac') \land (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right)
$$

$\{$Definition of $d2bmb\}$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\bot} \\ \left| \begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} \neg \, ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \land \\ \neg \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \in B \; ; \; ac \subseteq ac') \\ \land \\ (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right) \end{array} \right) \\ \Rightarrow \\ ((s, ac') \in B \; ; \; ac \subseteq ac') \land (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right) [ss/ac'] \\ \land \bot \notin ss \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \neg \left(\begin{array}{l} \neg \, ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \land \\ \neg \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \in B \; ; \; ac \subseteq ac') \\ \land \\ (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right) \end{array} \right) [ss \setminus \{\bot\}/ac'] \\ \land \bot \in ss \end{array} \right) \end{array} \right. \end{array} \right\}
$$

$\{$Subtitution$\}$

$$
= \left\{ \begin{array}{l}
s : State, ss : \mathbb{P}\, State_{\perp} \\
\left| \begin{array}{l}
\left(\begin{array}{l}
\left(\begin{array}{l}
\left(\begin{array}{l}
\neg\,((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\
\wedge \\
\neg \left(\begin{array}{l}
((s, ac' \cup \{\perp\}) \in B \;;\; ac \subseteq ss) \\
\wedge \\
(s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right)
\end{array} \right) \\
\Rightarrow \\
((s, ac') \in B \;;\; ac \subseteq ss) \wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right) \\
\wedge \perp \notin ss
\end{array} \right) \\
\vee \\
\left(\begin{array}{l}
\neg \left(\begin{array}{l}
\neg\,((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\
\wedge \\
\neg \left(\begin{array}{l}
((s, ac' \cup \{\perp\}) \in B \;;\; ac \subseteq (ss \setminus \{\perp\})) \\
\wedge \\
(s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right)
\end{array} \right) \\
\wedge \perp \in ss
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate claculus}

$$
= \left\{ \begin{array}{l}
s : State, ss : \mathbb{P}\, State_{\perp} \\
\left| \begin{array}{l}
\left(\begin{array}{l}
\neg \left(\begin{array}{l}
\left(\begin{array}{l}
\neg\,((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\
\wedge \\
\neg \left(\begin{array}{l}
((s, ac' \cup \{\perp\}) \in B \;;\; ac \subseteq ss) \\
\wedge \\
(s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right)
\end{array} \right) \\
\vee \\
((s, ac') \in B \;;\; ac \subseteq ss) \wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right) \\
\wedge \perp \notin ss
\end{array} \right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\
\vee \\
\left(\begin{array}{l}
((s, ac' \cup \{\perp\}) \in B \;;\; ac \subseteq (ss \setminus \{\perp\})) \\
\wedge \\
(s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array} \right)
\end{array} \right) \\
\wedge \perp \in ss
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B \land \perp \notin ss) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ (((s, ac') \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ ((s, \{\perp\}) \in B \land (s, \emptyset) \in B \land \perp \in ss) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq (ss \setminus \{\perp\})) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ (((s, ac') \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq (ss \setminus \{\perp\})) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Type: $\perp \notin ac$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ (((s, ac') \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ (((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B) \\ \lor \\ (((s, ac') \in B \ ; \ ac \subseteq ss) \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \in B \; ; \; ac \subseteq ss) \\ \vee \\ (((s, ac') \in B \; ; \; ac \subseteq ss) \wedge \perp \notin ss) \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\} \quad \{\text{Law B.2.2}\}
$$

$$
= \mathbf{bmh_{0,1,2}}(B) \quad \{\text{Assumption: } B \text{ is } \mathbf{BMH0\text{-}BMH2}\text{-healthy}\}
$$

$$
= B
$$

□

Theorem 5.3.4 *Provided P is $\mathbf{A}$-healthy.*

$$
bmb2d \circ d2bmb(P) = P
$$

Proof.

$$
bmb2d \circ d2bmb(P) \quad \{\text{Assumption: } P \text{ is } \mathbf{A}\text{-healthy}\}
$$

$$
= bmb2d \circ d2bmb(\mathbf{A}(P)) \quad \{\text{Definition of } bmb2d\}
$$

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, ac') \in d2bmb(\mathbf{A}(P)) \wedge \perp \notin ac' \wedge ok') \\ \vee \\ ((s, ac' \cup \{\perp\}) \in d2bmb(\mathbf{A}(P)) \wedge \perp \notin ac') \end{array} \right)
$$

$$
\{\text{Definition of } d2bmb(\mathbf{A}(P)) \text{ Lemma C.1.1}\}
$$

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\begin{array}{l}
(s, ac') \in \left\{\begin{array}{l}
s : State, ss : \mathbb{P}\, State_\perp \\
\left|\begin{array}{l}
\exists\, ac_0 : \mathbb{P}\, State \bullet \\
\left(\begin{array}{l}
P^f[ac_0/ac'] \\
\vee \\
(P^t[ac_0/ac'] \wedge \perp \notin ss \wedge ss \neq \emptyset)
\end{array}\right) \\
\wedge\, ac_0 \subseteq ss
\end{array}\right.
\end{array}\right\} \\
\wedge \perp \notin ac' \wedge ok'
\end{array}\right) \\
\vee \\
\left(\begin{array}{l}
(s, ac' \cup \{\perp\}) \in \left\{\begin{array}{l}
s : State, ss : \mathbb{P}\, State_\perp \\
\left|\begin{array}{l}
\exists\, ac_0 : \mathbb{P}\, State \bullet \\
\left(\begin{array}{l}
P^f[ac_0/ac'] \\
\vee \\
(P^t[ac_0/ac'] \wedge \perp \notin ss \wedge ss \neq \emptyset)
\end{array}\right) \\
\wedge\, ac_0 \subseteq ss
\end{array}\right.
\end{array}\right\} \\
\wedge \perp \notin ac'
\end{array}\right)
\end{array}\right)
$$

{Property of sets}

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\begin{array}{l}
\exists\, ac_0 : \mathbb{P}\, State \bullet \\
\left(\begin{array}{l}
P^f[ac_0/ac'] \\
\vee \\
(P^t[ac_0/ac'] \wedge \perp \notin ac' \wedge ac' \neq \emptyset)
\end{array}\right) \\
\wedge\, ac_0 \subseteq ac' \wedge \perp \notin ac' \wedge ok'
\end{array}\right) \\
\vee \\
\left(\begin{array}{l}
\exists\, ac_0 : \mathbb{P}\, State \bullet \\
\left(\begin{array}{l}
P^f[ac_0/ac'] \\
\vee \\
(P^t[ac_0/ac'] \wedge \perp \notin (ac' \cup \{\perp\}) \wedge (ac' \cup \{\perp\}) \neq \emptyset)
\end{array}\right) \\
\wedge\, ac_0 \subseteq (ac' \cup \{\perp\}) \wedge \perp \notin ac'
\end{array}\right)
\end{array}\right)
$$

{Property of sets and predicate calculus}

$$
= ok \Rightarrow \left(\begin{array}{l}
(\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac' \wedge \perp \notin ac' \wedge ok') \\
\vee \\
\left(\begin{array}{l}
\exists\, ac_0 : \mathbb{P}\, State \bullet \\
P^t[ac_0/ac'] \wedge ac_0 \subseteq ac' \wedge \perp \notin ac' \wedge ac' \neq \emptyset \wedge ok'
\end{array}\right) \\
\vee \\
(\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq (ac' \cup \{\perp\}) \wedge \perp \notin ac')
\end{array}\right)
$$

{Type restriction: $\perp \notin ac_0$ and property of sets}

$$= ok \Rightarrow \left(\begin{array}{l} (\exists ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ac' \land \bot \notin ac' \land ok') \\ \lor \\ \left(\begin{array}{l} \exists ac_0 : \mathbb{P}\, State \bullet \\ P^t[ac_0/ac'] \land ac_0 \subseteq ac' \land \bot \notin ac' \land ac' \neq \emptyset \land ok' \end{array}\right) \\ \lor \\ (\exists ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ac' \land \bot \notin ac') \end{array}\right)$$

{Predicate calculus}

$$= ok \Rightarrow \left(\begin{array}{l} \left(\begin{array}{l} \exists ac_0 : \mathbb{P}\, State \bullet \\ P^t[ac_0/ac'] \land ac_0 \subseteq ac' \land \bot \notin ac' \land ac' \neq \emptyset \land ok' \end{array}\right) \\ \lor \\ (\exists ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ac' \land \bot \notin ac') \end{array}\right)$$

{Type restriction: $\bot \notin ac'$}

$$= ok \Rightarrow \left(\begin{array}{l} (\exists ac_0 : \mathbb{P}\, State \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ac' \land ac' \neq \emptyset \land ok') \\ \lor \\ (\exists ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ac') \end{array}\right)$$

{Definition of sequential composition}

$$= ok \Rightarrow \left(\begin{array}{l} ((P^t \,;\, ac \subseteq ac') \land ac' \neq \emptyset \land ok') \\ \lor \\ (P^f \,;\, ac \subseteq ac') \end{array}\right) \qquad \text{\{Predicate calculus\}}$$

$$= (ok \land \neg\, (P^f \,;\, ac \subseteq ac')) \Rightarrow ((P^t \,;\, ac \subseteq ac') \land ac' \neq \emptyset \land ok')$$

{Definition of design}

$$= (\neg\, (P^f \,;\, ac \subseteq ac') \vdash (P^t \,;\, ac \subseteq ac') \land ac' \neq \emptyset)$$

{Definition of **PBMH**}

$$= (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \qquad \text{\{Definition of } \mathbf{A}\text{\}}$$

$$= \mathbf{A}(P) \qquad \text{\{Assumption: } P \text{ is } \mathbf{A}\text{-healthy\}}$$

$$= P$$

□

This result is of fundamental importance since it allows the same programs to be characterised using two different approaches. The binary multirelational model provides a set-theoretic approach, while the predicative theory proposed can easily be linked with other UTP theories of interest, namely the theory of reactive processes.

Furthermore, this dual approach enables us to justify the definition of certain aspects of our theory. This includes the healthiness conditions and the definition of certain operators such as sequential composition. The most intuitive and appropriate model can be used in each case. The results obtained in either model can then be related using the linking functions.

5.4 Refinement

The healthiness condition **A** can be understood as a function from the theory of designs into our theory. The theory of designs is a complete lattice [1]. Since **A** is idempotent and monotonic, a result in [1] establishes that such a function also yields a complete lattice. Therefore we can assert that the theory we propose is also a complete lattice under the implication ordering.

In the following Section 5.4.1 we define the extreme points of the lattice and explore basic properties. Finally, in Section 5.4.2 we prove that the refinement order of our theory corresponds to subset inclusion in the binary multirelational model of Chapter 4.

5.4.1 Extreme points

The extreme points of interest are **Abort** ($\bot_{\mathcal{D}\mathbf{ac}}$) and **Miracle** ($\top_{\mathcal{D}\mathbf{ac}}$) as expected of a theory of designs. In what follows we explore these two points and prove that they are **A**-healthy.

Abort

In the original theory of designs the bottom of the lattice is *true* and this can be expressed as a design, either ($false \vdash true$) or ($false \vdash false$). In the theory proposed in [14] the bottom of the lattice is also *true*. In the theory that we propose, the definition is also *true*.

Definition 57 (Abort)

$$\bot_{\mathcal{D}\mathbf{ac}} \mathrel{\widehat{=}} true$$

A program that aborts provides no guarantees about termination. Indeed it also leaves the set of angelic choices ac' unrestricted, so the empty set is a possibility. The following Law 5.4.1 establishes that *true* is an **A**-healthy predicate.

Law 5.4.1 ($\perp_{\mathcal{D}\mathbf{ac}}$-A-healthy)

$$\mathbf{A}(\perp_{\mathcal{D}\mathbf{ac}}) = \perp_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$\begin{aligned}
&\mathbf{A}(\perp_{\mathcal{D}\mathbf{ac}}) && \{\text{Definition of } \perp_{\mathcal{D}\mathbf{ac}}\}\\
&= \mathbf{A}(\mathit{true}) && \{\text{Property of designs}\}\\
&= \mathbf{A}(\mathit{false} \vdash \mathit{true}) && \{\text{Definition of } \mathbf{A}\}\\
&= (\neg\, \mathbf{PBMH}(\mathit{true}) \vdash \mathbf{PBMH}(\mathit{true}) \land ac' \neq \emptyset)\\
& && \{\text{Definition of } \mathbf{PBMH} \text{ and sequential composition}\}\\
&= \left(\begin{array}{l} \exists\, ac_0, ok_0 \bullet \mathit{true}[ac_0, ok_0/ac', ok'] \land ac_0 \subseteq ac' \\ \vdash \\ \exists\, ac_0, ok_0 \bullet \mathit{true}[ac_0, ok_0/ac', ok'] \land ac_0 \subseteq ac' \land ac' \neq \emptyset \end{array}\right)\\
& && \{\text{Property of substitution and propositional calculus}\}\\
&= (\mathit{false} \vdash ac' \neq \emptyset) && \{\text{Definition of design and propositional calculus}\}\\
&= \perp_{\mathcal{D}\mathbf{ac}}
\end{aligned}$$

□

This result establishes that $\perp_{\mathcal{D}\mathbf{ac}}$ is indeed a design in the theory.

Miracle

As explained previously, in the lattice of designs, the top of the lattice is **Miracle** ($\neg\, ok$). In the theory proposed in [14], the top is *false*. Since in our theory we include the observational variables ok and ok', the top is also $\neg\, ok$. This is shown in the following definition.

Definition 58 (Miracle)

$$\top_{\mathcal{D}\mathbf{ac}} \mathrel{\widehat=} \neg\, ok$$

The program $\top_{\mathcal{D}\mathbf{ac}}$ corresponds to the design specified as $(\mathit{true} \vdash \mathit{false})$. The following Law 5.4.2 establishes that $\neg\, ok$ is an **A**-healthy predicate.

Law 5.4.2 ($\top_{\mathcal{D}\mathbf{ac}}$-A-healthy)

$$\mathbf{A}(\top_{\mathcal{D}\mathbf{ac}}) = \top_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$\begin{aligned}
&\mathbf{A}(\top_{\mathcal{D}\mathbf{ac}}) && \{\text{Definition of } \top_{\mathcal{D}\mathbf{ac}}\}\\
&= \mathbf{A}(\neg\, ok) && \{\text{Property of designs}\}\\
&= \mathbf{A}(true \vdash false) && \{\text{Definition of } \mathbf{A}\}\\
&= (\neg\, \mathbf{PBMH}(false) \vdash \mathbf{PBMH}(false) \land ac' \neq \emptyset) && \{\text{Definition of } \mathbf{PBMH}\}\\
&= \left(\begin{array}{l} \exists\, ac_0, ok_0 \bullet false[ac_0, ok_0/ac', ok'] \land ac_0 \subseteq ac' \\ \vdash \\ (\exists\, ac_0, ok_0 \bullet false[ac_0, ok_0/ac', ok'] \land ac_0 \subseteq ac') \land ac' \neq \emptyset \end{array}\right)\\
& && \{\text{Property of substitution and propositional calculus}\}\\
&= (true \vdash false) && \{\text{Property of designs and propositional calculus}\}\\
&= \top_{\mathcal{D}\mathbf{ac}}
\end{aligned}$$

□

The program $\neg\ ok$ is the top of the lattice since it refines any **A**-healthy predicate. The proof for the bottom of the lattice, *true*, follows directly from the implication ordering. Thus we can establish the following property.

Law 5.4.3 (Ordering) *For any predicate P that is* **A***-healthy.*

$$\bot_{\mathcal{D}\mathbf{ac}} \sqsubseteq_{\mathcal{D}} P \sqsubseteq_{\mathcal{D}} \top_{\mathcal{D}\mathbf{ac}}$$

Proof. Follows from **A** monotonic, the definition of $\top_{\mathcal{D}\mathbf{ac}}$, $\bot_{\mathcal{D}\mathbf{ac}}$ and the implication ordering. □

This concludes our introduction to the extreme points of the theory. In the following Section 5.4.2 we establish the relationship between the refinement order of this theory and that of the binary multirelational model.

5.4.2 Relationship with binary multirelations

The development in Chapter 4 was meant to keep the model as similar as possible to the original model of binary multirelations. In Section 4.4 the refinement order was defined as subset inclusion, like in the original theory. The following Theorem 5.4.1 establishes that in fact the refinement order $\sqsubseteq_{BM_\perp}$ corresponds to the refinement order of designs $\sqsubseteq_{\mathcal{D}}$ in this theory.

Theorem 5.4.1 *Provided B_0 and B_1 are* **BMH0**-**BMH2**-*healthy.*

$$bmb2d(B_0) \sqsubseteq_{\mathcal{D}} bmb2d(B_1) \Leftrightarrow B_0 \sqsubseteq_{BM_\perp} B_1$$

Proof.

$$\begin{aligned}
& bmb2d(B_0) \sqsubseteq_{\mathcal{D}} bmb2d(B_1) && \{\text{Definition of } bmb2d\} \\
&= \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \notin B_0 \vdash (s, ac') \in B_0) \\ \sqsubseteq_{\mathcal{D}} \\ ((s, ac' \cup \{\perp\}) \notin B_1 \vdash (s, ac') \in B_1) \end{array}\right) && \{\text{Refinement of designs}\} \\
&= \left[\begin{array}{l} ((s, ac' \cup \{\perp\}) \notin B_0 \land (s, ac') \in B_1) \Rightarrow (s, ac') \in B_0 \\ \land \\ (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac' \cup \{\perp\}) \notin B_1 \end{array}\right] \\
& && \{\text{Predicate calculus}\} \\
&= \left[\begin{array}{l} \left(\begin{array}{l} (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac') \in B_0 \\ \lor \\ (s, ac') \in B_1 \Rightarrow (s, ac') \in B_0 \end{array}\right) \\ \land \\ (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac' \cup \{\perp\}) \notin B_1 \end{array}\right] \\
& && \{\text{Assumption: } B_0 \text{ is } \mathbf{BMH1}\text{-healthy}\} \\
&= \left[\begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac') \in B_0 \\ \land \\ (s, ac' \cup \{\perp\}) \in B_0 \Rightarrow (s, ac') \in B_0 \end{array}\right) \\ \lor \\ (s, ac') \in B_1 \Rightarrow (s, ac') \in B_0 \end{array}\right) \\ \land \\ (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac' \cup \{\perp\}) \notin B_1 \end{array}\right] \\
& && \{\text{Predicate calculus}\} \\
&= \left[\begin{array}{l} \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \notin B_0 \lor (s, ac' \cup \{\perp\}) \in B_0) \Rightarrow (s, ac') \in B_0 \\ \lor \\ (s, ac') \in B_1 \Rightarrow (s, ac') \in B_0 \end{array}\right) \\ \land \\ (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac' \cup \{\perp\}) \notin B_1 \end{array}\right] \\
& && \{\text{Predicate calculus}\} \\
&= \left[\begin{array}{l} (s, ac') \in B_0 \lor ((s, ac') \in B_1 \Rightarrow (s, ac') \in B_0) \\ \land \\ (s, ac' \cup \{\perp\}) \notin B_0 \Rightarrow (s, ac' \cup \{\perp\}) \notin B_1 \end{array}\right] \\
& && \{\text{Predicate calculus}\}
\end{aligned}$$

$$= \left[\begin{array}{l} (s, ac') \in B_1 \Rightarrow (s, ac') \in B_0 \\ \land \\ (s, ac' \cup \{\bot\}) \in B_1 \Rightarrow (s, ac' \cup \{\bot\}) \in B_0 \end{array} \right] \qquad \{\text{Lemma C.3.2}\}$$

$$= B_1 \subseteq B_0 \qquad \{\text{Definition of } \sqsubseteq_{BM_\bot}\}$$

$$= B_0 \sqsubseteq_{BM_\bot} B_1$$

□

It is reassuring to find that the refinement order in our theory of designs with angelic nondeterminism corresponds to subset ordering in the binary multirelational model. This is particularly important as it confirms the intuitive definition of the binary multirelational model.

5.5 Operators

In this section we define the main operators of the theory. This includes the definition of assignment in the following Section 5.5.1 and sequential composition in Section 5.5.2.

5.5.1 Assignment

Similarly to the theory of [14], the assignment operator is defined as follows.

Definition 59 (Assignment)

$$(x :=_{\mathcal{D}ac} e) \mathrel{\widehat=} (true \vdash s \oplus (x \mapsto e) \in ac')$$

It is defined by considering the design whose precondition is true, and whose postcondition establishes that every set of final states in ac' has a component where x is assigned the value of expression e. This is defined by considering the initial state s with the value of program variable x overridden.

5.5.2 Sequential composition

The most challenging aspect of the theory that we propose is its reliance on non-homogeneous relations. This means that sequential composition cannot simply be defined as relational composition like in other UTP theories. This is an unfortunate consequence. The definition we propose is layered upon that of the sequential composition operator defined in [14].

The definition of sequential composition for designs is defined by considering the auxiliary variables ok and ok' separately. The sequential composition of P and Q is defined as follows.

Definition 60 ($;_{\mathcal{D}}$-sequence)

$$P \ ;_{\mathcal{D}} Q \mathrel{\widehat=} \exists\, ok_0 \bullet P[ok_0/ok'] \ ;_{\mathcal{A}} Q[ok_0/ok]$$

This definition resembles relational composition with the notable difference that instead of conjunction we use another operator ($;_{\mathcal{A}}$) that handles the non-homogeneous alphabet of the relations. This operator corresponds to the definition of sequential composition as introduced in [14], bearing in mind that we have a slightly different alphabet. We present our definition.

Definition 61 ($;_{\mathcal{A}}$-sequence)

$$P \ ;_{\mathcal{A}} Q \mathrel{\widehat=} P[\{z : State \mid Q[z/s]\}/ac']$$

The operator $;_{\mathcal{A}}$ handles sequential composition in the relational world with angelic choices [14]. The composition can be understood as follows: a final state of $P \ ;_{\mathcal{A}} Q$ is a final state of Q that can be reached from a set of input states z of Q that is available to P as a set ac' of angelic choices.

Perhaps a more intuitive interpretation can be given by considering the operator $;_{\mathcal{A}}$ as back propagating the information concerning the valid final states, thus resembling a backtracking operation. In order to understand this definition we introduce the following example from [14].

Example 15

$$(s \oplus (x \mapsto 1)) \in ac' \ ;_{\mathcal{A}} \left(\begin{array}{l} (s \oplus (x \mapsto s.x + 1)) \in ac' \\ \land \\ (s \oplus (x \mapsto s.x + 2)) \in ac' \end{array}\right)$$

$$\{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\}$$

$$= (s \oplus (x \mapsto 1)) \in \left\{ z \,\middle|\, \left(\begin{array}{l} (s \oplus (x \mapsto s.x + 1)) \in ac' \\ \land \\ (s \oplus (x \mapsto s.x + 2)) \in ac' \end{array}\right)[z/s] \right\}$$

$$\{\text{Substitution}\}$$

$$= (s \oplus (x \mapsto 1)) \in \left\{ z \middle| \left(\begin{array}{l} (z \oplus (x \mapsto z.x + 1)) \in ac' \\ \wedge \\ (z \oplus (x \mapsto z.x + 2)) \in ac' \end{array} \right) \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} (s \oplus (x \mapsto 1) \oplus (x \mapsto (s \oplus (x \mapsto 1)).x + 1)) \in ac' \\ \wedge \\ (s \oplus (x \mapsto 1) \oplus (x \mapsto (s \oplus (x \mapsto 1)).x + 2)) \in ac' \end{array} \right)$$

$$\{\text{Record component}\}$$

$$= \left(\begin{array}{l} (s \oplus (x \mapsto 1) \oplus (x \mapsto 2)) \in ac' \\ \wedge \\ (s \oplus (x \mapsto 1) \oplus (x \mapsto 3)) \in ac' \end{array} \right) \qquad \{\text{Property of } \oplus\}$$

$$= (s \oplus (x \mapsto 2)) \in ac' \wedge (s \oplus (x \mapsto 3)) \in ac'$$

In this example we consider the sequential composition of a predicate that assigns 1 to x, followed by the conjunction of two predicates: one that increments the initial value of x by one, and the other by two. We observe that in [14] conjunction corresponds to angelic choice. If we take that interpretation, then the sequential composition yields two choices for assigning a value to x in ac' available to the angel.

In Appendix E we explore and prove the properties observed by the $\;;_{\mathcal{A}}$ operator. These results are important for proving and characterising the sequential composition of **A**-healthy designs. The follow theorem establishes this relationship.

Theorem 5.5.1 (Sequential composition) *Provided ok and ok′ are not free in P, Q, R and S, and that $\neg\, P$ and Q are* **PBMH***-healthy.*

$$\begin{array}{l} (P \vdash Q) \;;_{\mathcal{D}\mathbf{ac}} (R \vdash S) \\ = \\ (\neg\,(\neg\, P \;;_{\mathcal{A}} true) \wedge \neg\,(Q \;;_{\mathcal{A}} \neg\, R) \vdash Q \;;_{\mathcal{A}} (R \Rightarrow S)) \end{array}$$

Proof.

$$\begin{array}{ll} (P \vdash Q) \;;_{\mathcal{D}\mathbf{ac}} (R \vdash S) & \{\text{Definition of } ;_{\mathcal{D}\mathbf{ac}}\} \\ = \exists\, ok_0 \bullet (P \vdash Q)[ok_0/ok'] \;;_{\mathcal{A}} (R \vdash S)[ok_0/ok] & \{\text{Definition of design}\} \\ = \exists\, ok_0 \bullet ((ok \wedge P) \Rightarrow (Q \wedge ok'))[ok_0/ok'] \;;_{\mathcal{A}} ((ok \wedge R) \Rightarrow (S \wedge ok'))[ok_0/ok] & \\ & \{\text{Substitution and assumption}\} \end{array}$$

$$= \exists\, ok_0 \bullet ((ok \land P) \Rightarrow (Q \land ok_0)) \;\, ;_{\mathcal{A}} ((ok_0 \land R) \Rightarrow (S \land ok'))$$

{Case-analysis on ok_0 and predicate calculus}

$$= \left(\begin{array}{l} (((ok \land P) \Rightarrow Q) \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (\neg\, (ok \land P) \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} ((\neg\, ok \lor \neg\, P \lor Q) \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ ((\neg\, ok \lor \neg\, P) \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Right-distributivity of $;_{\mathcal{A}}$ (Law E.3.1)}

$$= \left(\begin{array}{l} (\neg\, ok \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (\neg\, P \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (Q \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (\neg\, ok \;\, ;_{\mathcal{A}}\, true) \lor (\neg\, P \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Law E.1.1 and predicate calculus}

$$= \left(\begin{array}{l} \neg\, ok \lor (\neg\, P \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (Q \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (\neg\, P \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Assumption: $\neg\, P$ is **PBMH**-healthy and Theorem E.8.1}

$$= \left(\begin{array}{l} \neg\, ok \lor (Q \;\, ;_{\mathcal{A}} (R \Rightarrow (S \land ok'))) \\ \lor \\ (\neg\, P \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Assumption: Q is **PBMH**-healthy and Lemma E.8.3}

$$= \left(\begin{array}{l} \neg\, ok \lor (Q \;\, ;_{\mathcal{A}} \neg\, R) \lor ((Q \;\, ;_{\mathcal{A}} (R \Rightarrow S)) \land ok') \\ \lor \\ (\neg\, P \;\, ;_{\mathcal{A}}\, true) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} (ok \land \neg\, (\neg\, P \;\, ;_{\mathcal{A}}\, true) \land \neg\, (Q \;\, ;_{\mathcal{A}} \neg\, R)) \\ \Rightarrow \\ ((Q \;\, ;_{\mathcal{A}} (R \Rightarrow S)) \land ok') \end{array} \right)$$

{Definition of design}

$$= \begin{pmatrix} \neg\,(\neg\, P \;\;;_{\mathcal{A}}\, true) \wedge \neg\,(Q \;\;;_{\mathcal{A}} \neg\, R) \\ \vdash \\ Q \;\;;_{\mathcal{A}}\,(R \Rightarrow S) \end{pmatrix}$$

□

The result obtained is very similar to that of sequential composition for the original theory of designs [1, 22], except for postcondition and the fact that we use the operator $\;;_{\mathcal{A}}$ instead of the sequential composition operator for relations [1]. The implication in the postcondition acts as a filter that removes final states of Q that fail to satisfy R. We consider the following example.

Example 16

$(true \vdash \{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{D}\mathbf{ac}}\, (s.x \neq 1 \vdash s \in ac')$

{Theorem 5.5.1}

$$= \begin{pmatrix} \neg\,(\neg\, true \;\;;_{\mathcal{A}}\, true) \wedge \neg\,((\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\, s.x = 1) \\ \vdash \\ (\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\,(s.x \neq 1 \Rightarrow s \in ac') \end{pmatrix}$$

{Predicate calculus}

$$= \begin{pmatrix} \neg\,(false \;\;;_{\mathcal{A}}\, true) \wedge \neg\,((\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\, s.x = 1) \\ \vdash \\ (\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\,(s.x \neq 1 \Rightarrow s \in ac') \end{pmatrix}$$

{Property of $\;;_{\mathcal{A}}$}

$$= \begin{pmatrix} \neg\, false \wedge \neg\,((\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\, s.x = 1) \\ \vdash \\ (\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\,(s.x \neq 1 \Rightarrow s \in ac') \end{pmatrix}$$

{Predicate calculus}

$$= \begin{pmatrix} \neg\,((\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\, s.x = 1) \\ \vdash \\ (\{x \mapsto 1\} \in ac' \wedge \{x \mapsto 2\} \in ac') \;\;;_{\mathcal{A}}\,(s.x \neq 1 \Rightarrow s \in ac') \end{pmatrix}$$

{Definition of $\;;_{\mathcal{A}}$ and substitution}

$$= \begin{pmatrix} \neg\,(\{x \mapsto 1\} \in \{s \mid s.x = 1\} \wedge \{x \mapsto 2\} \in \{s \mid s.x = 1\}) \\ \vdash \\ (\{x \mapsto 1\} \in \{s \mid s.x \neq 1 \Rightarrow s \in ac'\} \wedge \{x \mapsto 2\} \in \{s \mid s.x \neq 1 \Rightarrow s \in ac'\}) \end{pmatrix}$$

{Property of sets}

$$= \left(\begin{array}{l} \neg\,(\{x \mapsto 1\}.x = 1 \land \{x \mapsto 2\}.x = 1) \\ \vdash \\ (\{x \mapsto 1\}.x \neq 1 \Rightarrow \{x \mapsto 1\} \in ac') \land (\{x \mapsto 2\}.x \neq 1 \Rightarrow \{x \mapsto 2\} \in ac') \end{array} \right)$$

$$\{\text{Value of component } x\}$$

$$= \left(\begin{array}{l} \neg\,(1 = 1 \land 2 = 1) \\ \vdash \\ (1 \neq 1 \Rightarrow \{x \mapsto 1\} \in ac') \land (2 \neq 1 \Rightarrow \{x \mapsto 2\} \in ac') \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$= \left(\begin{array}{l} \mathit{true} \\ \vdash \\ (\mathit{false} \Rightarrow \{x \mapsto 1\} \in ac') \land (\mathit{true} \Rightarrow \{x \mapsto 2\} \in ac') \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$= (\mathit{true} \vdash \{x \mapsto 2\} \in ac')$$

In this case, there is an angelic choice between the assginment of the value 1 and 2 to the program variable x, sequentially composed with the program that aborts if x is 1 and that otherwhise behaves as *Skip*. The resulting design is just the assginment of 2 to x that avoids aborting. In the following section we establish closure of the sequential composition operator with respect to **A**.

If we consider designs that observe **H3**, we can simplify the result further as there are no dashed variables in the precondition.

$$(P \vdash Q) \;\;_{\mathcal{D}\mathbf{ac}}\, (R \vdash S) = (P \land (\neg\, R \;\;_{\mathcal{A}} \neg\, Q) \vdash (Q \;\;_{\mathcal{A}} (R \Rightarrow S)))$$

This is similar to the definition of sequential composition for designs where the precondition is a condition [22], except for the use of the operator $;_{\mathcal{A}}$ instead of sequential composition.

Closure

It is important that we establish closure of sequential composition ($;_{\mathcal{D}\mathbf{ac}}$) with respect to **A**. The following closure proof relies on laws established in Appendices D and E.

Law 5.5.1 ($;_{\mathcal{D}\mathbf{ac}}$-A-closure) *Provided P and Q are **A**-healthy and ok, ok' are not free in P and Q.*

$$\mathbf{A}(P \;;_{\mathcal{D}\mathbf{ac}}\, Q) = P \;;_{\mathcal{D}\mathbf{ac}}\, Q$$

Proof.

$$
\begin{aligned}
& P \ ;_{\mathcal{D}\mathbf{ac}} \ Q && \{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A}\text{-healthy}\} \\
&= \mathbf{A}(\neg\, P^f \vdash P^t) \ ;_{\mathcal{D}\mathbf{ac}} \ \mathbf{A}(\neg\, Q^f \vdash Q^t) && \{\text{Definition of } \mathbf{A}\} \\
&= \left(\begin{array}{l} (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\ \ ;_{\mathcal{D}\mathbf{ac}} \\ (\neg\, \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \end{array}\right) && \{\text{Definition of } \ ;_{\mathcal{D}\mathbf{ac}}\} \\
&= \exists\, ok_0 \bullet \left(\begin{array}{l} (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)[ok_0/ok'] \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset)[ok_0/ok] \end{array}\right) \\
& && \{\text{Definition of design}\} \\
&= \exists\, ok_0 \bullet \left(\begin{array}{l} ((ok \land \neg\, \mathbf{PBMH}(P^f)) \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset \land ok')[ok_0/ok'] \\ \ ;_{\mathcal{A}} \\ ((ok \land \neg\, \mathbf{PBMH}(Q^f)) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')[ok_0/ok] \end{array}\right) \\
& && \{\text{Substitution and assumption}\} \\
&= \exists\, ok_0 \bullet \left(\begin{array}{l} ((ok \land \neg\, \mathbf{PBMH}(P^f)) \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset \land ok_0) \\ \ ;_{\mathcal{A}} \\ ((ok_0 \land \neg\, \mathbf{PBMH}(Q^f)) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok') \end{array}\right) \\
& && \{\text{Case-analysis on } ok_0 \text{ and predicate calculus}\} \\
&= \left(\begin{array}{l} \left(\begin{array}{l} ((ok \land \neg\, \mathbf{PBMH}(P^f)) \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset)) \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')) \end{array}\right) \\ \lor \\ (\neg\, (ok \land \neg\, \mathbf{PBMH}(P^f)) \ ;_{\mathcal{A}} \ true) \end{array}\right) \\
& && \{\text{Predicate calculus}\} \\
&= \left(\begin{array}{l} \left(\begin{array}{l} (\neg\, ok \lor \mathbf{PBMH}(P^f) \lor (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset)) \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')) \end{array}\right) \\ \lor \\ ((\neg\, ok \lor \mathbf{PBMH}(P^f)) \ ;_{\mathcal{A}} \ true) \end{array}\right) \\
& && \{\text{Right-distributivity of } \ ;_{\mathcal{A}} \text{ (Law E.3.1)}\}
\end{aligned}
$$

$$= \left(\begin{array}{l} (\neg\, ok \ ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok'))) \\ \lor \\ (\mathbf{PBMH}(P^f) \ ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok'))) \\ \lor \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')) \end{array} \right) \\ \lor \\ (\neg\, ok \ ;_{\mathcal{A}} true) \lor (\mathbf{PBMH}(P^f) \ ;_{\mathcal{A}} true) \end{array} \right)$$

{Law E.1.1 and predicate calculus}

$$= \left(\begin{array}{l} \neg\, ok \\ \lor \\ (\mathbf{PBMH}(P^f) \ ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok'))) \\ \lor \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')) \end{array} \right) \\ \lor \\ (\mathbf{PBMH}(P^f) \ ;_{\mathcal{A}} true) \end{array} \right)$$

{Theorem E.8.1}

$$= \left(\begin{array}{l} \neg\, ok \\ \lor \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\ \ ;_{\mathcal{A}} \\ (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset \land ok')) \end{array} \right) \\ \lor \\ (\mathbf{PBMH}(P^f) \ ;_{\mathcal{A}} true) \end{array} \right)$$

{Lemma E.8.1}

$$
= \left(\begin{array}{l}
\neg\, ok \\
\lor \\
((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} \mathbf{PBMH}(Q^f)) \\
\lor \\
\left(\left(\begin{array}{l}
(\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\
\;;_{\mathcal{A}} \\
(\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t))
\end{array}\right) \land ac' \neq \emptyset \land ok'\right) \\
\lor \\
(\mathbf{PBMH}(P^f) \;\; ;_{\mathcal{A}} true)
\end{array}\right)
$$

{Predicate calculus}

$$
= \left(\begin{array}{l}
\left(\begin{array}{l}
ok \land \neg\, (\mathbf{PBMH}(P^f) \;\; ;_{\mathcal{A}} true) \\
\land \\
\neg\, ((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} \mathbf{PBMH}(Q^f))
\end{array}\right) \\
\Rightarrow \\
\left(\begin{array}{l}
((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t))) \\
\land \\
ac' \neq \emptyset \land ok'
\end{array}\right)
\end{array}\right)
$$

{Definition of design}

$$
= \left(\begin{array}{l}
\left(\begin{array}{l}
\neg\, (\mathbf{PBMH}(P^f) \;\; ;_{\mathcal{A}} true) \\
\land \\
\neg\, ((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} \mathbf{PBMH}(Q^f))
\end{array}\right) \\
\vdash \\
\left(\begin{array}{l}
((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t))) \\
\land \\
ac' \neq \emptyset
\end{array}\right)
\end{array}\right)
$$

{Definition of $\mathbf{A0}$}

$$
= \mathbf{A0}\left(\begin{array}{l}
\left(\begin{array}{l}
\neg\, (\mathbf{PBMH}(P^f) \;\; ;_{\mathcal{A}} true) \\
\land \\
\neg\, ((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} \mathbf{PBMH}(Q^f))
\end{array}\right) \\
\vdash \\
\left(((\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \;\; ;_{\mathcal{A}} (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t))) \right)
\end{array}\right)
$$

{Lemma D.4.5 and Laws D.3.1, D.3.2 and E.2.1}

$$= \mathbf{A0} \left(\begin{array}{l} \left(\begin{array}{l} \neg\, \mathbf{PBMH}(\mathbf{PBMH}(P^f) \;\;;_{\mathcal{A}}\, true) \\ \wedge \\ \neg\, \mathbf{PBMH}((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, \mathbf{PBMH}(Q^f)) \end{array} \right) \\ \vdash \\ \mathbf{PBMH} \left(\; ((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t)))\; \right) \end{array} \right)$$

{Predicate calculus and Law D.3.1}

$$= \mathbf{A0} \left(\begin{array}{l} \neg\, \mathbf{PBMH} \left(\begin{array}{l} (\mathbf{PBMH}(P^f) \;\;;_{\mathcal{A}}\, true) \\ \vee \\ ((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, \mathbf{PBMH}(Q^f)) \end{array} \right) \\ \vdash \\ \mathbf{PBMH} \left(\; ((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t)))\; \right) \end{array} \right)$$

{Definition of **A1** and predicate calculus}

$$= \mathbf{A0} \circ \mathbf{A1} \left(\begin{array}{l} \neg\, \mathbf{PBMH} \left(\begin{array}{l} \neg\, (\mathbf{PBMH}(P^f) \;\;;_{\mathcal{A}}\, true) \\ \wedge \\ \neg\, ((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, \mathbf{PBMH}(Q^f)) \end{array} \right) \\ \vdash \\ \left(\; ((\mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \;\;;_{\mathcal{A}}\, (\neg\, \mathbf{PBMH}(Q^f) \Rightarrow \mathbf{PBMH}(Q^t)))\; \right) \end{array} \right)$$

{Theorem 5.5.1}

$$= \mathbf{A0} \circ \mathbf{A1} \left(\begin{array}{l} (\neg\, \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \\ \;;_{\mathcal{D}\mathbf{ac}} \\ (\neg\, \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(Q^t) \wedge ac' \neq \emptyset) \end{array} \right)$$

{Definition of **A**}

$$= \mathbf{A}(\mathbf{A}(\neg\, P^f \vdash P^t) \;\;;_{\mathcal{D}\mathbf{ac}}\, \mathbf{A}(\neg\, Q^f \vdash Q^t))$$

{Assumption: P and Q are **A**-healthy}

$$= \mathbf{A}(P \;\;;_{\mathcal{D}\mathbf{ac}}\, Q)$$

□

This result establishes that $;_{\mathcal{D}\mathbf{ac}}$ is closed with respect to **A** provided both operands are also **A**-healthy.

In the following section we justify the definition of the sequential composition operator by proving that it corresponds to the definition of sequential composition for $BM_\perp$ relations.

Sequential composition in the binary multirelational model

The following Theorem 5.5.2 establishes that for designs that are **A**-healthy the definitions of sequential composition in both models correspond.

Theorem 5.5.2 *Provided P and Q are **A**-healthy.*

$$bmb2d(d2bmb(P) \;\;_{BM_\perp} d2bmb(Q)) = P \;\;_{\mathcal{D}\mathbf{ac}} Q$$

Proof.

$$\begin{aligned}
&bmb2d(d2bmb(P) \;;_{BM_\perp} d2bmb(Q)) && \{\text{Lemma C.2.9}\}\\
&= ok \Rightarrow \left(\begin{array}{l}((s, \{s_1 : State \mid (s_1, ac') \in d2bmb(Q)\}) \in d2bmb(P) \land \bot \notin ac' \land ok')\\ \lor\\ ((s, \{s_1 : State_\perp \mid true\}) \in d2bmb(P) \land \bot \notin ac')\\ \lor\\ ((s, \{s_1 : State \mid (s_1, ac' \cup \{\bot\}) \in d2bmb(Q)\}) \in d2bmb(P) \land \bot \notin ac')\end{array}\right)\\
& && \{\text{Lemma C.2.8}\}\\
&= ok \Rightarrow \left(\begin{array}{l}((\neg P^f \Rightarrow P^t)[\{s : State \mid (\neg Q^f \Rightarrow Q^t)\}/ac'] \land \bot \notin ac' \land ok')\\ \lor\\ ((s, \{s_1 : State_\perp \mid true\}) \in d2bmb(P) \land \bot \notin ac')\\ \lor\\ ((s, \{s_1 : State \mid (s_1, ac' \cup \{\bot\}) \in d2bmb(Q)\}) \in d2bmb(P) \land \bot \notin ac')\end{array}\right)\\
& && \{\text{Lemma C.2.7}\}\\
&= ok \Rightarrow \left(\begin{array}{l}((\neg P^f \Rightarrow P^t)[\{s : State \mid (\neg Q^f \Rightarrow Q^t)\}/ac'] \land \bot \notin ac' \land ok')\\ \lor\\ ((s, \{s_1 : State_\perp \mid true\}) \in d2bmb(P) \land \bot \notin ac')\\ \lor\\ ((\neg P^f \Rightarrow P^t)[\{s : State \mid Q^f\}/ac'] \land \bot \notin ac')\end{array}\right)\\
& && \{\text{Lemma C.2.4}\}\\
&= ok \Rightarrow \left(\begin{array}{l}((\neg P^f \Rightarrow P^t)[\{s : State \mid (\neg Q^f \Rightarrow Q^t)\}/ac'] \land \bot \notin ac' \land ok')\\ \lor\\ (P^f[\{s_1 : State \mid true\}/ac'] \land \bot \notin ac')\\ \lor\\ ((\neg P^f \Rightarrow P^t)[\{s : State \mid Q^f\}/ac'] \land \bot \notin ac')\end{array}\right)\\
& && \{\text{Assumption: } \bot \notin ac'\}
\end{aligned}$$

$$= ok \Rightarrow \left(\begin{array}{l} ((\neg P^f \Rightarrow P^t)[\{s : State \mid (\neg Q^f \Rightarrow Q^t)\}/ac'] \land ok') \\ \lor \\ (P^f[\{s_1 : State \mid true\}/ac']) \\ \lor \\ ((\neg P^f \Rightarrow P^t)[\{s : State \mid Q^f\}/ac']) \end{array}\right)$$

$$\{\text{Definition of } ;_{\mathcal{A}}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} (((\neg P^f \Rightarrow P^t) \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \\ \lor \\ ((\neg P^f \Rightarrow P^t) \ ;_{\mathcal{A}} Q^f) \end{array}\right)$$

$$\{\text{Predicate calculus and Law E.3.1}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} (((P^f \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \lor (P^t \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t))) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \\ \lor \\ (P^f \ ;_{\mathcal{A}} Q^f) \lor (P^t \ ;_{\mathcal{A}} Q^f) \end{array}\right)$$

$$\{\text{Predicate calculus}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} ((P^f \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ ((P^t \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \lor (P^f \ ;_{\mathcal{A}} Q^f) \lor (P^t \ ;_{\mathcal{A}} Q^f) \end{array}\right)$$

$$\{\text{Theorem E.8.1 under assumption that } P \text{ is } \mathbf{PBMH}\text{-healthy}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} ((P^t \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \lor (P^t \ ;_{\mathcal{A}} Q^f) \end{array}\right)$$

$$\{\text{Lemma E.8.3 under assumption that } P \text{ is } \mathbf{PBMH}\text{-healthy}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} (P^t \ ;_{\mathcal{A}} Q^f) \lor ((P^t \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \lor (P^t \ ;_{\mathcal{A}} Q^f) \end{array}\right)$$

$$\{\text{Predicate calculus}\}$$

$$= ok \Rightarrow \left(\begin{array}{l} (P^t \ ;_{\mathcal{A}} Q^f) \lor ((P^t \ ;_{\mathcal{A}} (\neg Q^f \Rightarrow Q^t)) \land ok') \\ \lor \\ (P^f \ ;_{\mathcal{A}} true) \end{array}\right)$$

$$\{\text{Predicate calculus}\}$$

$$= \left(\; (ok \land \neg\,(P^t \;\;;_{\mathcal{A}}\, Q^f) \land \neg\,(P^f \;\;;_{\mathcal{A}}\, true)) \Rightarrow ((P^t \;\;;_{\mathcal{A}}\, (\neg\, Q^f \Rightarrow Q^t)) \land ok') \; \right)$$

{Definition of design}

$$= \left(\begin{array}{l} \neg\,(P^t \;\;;_{\mathcal{A}}\, Q^f) \land \neg\,(P^f \;\;;_{\mathcal{A}}\, true) \\ \vdash \\ P^t \;\;;_{\mathcal{A}}\, (\neg\, Q^f \Rightarrow Q^t) \end{array} \right) \qquad \{\text{Theorem 5.5.1}\}$$

$$= (\neg\, P^f \vdash P^t) \;\;;_{\mathcal{D}\mathbf{ac}}\, (\neg\, Q^f \vdash Q^t)$$

{Assumption: P and Q are **A**-healthy designs}

$$= P \;\;;_{\mathcal{D}\mathbf{ac}}\, Q$$

□

Furthermore, together with the closure of $\;;_{\mathcal{D}\mathbf{ac}}$, this result enables us to ascertain the closure of $\;;_{BM_\perp}$.

This concludes our discussion of the definition of sequential composition. In what follows, we concentrate our attention on important properties observed by the sequential composition operator.

Skip

Similarly to the original theory of designs, we identify the **Skip** of the theory. We denote it by $I\!I_{\mathcal{D}\mathbf{ac}}$ and define it as follows.

Definition 62

$$I\!I_{\mathcal{D}\mathbf{ac}} \mathrel{\widehat=} (true \vdash s \in ac')$$

This is a design whose precondition is *true*, thus it is always applicable, and upon terminating it establishes that the input state s is in all sets of angelic choices ac'. The only results that can be guaranteed by the angel are those that are available in all demonic choices of the value of ac' that can be made. In this case, s is the only guarantee that we have, so the behaviour of $I\!I_{\mathcal{D}\mathbf{ac}}$ is to maintain the current state. In the following laws we prove that $I\!I_{\mathcal{D}\mathbf{ac}}$ is **A**-healthy and that it is the left-unit for sequential composition ($\;;_{\mathcal{D}\mathbf{ac}}$).

Law 5.5.2 ($I\!I_{\mathcal{D}}$-A-healthy)

$$\mathbf{A}(I\!I_{\mathcal{D}\mathbf{ac}}) = I\!I_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$
\begin{array}{ll}
\mathbf{A}(I\!I_{\mathcal{D}\mathbf{ac}}) & \{\text{Definition of } I\!I_{\mathcal{D}\mathbf{ac}}\}\\
= \mathbf{A}(true \vdash s \in ac') & \{\text{Definition of } \mathbf{A}\}\\
= (\neg\, \mathbf{PBMH}(\neg\, true) \vdash \mathbf{PBMH}(s \in ac') \land ac' \neq \emptyset) & \{\text{Lemma D.4.2}\}\\
= (\neg\, false \vdash \mathbf{PBMH}(s \in ac') \land ac' \neq \emptyset) & \{\text{Lemma D.4.3}\}\\
= (\neg\, false \vdash s \in ac' \land ac' \neq \emptyset) & \{\text{Property of sets and predicate calculus}\}\\
= (true \vdash s \in ac') & \{\text{Definition of } I\!I_{\mathcal{D}\mathbf{ac}}\}\\
= I\!I_{\mathcal{D}\mathbf{ac}} &
\end{array}
$$

□

Law 5.5.3 ($;_{\mathcal{D}\mathbf{ac}}$-left-unit) *Provided P is a design.*

$$I\!I_{\mathcal{D}\mathbf{ac}} \ ;_{\mathcal{D}\mathbf{ac}} P = P$$

Proof.

$$
\begin{array}{ll}
I\!I_{\mathcal{D}\mathbf{ac}} \ ;_{\mathcal{D}\mathbf{ac}} P & \{\text{Definition of } I\!I_{\mathcal{D}\mathbf{ac}} \text{ and design}\}\\
= (true \vdash s \in ac') \ ;_{\mathcal{D}\mathbf{ac}} (\neg\, P^f \vdash P^t) & \{\text{Theorem 5.5.1}\}\\
= (\neg\,(\neg\, true \ ;_{\mathcal{A}} true) \land \neg\,(s \in ac' \ ;_{\mathcal{A}} P^f) \vdash s \in ac' \ ;_{\mathcal{A}} (\neg\, P^f \Rightarrow P^t)) & \\
 & \{\text{Predicate calculus}\}\\
= (\neg\,(false \ ;_{\mathcal{A}} true) \land \neg\,(s \in ac' \ ;_{\mathcal{A}} P^f) \vdash s \in ac' \ ;_{\mathcal{A}} (\neg\, P^f \Rightarrow P^t)) & \\
 & \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\}\\
= (\neg\, false \land \neg\,(s \in ac' \ ;_{\mathcal{A}} P^f) \vdash s \in ac' \ ;_{\mathcal{A}} (\neg\, P^f \Rightarrow P^t)) & \\
 & \{\text{Predicate calculus}\}\\
= (\neg\,(s \in ac' \ ;_{\mathcal{A}} P^f) \vdash s \in ac' \ ;_{\mathcal{A}} (\neg\, P^f \Rightarrow P^t)) & \{\text{Law E.7.2}\}\\
= (\neg\, P^f \vdash (\neg\, P^f \Rightarrow P^t)) & \{\text{Predicate calculus}\}\\
= (\neg\, P^f \vdash P^t) & \{\text{Definition of design}\}\\
= P &
\end{array}
$$

□

These laws establish that $I\!I_{\mathcal{D}\mathbf{ac}}$ is indeed a suitable definition for **Skip**.

In what follows we establish that an **H3**-design in our theory requires the precondition not to mention dashed variables, as expected [1]. We first show the result of sequentially composing an **A**-healthy design P with $\mathit{II}_{\mathcal{D}\mathbf{ac}}$ in Law 5.5.4. Finally Law 5.5.5 establishes that $P \;;_{\mathcal{D}\mathbf{ac}} \mathit{II}_{\mathcal{D}\mathbf{ac}} = P$ restricts the precondition to a condition.

Law 5.5.4 (; $_{\mathcal{D}\mathbf{ac}}$-sequence-Skip) *Provided P is* **A***-healthy.*

$$P \;;_{\mathcal{D}} \mathit{II}_{\mathcal{D}\mathbf{ac}} = (\neg \exists ac' \bullet P^f \vdash P^t)$$

Proof.

$$\begin{aligned}
& P \;;_{\mathcal{D}\mathbf{ac}} \mathit{II}_{\mathcal{D}\mathbf{ac}} && \{\text{Definition of design and } \mathit{II}_{\mathcal{D}\mathbf{ac}}\}\\
&= (\neg P^f \vdash P^t) \;;_{\mathcal{D}\mathbf{ac}} (true \vdash s \in ac') && \{\text{Theorem 5.5.1}\}\\
&= (\neg (P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} false) \vdash P^t \;;_{\mathcal{A}} (true \Rightarrow s \in ac'))\\
& && \{\text{Predicate calculus}\}\\
&= (\neg (P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} false) \vdash P^t \;;_{\mathcal{A}} s \in ac')\\
& && \{\text{Assumption: } P \text{ is } \mathbf{A}\text{-healthy}\}\\
&= \left(\begin{array}{l} \neg (P^f \;;_{\mathcal{A}} true) \wedge \neg ((P^t \wedge ac' \neq \emptyset) \;;_{\mathcal{A}} false) \\ \vdash \\ (P^t \wedge ac' \neq \emptyset) \;;_{\mathcal{A}} (true \Rightarrow s \in ac') \end{array}\right)\\
& && \{\text{Right-distributivity of } ;_{\mathcal{A}} \text{ (Law E.4.1)}\}\\
&= \left(\begin{array}{l} \neg (P^f \;;_{\mathcal{A}} true) \wedge \neg ((P^t \;;_{\mathcal{A}} false) \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} false)) \\ \vdash \\ (P^t \;;_{\mathcal{A}} s \in ac') \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s \in ac') \end{array}\right)\\
& && \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\}\\
&= \left(\begin{array}{l} \neg (P^f \;;_{\mathcal{A}} true) \wedge \neg ((P^t \;;_{\mathcal{A}} false) \wedge \emptyset \neq \emptyset) \\ \vdash \\ (P^t \;;_{\mathcal{A}} s \in ac') \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s \in ac') \end{array}\right)\\
& && \{\text{Property of sets and predicate calculus}\}\\
&= (\neg (P^f \;;_{\mathcal{A}} true) \vdash (P^t \;;_{\mathcal{A}} s \in ac') \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s \in ac'))\\
& && \{s \in ac' \text{ is right-unit of } ;_{\mathcal{A}} \text{ (Law E.7.3)}\}\\
&= (\neg (P^f \;;_{\mathcal{A}} true) \vdash P^t \wedge ac' \neq \emptyset) && \{\text{Law E.5.2}\}\\
&= (\neg \exists ac' \bullet P^f \vdash P^t \wedge ac' \neq \emptyset) && \{\text{Assumption: } P \text{ is } \mathbf{A}\text{-healthy}\}\\
&= (\neg \exists ac' \bullet P^f \vdash P^t)
\end{aligned}$$

□

Law 5.5.5 (H3- $;_{\mathcal{D}\mathbf{ac}}$) *Provided P is* **A***-healthy, it is* **H3***-healthy if, and only if, its precondition does not mention ac'.*

$$(P \;\; ;_{\mathcal{D}} \; I\!I_{\mathcal{D}\mathbf{ac}}) = P \Leftrightarrow (\exists\, ac' \bullet \neg\, P^f = \neg\, P^f)$$

Proof.

$$
\begin{aligned}
&(P \;\; ;_{\mathcal{D}\mathbf{ac}} \; I\!I_{\mathcal{D}\mathbf{ac}}) = P && \{\text{Assumption: } P \text{ is } \mathbf{A}\text{-healthy}\}\\
&\Leftrightarrow (P \;\; ;_{\mathcal{D}\mathbf{ac}} \; I\!I_{\mathcal{D}}) = (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset) && \{\text{Law 5.5.4}\}\\
&\Leftrightarrow (\neg\, \exists\, ac' \bullet P^f \vdash P^t \wedge ac' \neq \emptyset) = (\neg\, P^f \vdash P^t \wedge ac' \neq \emptyset)\\
& && \{\text{Equality of designs}\}\\
&\Leftrightarrow [(\neg\, \exists\, ac' \bullet P^f) = \neg\, P^f] && \{\text{Predicate calculus}\}\\
&\Leftrightarrow [(\exists\, ac' \bullet P^f) = P^f] && \{\text{Predicate calculus (Lemma C.3.1)}\}\\
&\Leftrightarrow [(\exists\, ac' \bullet \neg\, P^f) = \neg\, P^f]
\end{aligned}
$$

$\square$

These results show that we have a theory whose essential properties concerning sequential composition hold as in the original theory of designs [1].

Sequential composition and the extreme points

In this section we establish the results of sequentially composing a program with the extreme points of the lattice. As expected, we establish the same left-zero laws that hold in the original theory of designs [1].

The following Law 5.5.6 establishes that it is impossible to recover from an aborting program. Law 5.5.7 establishes that if a design is miraculous then sequentially composing it with another design does not change its behaviour.

Law 5.5.6

$$\bot_{\mathcal{D}\mathbf{ac}} \;\; ;_{\mathcal{D}\mathbf{ac}} \; P = \bot_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$
\begin{aligned}
&\bot_{\mathcal{D}} \;\; ;_{\mathcal{D}\mathbf{ac}} \; P && \{\text{Definition of } \bot_{\mathcal{D}}\}\\
&= \mathit{true} \;\; ;_{\mathcal{D}\mathbf{ac}} \; P && \{\text{Definition of } \;;_{\mathcal{D}\mathbf{ac}}\}\\
&= \exists\, ok_0 \bullet \mathit{true}[ok_0/ok'] \;\; ;_{\mathcal{A}} \; P[ok_0/ok]\\
& && \{\text{Case-split on } ok_0 \text{ and property of substitution}\}
\end{aligned}
$$

$= (true \;\; ;_{\mathcal{A}} P[true/ok]) \vee (true \;\; ;_{\mathcal{A}} P[false/ok])$ {Definition of $;_{\mathcal{A}}$}
$= true \vee true$ {Propositional calculus and definition of $\perp_{\mathbf{\mathcal{D}ac}}$}
$= \perp_{\mathbf{\mathcal{D}ac}}$

□

Law 5.5.7

$$\top_{\mathbf{\mathcal{D}ac}} \;\; ;_{\mathbf{\mathcal{D}ac}} P = \top_{\mathbf{\mathcal{D}ac}}$$

Proof.

$\top_{\mathbf{\mathcal{D}ac}} \;\; ;_{\mathbf{\mathcal{D}ac}} P$ {Definition of $\top_{\mathbf{\mathcal{D}ac}}$}
$= (\neg\, ok) \;\; ;_{\mathbf{\mathcal{D}ac}} P$ {Definition of $;_{\mathbf{\mathcal{D}ac}}$}
$= \exists\, ok_0 \bullet (\neg\, ok)[ok_0/ok'] \;\; ;_{\mathcal{A}} P[ok_0/ok]$
{Substitution and case-split on ok_0}
$= (\neg\, ok \;\; ;_{\mathcal{A}} P[true/ok]) \vee (\neg\, ok \;\; ;_{\mathcal{A}} P[false/ok])$
{Definition of $;_{\mathcal{A}}$ and substitution}
$= \neg\, ok$ {Definition of $\top_{\mathbf{\mathcal{D}ac}}$}
$= \top_{\mathbf{\mathcal{D}ac}}$

□

Both of these results are expected of a theory of designs [1]. This concludes our discussion of the main operators of the theory and their properties. In the following section we concentrate our attention on nondeterminism.

5.6 Demonic and angelic nondeterminism

In this section we explore the two types of nondeterminism operators supported by the theory: angelic and demonic choice. We first discuss demonic nondeterminism in Section 5.6.1 followed by angelic nondeterminism in Section 5.6.2. For each operator we establish its closure and the relationship with the corresponding operator in the theory of binary multirelations of Chapter 4. In addition, based on similar results established in the literature [13, 16, 20, 27], we state and prove certain properties of the operators.

5.6.1 Demonic choice

The intuition for the demonic choice in our theory is related to the possible ways of choosing a value for ac'. In general, this can be described using disjunction like in the original theory of designs [1].

Definition 63

$$P \sqcap_{\mathcal{D}\mathbf{ac}} Q \mathrel{\widehat{=}} P \lor Q$$

This corresponds to the greatest lower bound of the lattice. We consider the following example, where $\oplus$ is the overriding operator [28].

Example 17

$$\begin{aligned}
&(x := 1) \sqcap_{\mathcal{D}\mathbf{ac}} (x := 2) && \{\text{Definition of assignment}\}\\
&= (\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac') \sqcap_{\mathcal{D}\mathbf{ac}} (\mathit{true} \vdash s \oplus (x \mapsto 2) \in ac')\\
& && \{\text{Definition of } \sqcap_{\mathcal{D}\mathbf{ac}} \text{ and disjunction of designs}\}\\
&= (\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto 2) \in ac')
\end{aligned}$$

In this example we have at least two choices for the final value of ac': one has a state where x is 1 and the other has a state where x is 2. The demon can choose any set ac' satisfying either predicate. In this case, the angel is not guaranteed to be able to choose a particular final value for x, since there are no choices in the intersection of all possible choices of ac'.

Closure properties

The demonic choice operator is closed with respect to **A**, provided that both operands are also **A**-healthy. This result follows from the distributive property of **A** with respect to disjunction, as established by the following Law 5.6.1.

Law 5.6.1 (A-disjunction-distribute)

$$\mathbf{A}(P \lor Q) = \mathbf{A}(P) \lor \mathbf{A}(Q)$$

Proof.

$$\mathbf{A}(P \lor Q) \qquad \{\text{Definition of design}\}$$

$$= \mathbf{A}((\neg P^f \vdash P^t) \lor (\neg Q^f \vdash Q^t)) \qquad \{\text{Disjunction of designs}\}$$
$$= \mathbf{A}(\neg P^f \land \neg Q^f \vdash P^t \lor Q^t) \qquad \{\text{Predicate calculus}\}$$
$$= \mathbf{A}(\neg (P^f \lor Q^f) \vdash P^t \lor Q^t) \qquad \{\text{Definition of } \mathbf{A}\}$$
$$= (\neg \mathbf{PBMH}(P^f \lor Q^f) \vdash \mathbf{PBMH}(P^t \lor Q^t) \land ac' \neq \emptyset)$$
$$\{\text{Distributivity of } \mathbf{PBMH} \text{ w.r.t. disjunction Law D.2.1}\}$$
$$= \left(\begin{array}{l} \neg (\mathbf{PBMH}(P^f) \lor \mathbf{PBMH}(Q^f)) \\ \vdash \\ (\mathbf{PBMH}(P^t) \lor \mathbf{PBMH}(Q^t)) \land ac' \neq \emptyset \end{array} \right) \qquad \{\text{Predicate calculus}\}$$
$$= \left(\begin{array}{l} \neg \mathbf{PBMH}(P^f) \land \neg \mathbf{PBMH}(Q^f) \\ \vdash \\ (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \lor (\mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \end{array} \right)$$
$$\{\text{Disjunction of designs}\}$$
$$= \left(\begin{array}{l} (\neg \mathbf{PBMH}(P^f) \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset) \\ \lor \\ (\neg \mathbf{PBMH}(Q^f) \vdash \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \end{array} \right) \qquad \{\text{Definition of } \mathbf{A}\}$$
$$= \mathbf{A}(\neg P^f \vdash P^t) \lor \mathbf{A}(\neg Q^f \vdash Q^t)$$

□

Law 5.6.2 *Provided P and Q are $\mathbf{A}$-healthy.*

$$\mathbf{A}(P \sqcap_{\mathcal{D}\mathbf{ac}} Q) = P \sqcap_{\mathcal{D}\mathbf{ac}} Q$$

Proof.

$$\mathbf{A}(P \sqcap_{\mathcal{D}\mathbf{ac}} Q) \qquad \{\text{Definition of } \sqcap_{\mathcal{D}\mathbf{ac}} \text{ and Law 5.6.1}\}$$
$$= \mathbf{A}(P) \lor \mathbf{A}(Q) \qquad \{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A}\text{-healthy}\}$$
$$= P \sqcap_{\mathcal{D}\mathbf{ac}} Q$$

□

This concludes the proof for closure of $\sqcap_{\mathcal{D}\mathbf{ac}}$ with respect to $\mathbf{A}$.

Relationship with binary multirelations

The demonic choice operator ($\sqcap_{\mathcal{D}\mathbf{ac}}$) corresponds exactly to the demonic choice operator ($\sqcap_{BM_\perp}$) of the binary multirelational model. This result is established by the following Theorem 5.6.1.

Theorem 5.6.1

$$bmb2p(B_0 \sqcap_{BM_\perp} B_1) = bmb2p(B_0) \sqcap_{\mathcal{D}\mathbf{ac}} bmb2p(B_1)$$

Proof.

$$\begin{aligned}
& bmb2p(B_0 \sqcap_{BM_\perp} B_1) && \{\text{Definition of } \sqcap_{BM_\perp}\} \\
& = bmb2p(B_0 \cup B_1) && \{\text{Definition of } bmb2p\} \\
& = ok \Rightarrow \left(\begin{array}{l} ((s, ac') \in (B_0 \cup B_1) \land \bot \notin ac' \land ok') \\ \lor \\ ((s, ac' \cup \{\bot\}) \in (B_0 \cup B_1) \land \bot \notin ac') \end{array} \right) && \{\text{Property of sets}\} \\
& = ok \Rightarrow \left(\begin{array}{l} (((s, ac') \in B_0 \lor (s, ac') \in B_1) \land \bot \notin ac' \land ok') \\ \lor \\ ((((s, ac' \cup \{\bot\}) \in B_0) \lor (s, ac' \cup \{\bot\}) \in B_1) \land \bot \notin ac') \end{array} \right) \\
& && \{\text{Propositional calculus}\} \\
& = ok \Rightarrow \left(\begin{array}{l} \left(\left(\begin{array}{l} ((s, ac') \in B_0 \land \bot \notin ac') \\ \lor \\ ((s, ac') \in B_1 \land \bot \notin ac') \end{array} \right) \land ok' \right) \\ \lor \\ ((s, ac' \cup \{\bot\}) \in B_0 \land \bot \notin ac') \\ \lor \\ ((s, ac' \cup \{\bot\}) \in B_1 \land \bot \notin ac') \end{array} \right) \\
& && \{\text{Propositional calculus}\} \\
& = \left(\begin{array}{l} \left(\begin{array}{l} ok \\ \land \\ \neg\, ((s, ac' \cup \{\bot\}) \in B_0 \land \bot \notin ac') \\ \land \\ \neg\, ((s, ac' \cup \{\bot\}) \in B_1 \land \bot \notin ac') \end{array} \right) \\ \Rightarrow \\ \left(\left(\begin{array}{l} ((s, ac') \in B_0 \land \bot \notin ac') \\ \lor \\ ((s, ac') \in B_1 \land \bot \notin ac') \end{array} \right) \land ok' \right) \end{array} \right) && \{\text{Property of designs}\}
\end{aligned}$$

$$= \left(\begin{array}{l} \left(\begin{array}{l} \neg\left((s, ac' \cup \{\bot\}) \in B_0 \land \bot \notin ac'\right) \\ \land \\ \neg\left((s, ac' \cup \{\bot\}) \in B_1 \land \bot \notin ac'\right) \end{array} \right) \\ \vdash \\ \left(\begin{array}{l} ((s, ac') \in B_0 \land \bot \notin ac') \\ \lor \\ ((s, ac') \in B_1 \land \bot \notin ac') \end{array} \right) \end{array} \right)$$

$$\{\text{Disjunction of designs and definition of } \sqcap_{\mathcal{D}\mathbf{ac}}\}$$

$$= \left(\begin{array}{l} (\neg\left((s, ac' \cup \{\bot\}) \in B_0 \land \bot \notin ac'\right) \vdash (s, ac') \in B_0 \land \bot \notin ac') \\ \sqcap_{\mathcal{D}\mathbf{ac}} \\ (\neg\left((s, ac' \cup \{\bot\}) \in B_1 \land \bot \notin ac'\right) \vdash (s, ac') \in B_1 \land \bot \notin ac') \end{array} \right)$$

$$\{\text{Definition of } bmb2p\}$$

$$= bmb2p(B_0) \sqcap_{\mathcal{D}\mathbf{ac}} bmb2p(B_1)$$

□

This result confirms the correspondence of demonic choice in both models. In the following section we focus our attention on its properties.

Properties

In general, and since demonic choice is the greatest lower bound, if presented with the possibility to abort ($\bot_{\mathcal{D}\mathbf{ac}}$), we expect the demon to choose the worst possible outcome as established by the following law.

Law 5.6.3 ($\sqcap$-$\bot_{\mathcal{D}\mathbf{ac}}$)

$$P \sqcap_{\mathcal{D}\mathbf{ac}} \bot_{\mathcal{D}\mathbf{ac}} = \bot_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$\begin{array}{ll} P \sqcap_{\mathcal{D}\mathbf{ac}} \bot_{\mathcal{D}\mathbf{ac}} & \{\text{Definition of } \sqcap_{\mathcal{D}\mathbf{ac}} \text{ and } \bot_{\mathcal{D}\mathbf{ac}}\} \\ = P \lor \mathit{true} & \{\text{Propositional calculus and definition of } \bot_{\mathcal{D}\mathbf{ac}}\} \\ = \bot_{\mathcal{D}\mathbf{ac}} & \end{array}$$

□

As observed in the original theory of designs [1], the sequential composition operator distributes through demonic choice, but only from the right as established by **??** and Law 5.6.4.

Law 5.6.4 ($\sqcap$-right-distributivity)

$$(P \sqcap_{\mathcal{D}\mathbf{ac}} Q) \;;_{\mathcal{D}\mathbf{ac}} R = (P \;;_{\mathcal{D}\mathbf{ac}} R) \sqcap_{\mathcal{D}\mathbf{ac}} (Q \;;_{\mathcal{D}\mathbf{ac}} R)$$

Proof.

$$
\begin{aligned}
&(P \;;_{\mathcal{D}\mathbf{ac}} R) \sqcap_{\mathcal{D}\mathbf{ac}} (Q \;;_{\mathcal{D}\mathbf{ac}} R) && \{\text{Definition of } ;_{\mathcal{D}\mathbf{ac}} \text{ and } \sqcap_{\mathcal{D}\mathbf{ac}}\}\\
&= (\exists\, ok_0 \bullet P[ok_0/ok'] \;;_{\mathcal{A}} R[ok_0/ok']) \vee (\exists\, ok_0 \bullet Q[ok_0/ok'] \;;_{\mathcal{A}} R[ok_0/ok])\\
& && \{\text{Propositional calculus}\}\\
&= \exists\, ok_0 \bullet (P[ok_0/ok'] \;;_{\mathcal{A}} R[ok_0/ok']) \vee (Q[ok_0/ok'] \;;_{\mathcal{A}} R[ok_0/ok])\\
& && \{\text{Right-distributivity of } ;_{\mathcal{A}} \text{ (Law E.3.1)}\}\\
&= \exists\, ok_0 \bullet ((P[ok_0/ok'] \vee Q[ok_0/ok']) \;;_{\mathcal{A}} R[ok_0/ok])\\
& && \{\text{Definition of } ;_{\mathcal{A}} \text{ and } \sqcap_{\mathcal{D}\mathbf{ac}}\}\\
&= (P \sqcap_{\mathcal{D}\mathbf{ac}} Q) \;;_{\mathcal{D}\mathbf{ac}} R
\end{aligned}
$$

□

These results conclude our discussion regarding the demonic choice operator and its properties. In the following section we focus our attention on the angelic choice operator and its respective properties.

5.6.2 Angelic choice

In the original theory of designs there is no angelic choice, and therefore the least upper bound of the lattice of designs, defined as conjunction, does not correspond to angelic choice. In other theories, such as in the predicate transformer model, angelic choice is defined exactly as the dual operator of demonic choice [13]. The same is applicable for the model of [14], where angelic choice is defined by conjunction, while demonic choice is disjunction. The definition adopted in our model is also conjunction of designs.

Definition 64 ($\sqcup_{\mathcal{D}\mathbf{ac}}$)

$$P \sqcup_{\mathcal{D}\mathbf{ac}} Q \mathrel{\widehat=} P \wedge Q$$

This definition is justified by the correspondence with the angelic choice operator of the binary multirelational model of Chapter 4.

To provide the intuition for this definition we consider the following Example 18.

Example 18

$$((x \mapsto 1) \notin ac' \vdash (x \mapsto 1) \in ac') \sqcup_{\mathcal{D}\mathbf{ac}} (true \vdash (x \mapsto 2) \in ac')$$

$$\{\text{Definition of } \sqcup_{\mathcal{D}\mathbf{ac}}\}$$

$$= \left(\begin{array}{l} (x \mapsto 1) \notin ac' \lor true \\ \vdash \\ \left(\begin{array}{l} (x \mapsto 1) \notin ac' \Rightarrow (x \mapsto 1) \in ac' \\ \land \\ true \Rightarrow (x \mapsto 2) \in ac' \end{array} \right) \end{array} \right) \qquad \{\text{Predicate calculus}\}$$

$$= (true \vdash (x \mapsto 1) \in ac' \land (x \mapsto 2) \in ac')$$

It considers the angelic choice between a design that assigns 1 to the only program variable x but does not necessarily terminate, and a design that assigns 2 to x but terminates. The result is a program that terminates and, for every set of final states, there is the possibility for the angel to choose the assignment of the value 1 or 2 to x.

Closure properties

Having defined angelic choice as the least upper bound operator, in the following Law 5.6.5 we prove that it is closed under **A**, provided that both operands are **A**-healthy.

Law 5.6.5 ($\sqcup_{\mathcal{D}\mathbf{ac}}$-A-closed) *Provided P and Q are* **A**-*healthy.*

$$\mathbf{A}(P \sqcup_{\mathcal{D}\mathbf{ac}} Q) = P \sqcup_{\mathcal{D}\mathbf{ac}} Q$$

Proof.

$$P \sqcup_{\mathcal{D}\mathbf{ac}} Q \qquad \{\text{Definition of design}\}$$

$$= (\neg P^f \vdash P^t) \sqcup_{\mathcal{D}\mathbf{ac}} (\neg Q^f \vdash Q^t) \qquad \{\text{Law A.2.6}\}$$

$$= (\neg P^f \lor \neg Q^f \vdash (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))$$

$$\{\text{Assumption: } P \text{ and } Q \text{ are } \mathbf{A}\text{-healthy}\}$$

$$= \left(\begin{array}{l} \neg \mathbf{PBMH}(P^f) \lor \neg \mathbf{PBMH}(Q^f) \\ \vdash \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \\ \lor \\ (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset \land \mathbf{PBMH}(Q^f)) \\ \lor \\ (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset \land \mathbf{PBMH}(Q^t)) \end{array} \right) \end{array} \right)$$

$\{$Predicate calculus$\}$

$$= \left(\begin{array}{l} \neg (\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^f)) \\ \vdash \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^t)) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^f)) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^t)) \end{array} \right) \land ac' \neq \emptyset \end{array} \right) \qquad \{\text{Law D.2.2}\}$$

$$= \left(\begin{array}{l} \neg \mathbf{PBMH}(\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^f)) \\ \vdash \\ \left(\begin{array}{l} \mathbf{PBMH}(\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^t)) \\ \lor \\ \mathbf{PBMH}(\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^f)) \\ \lor \\ \mathbf{PBMH}(\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^t)) \end{array} \right) \land ac' \neq \emptyset \end{array} \right)$$

$\{$Law D.2.1$\}$

$$= \left(\begin{array}{l} \neg \mathbf{PBMH}(\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^f)) \\ \vdash \\ \mathbf{PBMH} \left(\begin{array}{l} (\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^t)) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^f)) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^t)) \end{array} \right) \land ac' \neq \emptyset \end{array} \right)$$

$\{$Definition of $\mathbf{A}$ and predicate calculus$\}$

$$= \mathbf{A}\left(\begin{array}{l} \neg\, \mathbf{PBMH}(P^f) \lor \neg\, \mathbf{PBMH}(Q^f) \\ \vdash \\ \left(\begin{array}{l} (\mathbf{PBMH}(P^f) \land \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^f) \land ac' \neq \emptyset) \\ \lor \\ (\mathbf{PBMH}(P^t) \land \mathbf{PBMH}(Q^t) \land ac' \neq \emptyset) \end{array}\right) \end{array}\right)$$

$\{$Assumption: P and Q are $\mathbf{A}$-healthy$\}$

$= \mathbf{A}(\neg\, P^f \lor \neg\, Q^f \vdash (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))$ $\{$Law A.2.6$\}$

$= \mathbf{A}((\neg\, P^f \vdash P^t) \sqcup_{\mathcal{D}\mathbf{ac}} (\neg\, Q^f \vdash Q^t))$ $\{$Definition of design$\}$

$= \mathbf{A}(P \sqcup_{\mathcal{D}\mathbf{ac}} Q)$

□

This proof relies on properties of **PBMH** and on Law A.2.6 that provides a different result for the least upper bound of designs. Having established closure, in the following section we establish the correspondence with the binary multirelational model of Chapter 4.

Relationship with binary multirelations

In the following Theorem 5.6.2 we establish the correspondence of angelic choice in both models. This law requires the operands to be **BMH1**-healthy. This is satisfied by every binary multirelation that is **BMH0**-**BMH2**.

Theorem 5.6.2 *Provided* B_0 *and* B_1 *are* **BMH1**-*healthy.*

$$bmb2p(B_0 \sqcup_{BM_\perp} B_1) = bmb2p(B_0) \sqcup_{\mathcal{D}\mathbf{ac}} bmb2p(B_1)$$

Proof.

$bmb2p(B_0) \sqcup_{\mathcal{D}\mathbf{ac}} bmb2p(B_1)$ $\{$Definition of $bmb2p$ and $\sqcup_{\mathcal{D}\mathbf{ac}}\}$

$$= \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \notin B_0 \lor \perp \in ac' \vdash (s, ac') \in B_0 \land \perp \notin ac') \\ \vdash \\ ((s, ac' \cup \{\perp\}) \notin B_1 \lor \perp \in ac' \vdash (s, ac') \in B_1 \land \perp \notin ac') \end{array}\right)$$

$\{$Definition of $\sqcup$ for designs$\}$

$$= \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \notin B_0 \lor \bot \in ac' \lor (s, ac' \cup \{\bot\}) \notin B_1 \lor \bot \in ac') \\ \vdash \\ \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \notin B_0 \lor \bot \in ac') \Rightarrow ((s, ac') \in B_0 \land \bot \notin ac') \\ \land \\ ((s, ac' \cup \{\bot\}) \notin B_1 \lor \bot \in ac') \Rightarrow ((s, ac') \in B_1 \land \bot \notin ac') \end{array} \right) \end{array} \right)$$

{Propositional calculus}

$$= \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \notin B_0 \lor \bot \in ac' \lor (s, ac' \cup \{\bot\}) \notin B_1) \\ \vdash \\ \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \in B_0 \lor (s, ac') \in B_0) \\ \land \\ ((s, ac' \cup \{\bot\}) \in B_1 \lor (s, ac') \in B_1) \end{array} \right) \land \bot \notin ac' \end{array} \right)$$

{Assumption: B_0 and B_1 are **BMH1**-healthy}

$$= \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \notin B_0 \lor \bot \in ac' \lor (s, ac' \cup \{\bot\}) \notin B_1) \\ \vdash \\ \left(\begin{array}{l} (((s, ac' \cup \{\bot\}) \in B_0 \land (s, ac') \in B_0) \lor (s, ac') \in B_0) \\ \land \\ (((s, ac' \cup \{\bot\}) \in B_1 \land (s, ac') \in B_1) \lor (s, ac') \in B_1) \end{array} \right) \land \bot \notin ac' \end{array} \right)$$

{Propositional calculus: absorption law}

$$= \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \notin B_0 \lor \bot \in ac' \lor (s, ac' \cup \{\bot\}) \notin B_1) \\ \vdash \\ (s, ac') \in B_0 \land (s, ac') \in B_1 \land \bot \notin ac' \end{array} \right)$$

{Propositional calculus}

$$= \left(\begin{array}{l} \neg ((s, ac' \cup \{\bot\}) \in B_0 \land (s, ac' \cup \{\bot\}) \in B_1) \lor \bot \in ac' \\ \vdash \\ (s, ac') \in B_0 \land (s, ac') \in B_1 \land \bot \notin ac' \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} (s, ac' \cup \{\bot\}) \notin (B_0 \cap B_1) \lor \bot \in ac' \\ \vdash \\ (s, ac') \in (B_0 \cap B_1) \land \bot \notin ac' \end{array} \right)$$

{Definition of $bmb2p$}

$= bmb2p(B_0 \cap B_1)$ {Definition of $\sqcup_{BM_\bot}$}

$= bmb2p(B_0 \sqcup_{BM_\bot} B_1)$

$\square$

Having established the correspondence of the angelic choice operator in both models, in the following section we focus on its properties.

Properties

In general, and since angelic choice is the least upper bound, the angelic choice of a design P and the top of the lattice ($\top_{\mathcal{D}\mathbf{ac}}$) is also $\top_{\mathcal{D}\mathbf{ac}}$.

Law 5.6.6 *Provided P is a design.*

$$P \sqcup_{\mathcal{D}\mathbf{ac}} \top_{\mathcal{D}\mathbf{ac}} = \top_{\mathcal{D}\mathbf{ac}}$$

Proof.

$$
\begin{array}{ll}
P \sqcup_{\mathcal{D}\mathbf{ac}} \top_{\mathcal{D}\mathbf{ac}} & \{\text{Definition of } \sqcup_{\mathcal{D}\mathbf{ac}} \text{ and } \top_{\mathcal{D}\mathbf{ac}}\} \\
= P \land \neg\, ok & \{\text{Definition of design}\} \\
= (\neg\, P^f \vdash P^t) \land \neg\, ok & \{\text{Definition of design}\} \\
= ((ok \land \neg\, P^f) \Rightarrow (P^t \land ok')) \land \neg\, ok & \{\text{Predicate calculus}\} \\
= (\neg\, ok \lor P^f \lor (P^t \land ok')) \land \neg\, ok & \{\text{Predicate calculus: absorption law}\} \\
= \neg\, ok & \{\text{Definition of } \top_{\mathcal{D}\mathbf{ac}}\} \\
= \top_{\mathcal{D}\mathbf{ac}} &
\end{array}
$$

□

In this model, sequential composition does not necessarily distribute from the right nor from the left. In order to explain the intuition behind this we present the following Counter-example 2 for distribution from the left.

Counter-example 2

$$
\left(\begin{array}{l}
(true \vdash s \oplus (x \mapsto 1) \in ac') \\
\sqcap_{\mathcal{D}\mathbf{ac}} \\
(true \vdash s \oplus (x \mapsto -1) \in ac')
\end{array}\right) ;_{\mathcal{D}\mathbf{ac}} \left(\begin{array}{l}
(s.x = 1 \vdash false) \\
\sqcup \\
(s.x = -1 \vdash false)
\end{array}\right)
$$

$$\{\text{Assumption: } ;_{\mathcal{D}\mathbf{ac}} \text{ distributes over } \sqcap_{\mathcal{D}\mathbf{ac}}\}$$

$$= \left(\begin{array}{l} \left(\begin{array}{l} (\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac') \\ \sqcap_{\mathcal{D}\mathbf{ac}} \\ (\mathit{true} \vdash s \oplus (x \mapsto -1) \in ac') \end{array} \right) \ ;_{\mathcal{D}\mathbf{ac}} (s.x = 1 \vdash \mathit{false}) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} (\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac') \\ \sqcap_{\mathcal{D}\mathbf{ac}} \\ (\mathit{true} \vdash s \oplus (x \mapsto -1) \in ac') \end{array} \right) \ ;_{\mathcal{D}\mathbf{ac}} (s.x = -1 \vdash \mathit{false}) \end{array} \right)$$

{Definition of $\sqcap$}

$$= \left(\begin{array}{l} ((\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{D}\mathbf{ac}} (s.x = 1 \vdash \mathit{false})) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ ((\mathit{true} \vdash s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{D}\mathbf{ac}} (s.x = -1 \vdash \mathit{false})) \end{array} \right)$$

{Theorem 5.5.1}

$$= \left(\begin{array}{l} \left(\begin{array}{l} (\mathit{true} \ ;_{\mathcal{A}} \mathit{true}) \land \\ \neg ((s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq 1) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} (s.x = 1 \Rightarrow \mathit{false}) \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} (\mathit{true} \ ;_{\mathcal{A}} \mathit{true}) \land \\ \neg ((s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq -1) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} (s.x = -1 \Rightarrow \mathit{false}) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \left(\begin{array}{l} (\mathit{true} \ ;_{\mathcal{A}} \mathit{true}) \land \\ \neg ((s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq 1) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq 1 \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} (\mathit{true} \ ;_{\mathcal{A}} \mathit{true}) \land \\ \neg ((s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq -1) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \lor s \oplus (x \mapsto -1) \in ac') \ ;_{\mathcal{A}} s.x \neq -1 \end{array} \right) \end{array} \right)$$

{Property of $;_{\mathcal{A}}$ and propositional calculus}

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\left((s \oplus (x \mapsto 1) \in ac' \vee s \oplus (x \mapsto -1) \in ac') \;\;;_{\mathcal{A}}\; s.x \neq 1\right) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \vee s \oplus (x \mapsto -1) \in ac') \;\;;_{\mathcal{A}}\; s.x \neq 1 \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} \neg\left((s \oplus (x \mapsto 1) \in ac' \vee s \oplus (x \mapsto -1) \in ac') \;\;;_{\mathcal{A}}\; s.x \neq -1\right) \\ \vdash \\ (s \oplus (x \mapsto 1) \in ac' \vee s \oplus (x \mapsto -1) \in ac') \;\;;_{\mathcal{A}}\; s.x \neq -1 \end{array} \right) \end{array} \right)
$$

$$
\{\text{Definition of } ;_{\mathcal{A}} \text{ and subsitution}\}
$$

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\left(s \oplus (x \mapsto 1) \in \{z \mid z.x \neq 1\} \vee s \oplus (x \mapsto -1) \in \{z \mid z.x \neq 1\}\right) \\ \vdash \\ (s \oplus (x \mapsto 1) \in \{s \mid s.x \neq 1\} \vee s \oplus (x \mapsto -1) \in \{s \mid s.x \neq 1\}) \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} \neg\left(s \oplus (x \mapsto 1) \in \{z \mid z.x \neq -1\} \vee s \oplus (x \mapsto -1) \in \{z \mid z.x \neq -1\}\right) \\ \vdash \\ (s \oplus (x \mapsto 1) \in \{s \mid s.x \neq -1\} \vee s \oplus (x \mapsto -1) \in \{s \mid s.x \neq -1\}) \end{array} \right) \end{array} \right)
$$

$$
\{\text{Property of sets and predicate calculus}\}
$$

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\left(\neg\left(s \oplus (x \mapsto 1).x \neq 1\right) \vee \neg\left(s \oplus (x \mapsto -1).x \neq 1\right)\right) \\ \vdash \\ \mathit{true} \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} \neg\left(\neg\left(s \oplus (x \mapsto 1).x \neq -1\right) \vee \neg\left(s \oplus (x \mapsto -1).x \neq -1\right)\right) \\ \vdash \\ \mathit{true} \end{array} \right) \end{array} \right)
$$

$$
\{\text{Property of } \oplus\}
$$

$$
= \left(\begin{array}{l} (\neg(\neg \mathit{false} \vee \neg \mathit{true}) \vdash \mathit{true}) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ (\neg(\neg \mathit{true} \vee \neg \mathit{false}) \vdash \mathit{true}) \end{array} \right) \qquad \{\text{Propositional calculus}\}
$$

$$
\begin{array}{lr}
= (\mathit{false} \vdash \mathit{true}) \sqcup_{\mathcal{D}\mathbf{ac}} (\mathit{false} \vdash \mathit{true}) & \{\text{Property of } \sqcup_{\mathcal{D}\mathbf{ac}}\} \\
= (\mathit{false} \vdash \mathit{true}) & \{\text{Definition of design and propositional calculus}\} \\
= \mathit{true} & \{\text{Definitionf of } \bot_{\mathcal{D}\mathbf{ac}}\} \\
= \bot_{\mathcal{D}\mathbf{ac}} &
\end{array}
$$

This is a sequential composition. In the first program the precondition always holds and the program presents a choice to the demon. In this case, the demon can choose the set of final states, ac', by guaranteeing that either x is set to 1 or -1 in the final set of states ac'. The second program presents an angelic choice, but the precondition makes a restriction on the value of x in the initial state s: in either case, if the precondition is satisfied the program is $\top_{\mathcal{D}\mathbf{ac}}$, otherwise if no precondition can be satisfied, the program behaves as $\bot_{\mathcal{D}\mathbf{ac}}$.

It is expected that the angel will avoid $\bot_{\mathcal{D}\mathbf{ac}}$ if that it is possible. In this case, it is expected, since the angel can avoid aborting irrespective of the choice the demon makes before the angel. However, if we assume that the sequential composition operator $\,;_{\mathcal{D}\mathbf{ac}}$ left-distributes over angelic choice we get a different result as shown above.

In addition, sequential composition does not distribute from the right. We illustrate this problem in Counter-example 3. It is the sequential composition of two designs. The first design is the angelic choice between the program that assigns 2 to x, but may not terminate, and the program that always terminates but whose final set of states ac' is unrestricted, except that it cannot be the empty set. The second design is miraculous for $s.x = 2$ and for every other value of $s.x$ it aborts.

Counter-example 3

$$\left(\begin{array}{l} ((x \mapsto 2) \notin ac' \vdash (x \mapsto 2) \in ac') \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ (\mathit{true} \vdash ac' \neq \emptyset) \end{array}\right) \;;_{\mathcal{D}\mathbf{ac}} \left(\begin{array}{l} s.x = 2 \\ \vdash \\ s.x \neq 2 \land ac' \neq \emptyset \end{array}\right)$$

$$\{\text{Definition of } \sqcup_{\mathcal{D}\mathbf{ac}}\}$$

$$= \left(\begin{array}{l} (x \mapsto 2) \notin ac' \lor \mathit{true} \\ \vdash \\ \left(\begin{array}{l} (x \mapsto 2) \notin ac' \Rightarrow (x \mapsto 2) \in ac' \\ \land \\ \mathit{true} \Rightarrow ac' \neq \emptyset \end{array}\right) \end{array}\right) \;;_{\mathcal{D}\mathbf{ac}} \left(\begin{array}{l} s.x = 2 \\ \vdash \\ s.x \neq 2 \land ac' \neq \emptyset \end{array}\right)$$

$$\{\text{Predicate calculus}\}$$

$$= (\mathit{true} \vdash (x \mapsto 2) \in ac' \land ac' \neq \emptyset) \;;_{\mathcal{D}\mathbf{ac}} (s.x = 2 \vdash s.x \neq 2 \land ac' \neq \emptyset)$$

$$\{\text{Property of sets and predicate calculus}\}$$

$$= (\mathit{true} \vdash (x \mapsto 2) \in ac') \;;_{\mathcal{D}\mathbf{ac}} (s.x = 2 \vdash s.x \neq 2 \land ac' \neq \emptyset)$$

$$\{\text{Theorem 5.5.1}\}$$

$$= \left(\begin{array}{l} \neg\,(\mathit{false} \;\;;_{\mathcal{A}}\; \mathit{true}) \wedge \neg\,((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; s.x \neq 2) \\ \vdash \\ (x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; (s.x = 2 \Rightarrow (s.x \neq 2 \wedge ac' \neq \emptyset)) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \neg\,(\mathit{false} \;\;;_{\mathcal{A}}\; \mathit{true}) \wedge \neg\,((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; s.x \neq 2) \\ \vdash \\ (x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; s.x \neq 2) \end{array} \right)$$

{Definition of $;_{\mathcal{A}}$ and substitution}

$$= \left(\begin{array}{l} \neg\,\mathit{false} \wedge \neg\,((x \mapsto 2) \in \{z \mid z.x \neq 2\}) \\ \vdash \\ (x \mapsto 2) \in \{z \mid z.x \neq 2\} \end{array} \right) \qquad \text{\{Property of sets\}}$$

$$= \left(\begin{array}{l} \neg\,\mathit{false} \wedge \neg\,((x \mapsto 2).x \neq 2) \\ \vdash \\ (x \mapsto 2).x \neq 2 \end{array} \right) \qquad \text{\{Predicate calculus\}}$$

$$= (\neg\,(2 \neq 2) \vdash 2 \neq 2) \qquad \text{\{Predicate calculus\}}$$

$$= (\mathit{true} \vdash \mathit{false}) \qquad \text{\{Predicate calculus and definition of } \top_{\mathcal{D}\mathbf{ac}}\text{\}}$$

$$= \top_{\mathcal{D}\mathbf{ac}}$$

$$\neq$$

$$\left(\begin{array}{l} ((x \mapsto 2) \notin ac' \vdash (x \mapsto 2) \in ac') \;\;;_{\mathcal{D}\mathbf{ac}}\; (s.x = 2 \vdash s.x \neq 2 \wedge ac' \neq \emptyset) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ (\mathit{true} \vdash ac' \neq \emptyset) \;\;;_{\mathcal{D}\mathbf{ac}}\; (s.x = 2 \vdash s.x \neq 2 \wedge ac' \neq \emptyset) \end{array} \right)$$

{Theorem 5.5.1}

$$= \left(\begin{array}{l} \left(\begin{array}{l} \neg\,((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; \mathit{true}) \wedge \neg\,((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; s.x \neq 2) \\ \vdash \\ (x \mapsto 2) \in ac' \;\;;_{\mathcal{A}}\; (s.x = 2 \Rightarrow (s.x \neq 2 \wedge ac' \neq \emptyset)) \end{array} \right) \\ \sqcup_{\mathcal{D}\mathbf{ac}} \\ \left(\begin{array}{l} \neg\,(\mathit{false} \;\;;_{\mathcal{A}}\; \mathit{true}) \wedge \neg\,(ac' \neq \emptyset \;\;;_{\mathcal{A}}\; s.x \neq 2) \\ \vdash \\ ac' \neq \emptyset \;\;;_{\mathcal{A}}\; (s.x = 2 \Rightarrow (s.x \neq 2 \wedge ac' \neq \emptyset)) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\, ((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}} \mathit{true}) \wedge \neg\, ((x \mapsto 2) \in ac' \;\;;_{\mathcal{A}} s.x \neq 2) \\ \vdash \\ (x \mapsto 2) \in ac' \;\;;_{\mathcal{A}} s.x \neq 2 \end{array} \right) \\ \sqcup_{\mathbf{\mathcal{D}ac}} \\ \left(\begin{array}{l} \neg\, (\mathit{false} \;\;;_{\mathcal{A}} \mathit{true}) \wedge \neg\, (ac' \neq \emptyset \;\;;_{\mathcal{A}} s.x \neq 2) \\ \vdash \\ ac' \neq \emptyset \;\;;_{\mathcal{A}} s.x \neq 2 \end{array} \right) \end{array} \right)
$$

$\{$Definition of $\;;_{\mathcal{A}}$ and substitution$\}$

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\, ((x \mapsto 2) \in \{z \mid \mathit{true}\}) \wedge \neg\, ((x \mapsto 2) \in \{z \mid z.x \neq 2\}) \\ \vdash \\ (x \mapsto 2) \in \{z \mid z.x \neq 2\} \end{array} \right) \\ \sqcup_{\mathbf{\mathcal{D}ac}} \\ \left(\begin{array}{l} \neg\, \mathit{false} \wedge \neg\, (\{z \mid z.x \neq 2\} \neq \emptyset) \\ \vdash \\ \{z \mid z.x \neq 2\} \neq \emptyset \end{array} \right) \end{array} \right)
$$

$\{$Predicate calculus and property of sets$\}$

$$
= \left(\begin{array}{l} \left(\begin{array}{l} \neg\, \mathit{true} \wedge \neg\, (x \mapsto 2).x \neq 2 \\ \vdash \\ (x \mapsto 2).x \neq 2 \end{array} \right) \\ \sqcup_{\mathbf{\mathcal{D}ac}} \\ \left(\begin{array}{l} \neg\, \mathit{false} \wedge \neg\, \mathit{true} \\ \vdash \\ \mathit{true} \end{array} \right) \end{array} \right) \qquad \{\text{Predicate calculus}\}
$$

$= (\mathit{false} \vdash \mathit{false}) \sqcup_{\mathbf{\mathcal{D}ac}} (\mathit{false} \vdash \mathit{true})$

$\{$Predicate calculus and definition of $\perp_{\mathbf{\mathcal{D}ac}}\}$

$= \perp_{\mathbf{\mathcal{D}ac}} \sqcup_{\mathbf{\mathcal{D}ac}} \perp_{\mathbf{\mathcal{D}ac}}$ $\qquad$ $\{$Definition of $\sqcup_{\mathbf{\mathcal{D}ac}}$, $\perp_{\mathbf{\mathcal{D}ac}}$ and predicate calculus$\}$

$= \perp_{\mathbf{\mathcal{D}ac}}$

In the first case, the angelic choice is resolved first and the result is the program that always terminates and whose set of final states ac' has a state where x is assigned the value 2. Sequentially composing this with the second design results in a miracle ($\top_{\mathbf{\mathcal{D}ac}}$) as the only state available for angelic choice is where x has the value 2. However, this is precisely the case in which the design behaves miraculously.

In the second case, we assume that sequential composition distributes through angelic choice. In the resulting angelic choice there are two sequential compositions. In the first one, the result is $\perp_{\mathbf{\mathcal{D}ac}}$ as the first design may not terminate. While in the second, termination is guaranteed but any final set of states ($ac' \neq \emptyset$) may fail to satisfy the precondition $s.x = 2$, in which case the design aborts.

Finally, the demonic and angelic choice operators distribute over one another.

Law 5.6.7 (demonic-angelic-distributivity)

$$P \sqcap_{\mathcal{D}\mathbf{ac}} (Q \sqcup_{\mathcal{D}\mathbf{ac}} R) = (P \sqcap_{\mathcal{D}\mathbf{ac}} Q) \sqcup_{\mathcal{D}\mathbf{ac}} (P \sqcap_{\mathcal{D}\mathbf{ac}} R)$$

Proof. Follows from the distributive properties of conjunction and disjunction. Equivalently, this follows from the results established in the binary multirelational model of Chapter 4 and the respective isomorphism. □

This result has also been established in other models, such as the predicate transformer model [13]. Since the angelic choice operator is the least upper bound of the lattice, this result follows directly from the properties of the lattice.

5.7 Relationship of H3 designs with angelic nondeterminism

In this section we explore the relationship between the theory that we propose and that of [14]. An isomorphism is established for a subset of the theory of designs with angelic nondeterminism that are **A** and **H3**-healthy.

We begin Section 5.7.1 by characterising the correspondence between the alphabets of the two theories. In Section 5.7.2 and Section 5.7.3 the linking functions between the theories are defined: *d2pbmh* that maps from designs into predicates, and *pbmh2d* that maps in the inverse direction. We prove that both functions are closed within the respective theories. Finally in Section 5.7.4 the isomorphism is established.

5.7.1 Alphabets

As mentioned previously in Section 5.1, the alphabet of the theory we propose differs slightly from that of [14], in that ac' is a set of final states, but we consider undashed variables in the record components instead. In the following Law 5.7.1 we establish that the functions we presented earlier, $acdash2ac$ and $ac2acdash$, are actually the inverse of each other.

Law 5.7.1 ($acdash2ac \circ ac2acdash$)

$$acdash2ac \circ ac2acdash(ss) = ss$$

Proof.

$$
\begin{aligned}
& acdash2ac \circ ac2acdash(ss) && \{\text{Definition of } acdash2ac\}\\
&= \left\{\begin{array}{l} s_0 : S_{in\alpha}, s_1 : S_{out\alpha} \\ \mid s_1 \in ac2acdash(ss) \\ \mid \wedge (\bigwedge x : \alpha P \bullet s_0.x = s_1.(x')) \bullet s_0 \end{array}\right\} && \{\text{Definition of } ac2acdash\}\\
&= \left\{\begin{array}{l} s_0 : S_{in\alpha}, s_1 : S_{out\alpha} \\ \mid s_1 \in \left\{\begin{array}{l} z_0 : S_{in\alpha}, z_1 : S_{out\alpha} \\ \mid z_0 \in ss \wedge (\bigwedge x : \alpha \bullet z_0.x = z_1.(x')) \bullet z_1 \end{array}\right\} \\ \mid \wedge (\bigwedge x : \alpha \bullet s_0.x = s_1.(x')) \bullet s_0 \end{array}\right\} \\
& && \{\text{Property of sets}\}\\
&= \left\{\begin{array}{l} s_0 : S_{in\alpha}, s_1 : S_{out\alpha} \\ \mid \exists z_0 : S_{in\alpha} \bullet z_0 \in ss \wedge (\bigwedge x : \alpha \bullet z_0.x = s_1.(x')) \\ \mid \wedge (\bigwedge x : \alpha P \bullet s_0.x = s_1.(x')) \bullet s_0 \end{array}\right\} \\
& && \{\text{Equality of records}\}\\
&= \{s_0 : S_{in\alpha} \mid \exists z_0 : S_{in\alpha} \bullet z_0 \in ss \wedge z_0 = s_0 \bullet s_0\} && \{\text{One-point rule}\}\\
&= \{s_0 : S_{in\alpha} \mid s_0 \in ss \bullet s_0\} && \{\text{Property of sets}\}\\
&= ss
\end{aligned}
$$

□

This means that we can recover the ac' of either theory as needed. Some of the proofs in this section use auxiliary results about these functions that are established in Appendix G.

We observe that we also need to address the fact that we have a single initial state s that encapsulates the values of the initial program variables as record components. This notion is handled directly by the linking functions.

5.7.2 From designs to PBMH predicates ($d2pbmh$)

The first linking function of interest is $d2pbmh$ that maps from designs that are **A** and **H3**-healthy into the theory of [14]. Its definition is presented below.

Definition 65

$$
\begin{array}{l}
d2pbmh : \mathbf{A} \nrightarrow \mathbf{PBMH} \\
d2pbmh(P) \\
\mathrel{\widehat=} \\
\exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac'
\end{array}
$$

For a design P, via the substitution in P^f and P^t, we consider both its pre and postconditions directly. This is sufficient since we require ok to be $true$ and hide ok' (Law A.2.3). The substitution of $in\alpha$ for s corresponds to the substitution of every occurrence of a record component $s.x$ for x, where x is an input program variable. Finally, we substitute ac' in P with the temporary variable ac_0. This allows us to relate the set of final states ac_0 with ac' by applying $ac2acdash$ that replaces every undashed variable in all sets of states in ac_0 into dashed ones. Although the definition considers a superset of $ac2acdash(ac_0)$ rather than equality this is not an issue, since for every P that is **A**-healthy the sets of final states are always upward closed.

In the following Theorem 5.7.1 we prove that $d2pbmh$ yields predicates that are **PBMH**-healthy.

Theorem 5.7.1 *Provided P is* **A** *and* **H3***-healthy.*

$$\mathbf{PBMH}(d2pbmh(P)) = d2pbmh(P)$$

Proof.

$$
\begin{array}{l}
\mathbf{PBMH}(d2pbmh(P)) \hfill \{\text{Definition of } d2pbmh\} \\
= \mathbf{PBMH}(\exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac') \\
\hfill \{\text{Definition of } \mathbf{PBMH}\} \\
= (\exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac')\ ;\ ac \subseteq ac' \\
\hfill \{\text{Definition of sequential composition}\} \\
= \exists\, ac_1, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac_1 \land ac_1 \subseteq ac' \\
\hfill \{\text{Transitivity of subset inclusion}\}
\end{array}
$$

$= \exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac'$

{Definition of $d2pbmh$}

$= d2pbmh(P)$

□

The upward closure of $d2pbmh$ follows directly from the definition of $d2pbmh$. The proviso of Theorem 5.7.1 ensures that the function is only applied to designs that are **A** and **H3**-healthy.

5.7.3 From PBMH predicates to designs ($pbmh2d$)

In this section we define the second linking function $pbmh2d$ that maps from predicates in the theory of [14] into designs that are **A** and **H3**-healthy.

Definition 66

$$pbmh2d : \mathbf{PBMH} \nrightarrow \mathbf{A}$$

$$pbmh2d(P) \mathrel{\widehat=} \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \end{array}\right)$$

The definition yields a design whose precondition guarantees successful termination, the postcondition follows the same idea explored in the definition of $d2pbmh$. Every input program variable x in $in\alpha$ is substituted with $s.x$, where s is the initial state, and ac_0 is related to ac' in our theory by application of $acdash2ac$. In the model of [14], the possibility of non termination occurs when ac' is the empty set. Therefore the negation of this predicate can be taken as a precondition.

In the following Theorem 5.7.2 we prove that $pbmh2d$ yields designs that are **A** and **H3**-healthy.

Theorem 5.7.2 *Provided P is satisfies* **PBMH***.*

$$\mathbf{A} \circ \mathbf{H3}(pbmh2d(P)) = pbmhd2d(P)$$

Proof.

$\mathbf{A} \circ \mathbf{H3}(pbmh2d(P))$ {Definition of $pbmh2d$}

$$= \mathbf{A} \circ \mathbf{H3} \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \end{array} \right)$$

$$\{\text{Definition of } \mathbf{A} \circ \mathbf{H3}\}$$

$$= \left(\begin{array}{l} \exists\, ac' \bullet \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \mathbf{PBMH}(\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac') \\ \wedge\, ac' \neq \emptyset \end{array} \right)$$

$$\{\text{Predicate calculus: } ac' \text{ not free}\}$$

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \mathbf{PBMH}(\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac') \\ \wedge\, ac' \neq \emptyset \end{array} \right)$$

$$\{\text{Definition of } \mathbf{PBMH}\}$$

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ ((\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac') \,;\, ac \subseteq ac') \\ \wedge\, ac' \neq \emptyset \end{array} \right)$$

$$\{\text{Definition of sequential composition and substitution}\}$$

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ (\exists\, ac_1, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac_1 \wedge ac_1 \subseteq ac') \\ \wedge\, ac' \neq \emptyset \end{array} \right)$$

$$\{\text{Transitivity of subset inclusion}\}$$

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' \neq \emptyset \end{array} \right)$$

$$\{\text{Lemma G.3.1}\}$$

$$= pbmh2d(P)$$

$\square$

Similarly to the definition of $d2pbmh$, the proviso of Theorem 5.7.2 ensures that the function is only applied to predicates that are **PBMH**-healthy.

5.7.4 Isomorphism: $d2pbmh$ and $pbmh2d$

In this section we establish that the linking functions $d2pbmh$ and $pbmh2d$ are bijections. This result is established by Theorems 5.7.3 and 5.7.4.

Theorem 5.7.3 *Provided P is* $\mathbf{A} \circ \mathbf{H3}$*-healthy.*

$$pbmh2d \circ d2pbmh(P) = P$$

Proof.

$$\begin{aligned}
& pbmh2d \circ d2pbmh(P) && \{\text{Definition of } d2pbmh\} \\
&= pbmh2d(\exists ac_0 \bullet (\neg P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \land ac2acdash(ac_0) \subseteq ac') \\
& && \{\text{Definition of } pbmh2d\} \\
&= \left(\begin{array}{l}
\neg \left(\begin{array}{l} \exists ac_0 \bullet (\neg P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \\ \land ac2acdash(ac_0) \subseteq ac' \end{array}\right)[\emptyset/ac'][s/in\alpha] \\
\vdash \\
\exists ac_0 \bullet \left(\begin{array}{l} \exists ac_0 \bullet (\neg P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \\ \land ac2acdash(ac_0) \subseteq ac' \end{array}\right)[ac_0/ac'][s/in\alpha] \\
\land acdash2ac(ac_0) \subseteq ac'
\end{array}\right) \\
& && \{\text{Variable renaming}\} \\
&= \left(\begin{array}{l}
\neg \left(\begin{array}{l} \exists ac_0 \bullet (\neg P^f \Rightarrow P^t)[ac_0/ac'][in\alpha/s] \\ \land ac2acdash(ac_0) \subseteq ac' \end{array}\right)[\emptyset/ac'][s/in\alpha] \\
\vdash \\
\exists ac_0 \bullet \left(\begin{array}{l} \exists ac_1 \bullet (\neg P^f \Rightarrow P^t)[ac_1/ac'][in\alpha/s] \\ \land ac2acdash(ac_1) \subseteq ac' \end{array}\right)[ac_0/ac'][s/in\alpha] \\
\land acdash2ac(ac_0) \subseteq ac'
\end{array}\right) \\
& && \{\text{Substitution}\} \\
&= \left(\begin{array}{l}
\neg (\exists ac_0 \bullet (\neg P^f \Rightarrow P^t)[ac_0/ac'] \land ac2acdash(ac_0) \subseteq \emptyset) \\
\vdash \\
\left(\begin{array}{l} \exists ac_0, ac_1 \bullet (\neg P^f \Rightarrow P^t)[ac_1/ac'] \\ \land ac2acdash(ac_1) \subseteq ac_0 \land acdash2ac(ac_0) \subseteq ac' \end{array}\right)
\end{array}\right) \\
& && \{\text{Property of } ac2acdash \text{ and } acdash2ac\}
\end{aligned}$$

$$= \left(\begin{array}{l} \neg \left(\begin{array}{l} \exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'] \land \\ acdash2ac \circ ac2acdash(ac_0) \subseteq acdash2ac(\emptyset) \end{array}\right) \\ \vdash \\ \left(\begin{array}{l} \exists\, ac_0, ac_1 \bullet (\neg\, P^f \Rightarrow P^t)[ac_1/ac'] \\ \land\, acdash2ac \circ ac2acdash(ac_1) \subseteq acdash2ac(ac_0) \\ \land\, acdash2ac(ac_0) \subseteq ac' \end{array}\right) \end{array}\right)$$

{Transitivity of subset inclusion and Law G.1.3}

$$= \left(\begin{array}{l} \neg\, (\exists\, ac_0 \bullet (\neg\, P^f \Rightarrow P^t)[ac_0/ac'] \land ac_0 \subseteq \emptyset) \\ \vdash \\ \exists\, ac_1 \bullet (\neg\, P^f \Rightarrow P^t)[ac_1/ac'] \land ac_1 \subseteq ac' \end{array}\right)$$

{Case-analysis on ac_0 and definition of sequential composition}

$$= \left(\begin{array}{l} \neg\, (\neg\, P^f \Rightarrow P^t)[ac_0/ac'][\emptyset/ac_0] \\ \vdash \\ (\neg\, P^f \Rightarrow P^t) \,;\, ac \subseteq ac' \end{array}\right)$$

{Substitution and definition of **PBMH**}

$$= (\neg\, (\neg\, P^f \Rightarrow P^t)[\emptyset/ac'] \vdash \mathbf{PBMH}(\neg\, P^f \Rightarrow P^t))$$

{Assumption: P is $\mathbf{A} \circ \mathbf{H3}$-healthy}

$$= \left(\begin{array}{l} \neg\, (\neg\, P^f \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset))[\emptyset/ac'] \\ \vdash \\ \mathbf{PBMH}(\neg\, P^f \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset)) \end{array}\right)$$

{Substitution under assumption that ac' is not free in P^f}

$$= \left(\begin{array}{l} \neg\, (\neg\, P^f \Rightarrow (\mathbf{PBMH}(P^t) \land \emptyset \neq \emptyset)) \\ \vdash \\ \mathbf{PBMH}(\neg\, P^f \Rightarrow (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset)) \end{array}\right)$$

{Property of sets and predicate calculus}

$$= (\neg\, P^f \vdash \mathbf{PBMH}(P^f \lor (\mathbf{PBMH}(P^t) \land ac' \neq \emptyset))) \qquad \text{\{Law D.2.1\}}$$

$$= (\neg\, P^f \vdash \mathbf{PBMH}(P^f) \lor \mathbf{PBMH}(\mathbf{PBMH}(P^t) \land ac' \neq \emptyset))$$

{Lemma D.4.5 and Law D.2.2}

$$= (\neg\, P^f \vdash \mathbf{PBMH}(P^f) \lor \mathbf{PBMH} \circ \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$$

{Law D.1.1 and Lemma D.4.6}

$$= (\neg\, P^f \vdash P^f \lor \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$$

{Property of sets and predicate calculus}

$$= (\neg\, P^f \land (\neg\, P^f \lor \neg\, ok') \vdash \mathbf{PBMH}(P^t) \land ac' \neq \emptyset)$$

{Predicate calculus: absorption law}

$$= (\neg P^f \vdash \mathbf{PBMH}(P^t) \wedge ac' \neq \emptyset) \qquad \{\text{Definition of } \mathbf{A} \circ \mathbf{H3}\}$$
$$= \mathbf{A} \circ \mathbf{H3}(P) \qquad \{\text{Assumption: } P \text{ is } \mathbf{A} \circ \mathbf{H3}\text{-healthy}\}$$
$$= P$$

$\square$

Theorem 5.7.4 *Provided P is* **PBMH**-*healthy.*

$$d2pbmh \circ pbmh2d(P) = P$$

Proof.

$$d2pbmh \circ pbmh2d(P) \qquad \{\text{Definition of } pbmh2d\}$$
$$= d2pbmh \left(\begin{array}{l} \neg P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists ac_0 \bullet P[ac_0/ac'][s/in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \end{array} \right)$$
$$\{\text{Definition of } d2pbmh\}$$
$$= \left(\begin{array}{l} \exists ac_0 \bullet \left(\begin{array}{l} \neg P[\emptyset/ac'][s/in\alpha] \\ \Rightarrow \\ \left(\begin{array}{l} \exists ac_0 \bullet P[ac_0/ac'][s/in\alpha] \\ \wedge acdash2ac(ac_0) \subseteq ac' \end{array} \right) \end{array} \right) [ac_0/ac'][in\alpha/s] \\ \wedge ac2acdash(ac_0) \subseteq ac' \end{array} \right)$$
$$\{\text{Variable renaming}\}$$
$$= \left(\begin{array}{l} \exists ac_0 \bullet \left(\begin{array}{l} \neg P[\emptyset/ac'][s/in\alpha] \\ \Rightarrow \\ \left(\begin{array}{l} \exists ac_1 \bullet P[ac_1/ac'][s/in\alpha] \\ \wedge acdash2ac(ac_1) \subseteq ac' \end{array} \right) \end{array} \right) [ac_0/ac'][in\alpha/s] \\ \wedge ac2acdash(ac_0) \subseteq ac' \end{array} \right)$$
$$\{\text{Predicate calculus and substitution}\}$$
$$= \left(\begin{array}{l} \exists ac_0 \bullet P[\emptyset/ac'][s/in\alpha][ac_1/ac'][in\alpha/s] \wedge ac2acdash(ac_0) \subseteq ac' \\ \vee \\ \left(\begin{array}{l} \exists ac_0, ac_1 \bullet P[ac_1/ac'][s/in\alpha][ac_0/ac'][in\alpha/s] \\ \wedge acdash2ac(ac_1) \subseteq ac_0 \\ \wedge ac2acdash(ac_0) \subseteq ac' \end{array} \right) \end{array} \right)$$
$$\{\text{Substitution}\}$$

$$= \left(\begin{array}{l} P[\emptyset/ac'] \land \exists\, ac_0 \bullet ac2acdash(ac_0) \subseteq ac' \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0, ac_1 \bullet P[ac_1/ac'] \land acdash2ac(ac_1) \subseteq ac_0 \\ \land\ ac2acdash(ac_0) \subseteq ac' \end{array} \right) \end{array} \right)$$

$$\{\text{Property of } ac2acdash \text{ and } acdash2ac\}$$

$$= \left(\begin{array}{l} P[\emptyset/ac'] \land \exists\, ac_0 \bullet ac2acdash(ac_0) \subseteq ac' \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0, ac_1 \bullet P[ac_1/ac'] \land ac2acdash \circ acdash2ac(ac_1) \subseteq ac2acdash(ac_0) \\ \land\ ac2acdash(ac_0) \subseteq ac' \end{array} \right) \end{array} \right)$$

$$\{\text{Property of } ac2acdash \text{ and transitivity of subset inclusion}\}$$

$$= \left(\begin{array}{l} P[\emptyset/ac'] \land \exists\, ac_0 \bullet ac2acdash(ac_0) \subseteq ac' \\ \lor \\ \exists\, ac_1 \bullet P[ac_1/ac'] \land ac_1 \subseteq ac' \end{array} \right) \qquad \{\text{Case-analysis on } ac_0\}$$

$$= P[\emptyset/ac'] \lor (\exists\, ac_1 \bullet P[ac_1/ac'] \land ac_1 \subseteq ac')$$

$$\{\text{Instantiation of } ac_1 \text{ for } ac_1 = \emptyset\}$$

$$= \exists\, ac_1 \bullet P[ac_1/ac'] \land ac_1 \subseteq ac')$$

$$\{\text{Definition of } \mathbf{PBMH} \text{ and assumption that } P \text{ satisfies } \mathbf{PBMH}\}$$

$$= P$$

□

While this is an expected result, it is reassuring that the subset of our theory that is **H3**-healthy is in exact correspondence with the UTP theory of [14].

We observe that the subset of the binary multirelational model of Chapter 4 that is **BMH3**-healthy is isomorphic to the original theory of binary multirelations. Since binary multirelations are also isomorphic to the UTP theory of [14], the result presented in this section is also in agreement. This result completes the relationships depicted in Figure 1.1.

5.8 Final considerations

In this chapter we have presented a new UTP theory of designs that is capable of modelling angelic and demonic nondeterminism. The novel contribution lies in the use of the variables *ok* and *ok'*, as in every theory of designs, and the capability to express non-**H3**-designs as well as both demonic and angelic choice. While all known existing models for program correctness restrict their attention to necessarily terminating programs, we relax this constraint in order to pave the way for the development of a theory of reactive designs with angelic nondeterminism.

The healthiness conditions of the theory have been presented and their properties proved, including idempotency and monotonicity. Through the co-development of the binary multirelational model in Chapter 4, and its subsequent isomorphism, we have been able to justify and explore the definition of the operators and the refinement ordering. It is reassuring to know that the refinement order as given by universal reverse implication corresponds to subset inclusion in the binary multirelational model.

Perhaps the most challenging aspect of the theory is that it is non-homogeneous. As a consequence sequential composition cannot be defined as relational composition. While the definition for sequential composition is not immediately obvious, it is more intuitive when considered in the equivalent binary multirelational model.

Finally, we have also linked a subset of this model with the UTP theory of [14]. This is a complementary result to the link between the binary multirelational model of $BM_\perp$ relations and that of the original theory of binary multirelations. This gives us further assurance as to the capability to express the existing theories as a subset of our own correctly.

Chapter 6

Conclusions

In this chapter we present a summary of our findings in Section 6.1. This is followed by the discussion of future work in Section 6.2.

6.1 Summary

The concept of angelic nondeterminism is useful in the context of formal specifications. It has traditionally been studied in the context of the refinement calculus [11–13]. However, as far as we know, it has not been characterised in a relational setting capable of modelling reactive programs.

In this work we have presented a new UTP theory of designs with angelic nondeterminism that can cope with non-**H3** designs, a first step in the definition of a theory of reactive designs with angelic nondeterminism. The healthiness conditions and the main operators have been defined and their properties proved.

In order to motivate our predicative model, we developed an equivalent extended binary multirelational model. This provides an insight into the definition of some of the operators, such as sequential composition. Its definition in the binary multirelational model is based on our understanding of the original theory of designs [1] and the theory of binary multirelations [15].

Unfortunately, in the model we propose sequential composition cannot be defined as relational composition as we use non-homogenenous relations. Instead, we provide an alternative definition that is partially based on substitution as proposed by Cavalanti et al. [14]. We extend that notion for non-**H3** designs and justify its definition with the isomorphism between the models. It is pleasing that our definition resembles that of the theory of designs.

For both of the models that we have developed, we have studied the relationship between their subsets of interest and the existing theories. The fact that we have been able to prove that they are equivalent is reassuring. These results consolidate our understanding of the models.

6.2 Future work

As already mentioned, the theory proposed in this work is the first step towards the definition of a theory of reactive designs with angelic nondeterminism. Although the results we have obtained are consistent with a theory of designs, it remains to be seen what are the implications with respect to a theory of reactive programs.

In addition, since there is a collection of different binary multirelational models as pointed out by Rewitzky [15], it would be interesting to explore whether other isomorphisms can be established. In fact, exploring the relationship between our model and any other existing theories would further help validate the model and consolidate our understanding of it.

Since it is our goal to provide a mathematically rigorous theory for software engineering, it is only recommended that, in the future, further validation of all applicable theorems and lemmas is carried out by mechanising the theory with the help of a theorem prover.

Finally, due to the foundational importance of our contribution, it would be desirable if this model could be exploited in practice, perhaps even in the context of unforeseen domains.

Acronyms

CSP Communicating Sequential Processes

ZRC Z Refinement Calculus

VDM Vienna Development Method

ASM Abstract State Machine

FSM Finite State Machines

CCS Calculus of Concurrent Systems

JCSP Java Communicating Sequential Processes

FDR Failures-Divergence Refinement

UTP Unifying Theories of Programming

BNF Backus-Naur Normal Form

Bibliography

[1] C. A. R. Hoare and H. Jifeng, *Unifying Theories of Programming*. Prentice Hall International Series in Computer Science, 1998.

[2] T. Santos, A. Cavalcanti, and A. Sampaio, "Object-Orientation in the UTP," in *Unifying Theories of Programming*, ser. Lecture Notes in Computer Science, S. Dunne and B. Stoddart, Eds. Springer Berlin / Heidelberg, 2006, vol. 4010, pp. 18–37. [Online]. Available: http://dx.doi.org/10.1007/11768173_2

[3] W. Harwood, A. Cavalcanti, and J. Woodcock, "A Theory of Pointers for the UTP," in *Theoretical Aspects of Computing - ICTAC 2008*, ser. Lecture Notes in Computer Science, J. Fitzgerald, A. Haxthausen, and H. Yenigun, Eds. Springer Berlin / Heidelberg, 2008, vol. 5160, pp. 141–155. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85762-4_10

[4] A. Sherif and J. He, "Towards a Time Model for *circus*," in *Proceedings of the 4th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering*, ser. ICFEM '02. London, UK, UK: Springer-Verlag, 2002, pp. 613–624. [Online]. Available: http://portal.acm.org/citation.cfm?id=646272.685816

[5] A. Sherif, "A Framework for Specification and Validation of Real-Time Systems using *Circus Actions*," Ph.D. dissertation, Center of Informatics - Federal University of Pernambuco, Brazil, 2006. [Online]. Available: http://www.cs.york.ac.uk/circus/publications/papers/06-sherif.pdf

[6] K. Wei, J. Woodcock, and A. Cavalcanti, "New *Circus Time*," University of York, Tech. Rep., February 2012. [Online].

Available: http://www.cs.york.ac.uk/circus/publications/techreports/reports/Circus%20Time.pdf

[7] A. Cavalcanti and J. Woodcock, "A Tutorial Introduction to CSP in *Unifying Theories of Programming*," in *Refinement Techniques in Software Engineering*, ser. Lecture Notes in Computer Science, A. Cavalcanti, A. Sampaio, and J. Woodcock, Eds. Springer Berlin / Heidelberg, 2006, vol. 3167, pp. 220–268. [Online]. Available: http://dx.doi.org/10.1007/11889229_6

[8] A. W. Roscoe, *The Theory and Practice of Concurrency*. Prentice Hall, 1998.

[9] A. Cavalcanti, A. Sampaio, and J. Woodcock, "A Refinement Strategy for *Circus*," *Formal Aspects of Computing*, vol. 15, pp. 146–181, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00165-003-0006-5

[10] M. Oliveira, "Formal Derivation of State-Rich Reactive Programs using *Circus*," Ph.D. dissertation, University of York, 2005.

[11] J. M. Morris, "A theoretical basis for stepwise refinement and the programming calculus," *Sci. Comput. Program.*, vol. 9, pp. 287–306, December 1987. [Online]. Available: http://dl.acm.org/citation.cfm?id=34898.34903

[12] C. Morgan, *Programming from specifications*. Prentice Hall, 1994.

[13] R. Back and J. Wright, *Refinement calculus: a systematic introduction*, ser. Graduate texts in computer science. Springer, 1998. [Online]. Available: http://books.google.co.uk/books?id=fRWQe23oB7kC

[14] A. Cavalcanti, J. Woodcock, and S. Dunne, "Angelic nondeterminism in the unifying theories of programming," *Formal Aspects of Computing*, vol. 18, pp. 288–307, 2006. [Online]. Available: http://dx.doi.org/10.1007/s00165-006-0001-8

[15] I. Rewitzky, "Binary Multirelations," in *Theory and Applications of Relational Structures as Knowledge Instruments*, ser. Lecture Notes in Computer Science, H. de Swart, E. Orlowska, G. Schmidt, and M. Roubens, Eds. Springer Berlin / Heidelberg, 2003, vol.

2929, pp. 1964–1964. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24615-2_12

[16] J. M. Morris and M. Tyrrell, "Terms with unbounded demonic and angelic nondeterminacy," *Science of Computer Programming*, vol. 65, no. 2, pp. 159 – 172, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167642306002127

[17] J. Morris and M. Tyrrell, "Dual unbounded nondeterminacy, recursion, and fixpoints," *Acta Informatica*, vol. 44, pp. 323–344, 2007. [Online]. Available: http://dx.doi.org/10.1007/s00236-007-0049-9

[18] J. M. Morris and M. Tyrrell, "Dually nondeterministic functions," *ACM Trans. Program. Lang. Syst.*, vol. 30, pp. 34:1–34:34, October 2008. [Online]. Available: http://doi.acm.org/10.1145/1391956.1391961

[19] J. Morris and M. Tyrrell, "Modelling higher-order dual nondeterminacy," *Acta Informatica*, vol. 45, pp. 441–465, 2008. [Online]. Available: http://dx.doi.org/10.1007/s00236-008-0076-1

[20] W. H. Hesselink, "Alternating states for dual nondeterminism in imperative programming," *Theoretical Computer Science*, vol. 411, no. 22-24, pp. 2317 – 2330, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030439751000143X

[21] M. Tyrrell, J. Morris, A. Butterfield, and A. Hughes, "A Lattice-Theoretic Model for an Algebra of Communicating Sequential Processes," in *Theoretical Aspects of Computing - ICTAC 2006*, ser. Lecture Notes in Computer Science, K. Barkaoui, A. Cavalcanti, and A. Cerone, Eds. Springer Berlin / Heidelberg, 2006, vol. 4281, pp. 123–137. [Online]. Available: http://dx.doi.org/10.1007/11921240_9

[22] J. Woodcock and A. Cavalcanti, "A Tutorial Introduction to Designs in Unifying Theories of Programming," in *Integrated Formal Methods*, ser. Lecture Notes in Computer Science, E. Boiten, J. Derrick, and G. Smith, Eds. Springer Berlin / Heidelberg, 2004, vol. 2999, pp. 40–66. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24756-2_4

[23] A. L. C. Cavalcanti, "A Refinement Calculus for Z," Ph.D. dissertation, Oxford University Computing Laboratory, Oxford - UK, 1997, technical Monograph TM-PRG-123, ISBN 00902928-97-X.

[24] B. Davey and H. Priestley, *Introduction to Lattices and Order*, ser. Cambridge mathematical textbooks. Cambridge University Press, 2002. [Online]. Available: http://books.google.co.uk/books?id=vVVTxeuiyvQC

[25] A. Cavalcanti and J. Woodcock, "Angelic Nondeterminism and Unifying Theories of Programming," University of Kent, Tech. Rep., 2004. [Online]. Available: http://kar.kent.ac.uk/14151/

[26] C. E. Martin, S. A. Curtis, and I. Rewitzky, "Modelling Nondeterminism," in *MPC, volume 3125 of LNCS*. Springer, 2004, pp. 228–251.

[27] J. Morris, "Augmenting Types with Unbounded Demonic and Angelic Nondeterminacy," in *Mathematics of Program Construction*, ser. Lecture Notes in Computer Science, D. Kozen, Ed. Springer Berlin / Heidelberg, 2004, vol. 3125, pp. 274–288. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-27764-4_15

[28] J. M. Spivey, *The Z notation: A Reference Manual*. Prentice Hall, 1989. [Online]. Available: http://spivey.oriel.ox.ac.uk/~mike/zrm/zrm.pdf

[29] J. Woodcock and J. Davies, *Using Z: Specification, Refinement, and Proof*. Prentice Hall, 1996.

[30] A. Cavalcanti, A. Wellings, and J. Woodcock, "The Safety-Critical Java Memory Model: A Formal Account," in *FM 2011: Formal Methods*, ser. Lecture Notes in Computer Science, M. Butler and W. Schulte, Eds. Springer Berlin / Heidelberg, 2011, vol. 6664, pp. 246–261. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21437-0_20

Appendix A

Theory of designs

A.1 Healthiness conditions

H2A

Definition 67

$$\mathbf{H2A}(P) \mathrel{\widehat=} \neg\, P^f \Rightarrow (P^t \wedge ok')$$

Law A.1.1 (H2A $\Leftrightarrow$ H2) *The definition of* **H2A** *implies that the fixpoints are the same as those of* **H2**.

Proof for implication. The following proof is based on [29].

$$\begin{array}{ll}
P & \{\text{Introduce fresh variable and substitution}\}\\
= \exists\, ok_0 \bullet P \wedge ok' = ok_0 & \{\text{Case-split on } ok_0\}\\
= (\neg\, ok' \wedge P^f) \vee (ok' \wedge P^t) & \{\text{Assumption: P is } \mathbf{H2}\text{-healthy}\}\\
= (\neg\, ok' \wedge P^f \wedge P^t) \vee (ok' \wedge P^t) & \{\text{Propositional calculus}\}\\
= (((\neg\, ok' \wedge P^f) \vee ok') \wedge P^t) & \{\text{Propositional calculus}\}\\
= ((P^f \vee ok') \wedge P^t) & \{\text{Propositional calculus}\}\\
= (P^f \wedge P^t) \vee (ok' \wedge P^t) & \{\text{Assumption: P is } \mathbf{H2}\text{-healthy}\}\\
= P^f \vee (ok' \wedge P^t) & \{\text{Propositional calculus}\}\\
= \neg\, P^f \Rightarrow (P^t \wedge ok') &
\end{array}$$

□

Proof for reverse implication.

$$
\begin{aligned}
&[(\mathbf{H2A}(P))^f \Rightarrow (\mathbf{H2A}(P))^t] && \{\text{Definition of } \mathbf{H2A}\}\\
&= [(\neg\, P^f \Rightarrow (P^t \wedge ok'))^f \Rightarrow (\neg\, P^f \Rightarrow (P^t \wedge ok'))^t] && \{\text{Substitution}\}\\
&= [(P^f \Rightarrow (\neg\, P^f \Rightarrow P^t)] && \{\text{Propositional calculus}\}\\
&= [\neg\, P^f \vee P^f \vee P^t] && \{\text{Propositional calculus}\}\\
&= \mathit{true}
\end{aligned}
$$

□

A.2 Lemmas

Law A.2.1 (design-true-ok') *Provided $ok \wedge P$ and ok' is not free in P.*

$$(P \vdash Q)^t = Q$$

Proof. As stated and proved in [30] (Lemma 4.2). □

Law A.2.2 (design-false-ok') *Provided ok' is not free in P.*

$$ok \wedge \neg\, (P \vdash Q)^f = ok \wedge P$$

Proof. As stated and proved in [30] (Lemma 4.3). □

Law A.2.3 (design-exists-ok')

$$\exists\, ok' \bullet (P \vdash Q) = (ok \wedge P) \Rightarrow Q$$

Proof.

$$
\begin{aligned}
&\exists\, ok' \bullet (P \vdash Q) && \{\text{Definition of design}\}\\
&= \exists\, ok' \bullet (ok \wedge P) \Rightarrow (Q \wedge ok') && \{\text{Case-split on } ok'\}\\
&= ((ok \wedge P) \Rightarrow Q) \vee \neg\, (ok \wedge P) && \{\text{Propositional calculus}\}\\
&= (ok \wedge P) \Rightarrow Q
\end{aligned}
$$

□

Law A.2.4 (design-$\sqcup$)

$$(\neg\, P^f \vdash P^t) \sqcup (\neg\, Q^f \vdash Q^t)$$
$$=$$
$$(\neg\, P^f \lor \neg\, Q^f \vdash (\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t))$$

Proof.

$$
\begin{aligned}
& (\neg\, P^f \vdash P^t) \sqcup (\neg\, Q^f \vdash Q^t) && \{\text{Definition of design}\}\\
&= ((ok \land \neg\, P^f) \Rightarrow (P^t \land ok')) \sqcup ((ok \land \neg\, Q^f) \Rightarrow (Q^t \land ok'))\\
& && \{\text{Definition of } \sqcup\}\\
&= ((ok \land \neg\, P^f) \Rightarrow (P^t \land ok')) \land ((ok \land \neg\, Q^f) \Rightarrow (Q^t \land ok'))\\
& && \{\text{Propositional calculus}\}\\
&= ok \Rightarrow ((P^t \land ok') \lor P^f) \land ((Q^t \land ok') \lor Q^f) && \{\text{Propositional calculus}\}\\
&= ok \Rightarrow (P^t \lor P^f) \land (ok' \lor P^f) \land (Q^t \lor Q^f) \land (ok' \lor Q^f)\\
& && \{\text{Propositional calculus}\}\\
&= ok \Rightarrow (P^t \lor P^f) \land (Q^t \lor Q^f) \land (ok' \lor (P^f \land Q^f))\\
& && \{\text{Propositional calculus: absorption law}\}\\
&= ok \Rightarrow ((P^f \land Q^f) \lor P^t \lor P^f) \land ((P^f \land Q^f) \lor Q^t \lor Q^f) \land (ok' \lor (P^f \land Q^f))\\
& && \{\text{Propositional calculus}\}\\
&= ok \Rightarrow (P^f \land Q^f) \lor ((P^t \lor P^f) \land (Q^t \lor Q^f) \land ok')\\
& && \{\text{Propositional calculus}\}\\
&= (ok \land \neg\, (P^f \land Q^f)) \Rightarrow ((\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t) \land ok')\\
& && \{\text{Definition of design}\}\\
&= (\neg\, P^f \lor \neg\, Q^f \vdash (\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t))
\end{aligned}
$$

□

Law A.2.5 (design-exists-ok'-$\sqcup$) *Provided P and Q are designs.*

$$\exists\, ok' \bullet (P \land Q) = (\exists\, ok' \bullet P) \land (\exists\, ok' \bullet Q)$$

Proof.

$$
\begin{aligned}
& (\exists\, ok' \bullet P) \land (\exists\, ok' \bullet Q) && \{\text{Assumption: } P \text{ and } Q \text{ are designs}\}\\
&= (\exists\, ok' \bullet (\neg\, P^f \vdash P^t)) \land (\exists\, ok' \bullet (\neg\, Q^f \vdash Q^t)) && \{\text{Law A.2.3}\}
\end{aligned}
$$

$= ((ok \land \neg\, P^f) \Rightarrow P^t) \land ((ok \land \neg\, Q^f) \Rightarrow Q^t)$ {Propositional calculus}
$= (ok \Rightarrow (P^t \lor P^f)) \land (ok \Rightarrow (Q^t \lor Q^f))$ {Propositional calculus}
$= ok \Rightarrow ((P^t \lor P^f) \land (Q^t \lor Q^f))$
{Propositional calculus: absorption law}
$= ok \Rightarrow (((P^f \land Q^f) \lor P^t \lor P^f) \land ((P^f \land Q^f) \lor Q^t \lor Q^f))$
{Propositional calculus}
$= ok \Rightarrow ((P^f \land Q^f) \lor ((P^t \lor P^f) \land (Q^t \lor Q^f)))$ {Propositional calculus}
$= (ok \land \neg\,(P^f \land Q^f)) \Rightarrow ((\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t))$ {Law A.2.3}
$= \exists\, ok' \bullet (\neg\,(P^f \land Q^f) \vdash (\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t))$
{Conjunction of designs}
$= \exists\, ok' \bullet (\neg\, P^f \vdash P^t) \land (\neg\, Q^f \vdash Q^t)$
{Assumption: P and Q are designs}
$= \exists\, ok' \bullet (P \land Q)$

□

Law A.2.6

$$
\begin{array}{l}
(\neg\, P^f \vdash P^t) \sqcup (\neg\, Q^f \vdash Q^t) \\
= \\
(\neg\, P^f \lor \neg\, Q^f \vdash (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))
\end{array}
$$

Proof.

$(\neg\, P^f \vdash P^t) \sqcup (\neg\, Q^f \vdash Q^t)$ {Conjunction of designs}
$= (\neg\, P^f \lor \neg\, Q^f \vdash (\neg\, P^f \Rightarrow P^t) \land (\neg\, Q^f \Rightarrow Q^t))$
{Propositional calculus}
$= (\neg\, P^f \lor \neg\, Q^f \vdash (P^f \lor P^t) \land (Q^f \lor Q^t))$ {Predicate calculus}
$= (\neg\,(P^f \land Q^f) \vdash (P^f \land Q^f) \lor (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))$
{Definition of design}

$$
= \left(\begin{array}{l}
(ok \land \neg\,(P^f \land Q^f)) \\
\Rightarrow \\
(((P^f \land Q^f) \lor (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t)) \land ok')
\end{array}\right)
$$

{Predicate calculus}

$$= \left(\begin{array}{l} (ok \land \neg\,(P^f \land Q^f) \land (\neg\,(P^f \land Q^f) \lor \neg\, ok')) \\ \Rightarrow \\ (((P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t)) \land ok') \end{array} \right)$$

{Predicate calculus: absorption law}

$$= (ok \land \neg\,(P^f \land Q^f)) \Rightarrow (((P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t)) \land ok')$$

{Definition of design}

$$= (\neg\,(P^f \land Q^f) \vdash (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))$$

{Predicate calculus}

$$= (\neg\, P^f \lor \neg\, Q^f \vdash (P^f \land Q^t) \lor (P^t \land Q^f) \lor (P^t \land Q^t))$$

$\square$

Appendix B

Binary multirelational model

B.1 Healthiness conditions

$\mathbf{bmh_0}$

Lemma B.1.1 ($\mathbf{bmh_0}$-idempotent)

$$\mathbf{bmh_0} \circ \mathbf{bmh_0}(B) = \mathbf{bmh_0}(B)$$

Proof.

$$\mathbf{bmh_0} \circ \mathbf{bmh_0}(B) \qquad \{\text{Definition of } \mathbf{bmh_0}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in \mathbf{bmh_0}(B) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

$$\{\text{Definition of } \mathbf{bmh_0}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

$$\{\text{Variable renaming}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_1 \bullet (s, ss_1) \in B \\ \land\, ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0, ss_1 \bullet (s, ss_1) \in B \\ \mid \wedge\, ss_1 \subseteq ss_0 \wedge (\perp \in ss_1 \Leftrightarrow \perp \in ss_0) \\ \mid \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

{Predicate calculus and transitivity of subset inclusion}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_1 \bullet (s, ss_1) \in B \wedge ss_1 \subseteq ss \wedge (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

{Definition of $\mathbf{bmh_0}$}

$$= \mathbf{bmh_0}(B)$$

□

$\mathbf{bmh_1}$

Lemma B.1.2 ($\mathbf{bmh_1}$-idempotent)

$$\mathbf{bmh_1} \circ \mathbf{bmh_1}(B) = \mathbf{bmh_1}(B)$$

Proof.

$\mathbf{bmh_1} \circ \mathbf{bmh_1}(B)$ {Definition of $\mathbf{bmh_1}$}

$$= \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in \mathbf{bmh_1}(B) \vee (s, ss) \in \mathbf{bmh_1}(B)\}$$

{Definition of $\mathbf{bmh_1}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss \cup \{\perp\}) \in B \vee (s, ss) \in B \end{array} \right\} \\ \mid \vee \\ \mid (s, ss) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \vee (s, ss) \in B\} \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss \cup \{\perp\} \cup \{\perp\}) \in B \vee (s, ss \cup \{\perp\}) \in B \\ \mid \vee \\ \mid (s, ss \cup \{\perp\}) \in B \vee (s, ss) \in B \end{array} \right\}$$

{Property of sets and predicate calculus}

$$= \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \vee (s, ss) \in B\}$$

{Definition of $\mathbf{bmh_1}$}

$$= \mathbf{bmh_1}(B)$$

$\square$

$\mathbf{bmh_2}$

Lemma B.1.3 ($\mathbf{bmh_2}$-idempotent)

$$\mathbf{bmh_2} \circ \mathbf{bmh_2}(B) = \mathbf{bmh_2}(B)$$

Proof.

$$
\begin{array}{ll}
\mathbf{bmh_2} \circ \mathbf{bmh_2}(B) & \{\text{Definition of } \mathbf{bmh_2}\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in \mathbf{bmh_2}(B) \\ \land \\ ((s, \{\perp\}) \in \mathbf{bmh_2}(B) \Leftrightarrow (s, \emptyset) \in \mathbf{bmh_2}(B)) \end{array} \right. \end{array} \right\} & \\
& \{\text{Definition of } \mathbf{bmh_2}\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \land \left(\begin{array}{l} (s, \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \Leftrightarrow \\ (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \end{array} \right) \end{array} \right. \end{array} \right\} & \\
& \{\text{Property of sets}\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \left(\begin{array}{l} ((s, \{\perp\}) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \\ \Leftrightarrow \\ ((s, \emptyset) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \end{array} \right) \end{array} \right. \end{array} \right\} & \\
& \{\text{Predicate calculus}\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land (((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \Leftrightarrow ((s, \emptyset) \in B \land (s, \{\perp\}) \in B)) \end{array} \right. \end{array} \right\} & \\
& \{\text{Predicate calculus}\} \\
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} & \{\text{Definition of } \mathbf{bmh_2}\}
\end{array}
$$

$= \mathbf{bmh_2}(B)$

$\square$

$\mathbf{bmh_3}$

Lemma B.1.4 ($\mathbf{bmh_3}$-idempotent)

$$\mathbf{bmh_3} \circ \mathbf{bmh_3}(B) = B$$

Proof.

$\mathbf{bmh_3} \circ \mathbf{bmh_3}(B)$ $\{\text{Definition of } \mathbf{bmh_3}\}$

$$= \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \mid ((s, \emptyset) \in \mathbf{bmh_3}(B) \lor \perp \notin ss) \land (s, ss) \in \mathbf{bmh_3}(B) \end{array} \right\}$$

$\{\text{Definition of } \mathbf{bmh_3}\}$

$$= \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \left| \begin{array}{l} \left((s, \emptyset) \in \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \end{array} \right\} \lor \perp \notin ss \right) \\ \land \\ (s, ss) \in \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \end{array} \right\} \end{array} \right. \end{array} \right\}$$

$\{\text{Property of sets}\}$

$$= \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \left| \begin{array}{l} ((((s, \emptyset) \in B \lor \perp \notin \emptyset) \land (s, \emptyset) \in B) \lor \perp \notin ss) \\ \land \\ (((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B) \end{array} \right. \end{array} \right\}$$

$\{\text{Predicate calculus: absorption law}\}$

$$= \left\{ \begin{array}{l} s : \mathit{State}, ss : \mathbb{P}\,\mathit{State}_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \end{array} \right\}$$

$\{\text{Predicate calculus and definition of } \mathbf{bmh_3}\}$

$= \mathbf{bmh_3}(B)$

$\square$

$\mathbf{bmh_0}$ and $\mathbf{bmh_1}$

Lemma B.1.5

$$\mathbf{bmh_0} \circ \mathbf{bmh_1}(B)$$
$$=$$
$$\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

Proof.

$\mathbf{bmh_0} \circ \mathbf{bmh_1}(B)$ {Definition of $\mathbf{bmh_0}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in \mathbf{bmh_1}(B) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

{Definition of $\mathbf{bmh_1}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0 \cup \{\perp\}) \in B \lor (s, ss_0) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0 \cup \{\perp\}) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

□

Properties

Lemma B.1.6 ($\mathbf{bmh_0} \circ \mathbf{bmh_1}$-commutative)

$$\mathbf{bmh_0} \circ \mathbf{bmh_1}(B) = \mathbf{bmh_1} \circ \mathbf{bmh_0}(B)$$

Proof.

$$\mathbf{bmh_1} \circ \mathbf{bmh_0}(B) \qquad \{\text{Definition of } \mathbf{bmh_1}\}$$

$$= \left\{ s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in \mathbf{bmh_0}(B) \lor (s, ss) \in \mathbf{bmh}_0(B) \right\}$$

$$\{\text{Definition of } \mathbf{bmh_0}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} \\ \lor \\ (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\perp\}) \land (\perp \in ss_0 \Leftrightarrow \perp \in (ss \cup \{\perp\}))) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets and predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\perp\}) \land \perp \in ss_0) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

$$\{\text{Lemma B.3.1}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0 \cup \{\perp\}) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

$$\{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0 \cup \{\perp\}) \in B \lor (s, ss_0) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\} \qquad \{\text{Lemma B.1.5}\}$$

$$= \mathbf{bmh_0} \circ \mathbf{bmh_1}(B)$$

□

$\mathbf{bmh_1}$ and $\mathbf{bmh_2}$

Lemma B.1.7

$$\mathbf{bmh_1} \circ \mathbf{bmh_2}(B) = \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \end{array} \right\}$$

Proof.

$$\begin{array}{lr} \mathbf{bmh_1} \circ \mathbf{bmh_2}(B) & \{\text{Definition of } \mathbf{bmh_1}\} \\ = \{ s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in \mathbf{bmh_2}(B) \lor (s, ss) \in \mathbf{bmh_2}(B) \} & \\ & \{\text{Definition of } \mathbf{bmh_2}\} \end{array}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \lor \\ (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, ss \cup \{\perp\}) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \\ \lor \\ ((s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \end{array} \right. \end{array} \right\}$$

$$\{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \land ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \end{array} \right\}$$

□

Lemma B.1.8

$$\mathbf{bmh_2} \circ \mathbf{bmh_1}(B) = \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \land ((s, \emptyset) \in B) \Rightarrow (s, \{\perp\}) \in B) \end{array} \right\}$$

Proof.

$$\mathbf{bmh_2} \circ \mathbf{bmh_1}(B) \qquad \{\text{Definition of } \mathbf{bmh_2}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in \mathbf{bmh_1}(B) \land ((s, \{\perp\}) \in \mathbf{bmh_1}(B) \Leftrightarrow (s, \emptyset) \in \mathbf{bmh_1}(B)) \end{array} \right\}$$

{Definition of $\mathbf{bmh_1}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \\ \land \\ \left(\begin{array}{l} (s, \{\perp\}) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \\ \Leftrightarrow \\ (s, \emptyset) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \\ \land \\ \left(\begin{array}{l} ((s, \{\perp\} \cup \{\perp\}) \in B \lor (s, \{\perp\}) \in B) \\ \Leftrightarrow \\ ((s, \emptyset \cup \{\perp\}) \in B \lor (s, \emptyset) \in B) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Property of sets and predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \\ \land \\ \left(\begin{array}{l} (s, \{\perp\}) \in B \\ \Leftrightarrow \\ ((s, \{\perp\}) \in B \lor (s, \emptyset) \in B) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \land ((s, \emptyset) \in B) \Rightarrow (s, \{\perp\}) \in B) \end{array} \right\}$$

□

It can be conclued from Lemma B.1.8 and Lemma B.1.7 that the functional application of $\mathbf{bmh_1} \circ \mathbf{bmh_2}$ is stronger than that of $\mathbf{bmh_2} \circ \mathbf{bmh_1}$. The order in which these two healthiness conditions are functionally composed is important, since they are not necessarily commutative. The following counter-example illustrates the issue for a relation that is not **BMH2**-healthy.

Counter-example 4

$\mathbf{bmh_2} \circ \mathbf{bmh_1}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp\}\})$ {Lemma B.1.8}

$= \{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp\} \lor ss = \emptyset\}$

$\mathbf{bmh_1} \circ \mathbf{bmh_2}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp\}\})$ $\{\text{Lemma B.1.7}\}$
$= \emptyset$

$\mathbf{bmh_2}$ and $\mathbf{bmh_3}$

Lemma B.1.9

$$\mathbf{bmh_2} \circ \mathbf{bmh_3}(B) = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s,\emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \land ((s,\emptyset) \in B \Rightarrow (s,\{\perp\}) \in B) \end{array}\right\}$$

Proof.

$$\begin{array}{ll} \mathbf{bmh_2} \circ \mathbf{bmh_3}(B) & \{\text{Definition of } \mathbf{bmh_2}\} \\ = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid (s, ss) \in \mathbf{bmh_3}(B) \land ((s,\{\perp\}) \in \mathbf{bmh_3}(B) \Leftrightarrow (s,\emptyset) \in \mathbf{bmh_3}(B)) \end{array}\right\} & \\ & \{\text{Definition of } \mathbf{bmh_3}\} \\ = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \begin{array}{l} (s, ss) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s,\emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B\} \\ \land \\ \left(\begin{array}{l} (s,\{\perp\}) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s,\emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B\} \\ \Leftrightarrow \\ (s,\emptyset) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s,\emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B\} \end{array}\right) \end{array} \end{array}\right\} & \\ & \{\text{Property of sets}\} \\ = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \begin{array}{l} (((s,\emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B) \\ \land \\ \left(\begin{array}{l} (((s,\emptyset) \in B \lor \perp \notin \{\perp\}) \land (s,\{\perp\}) \in B) \\ \Leftrightarrow \\ (((s,\emptyset) \in B \lor \perp \notin \emptyset) \land (s,\emptyset) \in B) \end{array}\right) \end{array} \end{array}\right\} & \\ & \{\text{Property of sets and predicate calculus}\} \end{array}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} (((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B) \\ \land \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \Leftrightarrow \\ ((s, \emptyset) \in B) \end{array} \right) \end{array} \right. \end{array} \right\} \qquad \{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \land ((s, \emptyset) \in B \Rightarrow (s, \{\perp\}) \in B) \end{array} \right\}$$

□

Lemma B.1.10

$$\mathbf{bmh_3} \circ \mathbf{bmh_2}(B)$$
$$=$$
$$\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\}$$

Proof.

$$\mathbf{bmh_3} \circ \mathbf{bmh_2}(B) \qquad \{\text{Definition of } \mathbf{bmh_3}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid ((s, \emptyset) \in \mathbf{bmh_2}(B) \lor \perp \notin ss) \land (s, ss) \in \mathbf{bmh_2}(B) \end{array} \right\} \qquad \{\text{Definition of } \mathbf{bmh_2}(B)\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \\ \lor \perp \notin ss \end{array} \right) \\ \land \\ (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \mid (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\} \end{array} \right. \end{array} \right\} \qquad \{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} (((s, \emptyset) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)) \lor \perp \notin ss) \\ \land \\ (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right. \end{array} \right\} \qquad \{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \lor \perp \notin ss) \\ \land \\ (((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \lor \perp \notin ss) \\ \land \\ (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right\}$$

□

The functions $\mathbf{bmh_2}$ and $\mathbf{bmh_3}$ are not in general commutative. The following counter-example illustrates the issue for a relation that is not **BMH2**-healthy.

Counter-example 5

$\mathbf{bmh_2} \circ \mathbf{bmh_3}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp\} \lor ss = \{s\}\})$

{Lemma B.1.9}

$= \{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{s\}\}$

$\mathbf{bmh_3} \circ \mathbf{bmh_2}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp\} \lor ss = \{s\}\})$

{Lemma B.1.10}

$= \emptyset$

$\mathbf{bmh_1}$ and $\mathbf{bmh_3}$

Lemma B.1.11

$$\mathbf{bmh_3} \circ \mathbf{bmh_1}(B)$$
$$=$$
$$\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \lor (s, \emptyset) \in B \lor \perp \notin ss) \\ \land \\ ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \end{array} \right. \end{array} \right\}$$

Proof.

$$\begin{aligned}
&\mathbf{bmh_3} \circ \mathbf{bmh_1}(B) && \{\text{Definition of } \mathbf{bmh_3}\} \\
&= \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, \emptyset) \in \mathbf{bmh_1}(B) \lor \perp \notin ss) \land (s, ss) \in \mathbf{bmh_1}(B)\} \\
& && \{\text{Definition of } \mathbf{bmh_1}\} \\
&= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} (s, \emptyset) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \\ \lor \\ \perp \notin ss \end{array} \right) \\ \land \\ (s, ss) \in \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B\} \end{array} \right. \end{array} \right\} \\
& && \{\text{Property of sets}\} \\
&= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \lor (s, \emptyset) \in B \lor \perp \notin ss) \\ \land \\ ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B) \end{array} \right. \end{array} \right\}
\end{aligned}$$

□

Lemma B.1.12

$$\mathbf{bmh_1} \circ \mathbf{bmh_3}(B) \\ = \\ \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B)) \\ \lor \\ (\perp \notin ss \land (s, ss) \in B) \end{array} \right. \end{array} \right\}$$

Proof.

$$\begin{aligned}
&\mathbf{bmh_1} \circ \mathbf{bmh_3}(B) && \{\text{Definition of } \mathbf{bmh_1}\} \\
&= \{s : State, ss : \mathbb{P}\, State_\perp \mid (s, ss \cup \{\perp\}) \in \mathbf{bmh_3}(B) \lor (s, ss) \in \mathbf{bmh_3}(B)\} \\
& && \{\text{Definition of } \mathbf{bmh_3}\}
\end{aligned}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \end{array} \right\} \\ \lor \\ (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid ((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B \end{array} \right\} \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (((s, \emptyset) \in B \lor \perp \notin (ss \cup \{\perp\})) \land (s, ss \cup \{\perp\}) \in B) \\ \lor \\ (((s, \emptyset) \in B \lor \perp \notin ss) \land (s, ss) \in B) \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets and predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land ((s, ss \cup \{\perp\}) \in B \lor (s, ss) \in B)) \\ \lor \\ (\perp \notin ss \land (s, ss) \in B) \end{array} \right. \end{array} \right\}$$

□

The functions $\mathbf{bmh_3}$ and $\mathbf{bmh_1}$ do not necessarily commute. The following counter-example shows this for a relation that is not **BMH3**-healthy. In fact, the functional application $\mathbf{bmh_3} \circ \mathbf{bmh_1}$ is not suitable as the counter-example shows that we have a fixed point.

Counter-example 6

$$\mathbf{bmh_3} \circ \mathbf{bmh_1}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp, s\} \lor ss = \{\perp\}\})$$

$$\{\text{Lemma B.1.11}\}$$

$$= \{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp, s\} \lor ss = \{\perp\}\}$$

$$\mathbf{bmh_1} \circ \mathbf{bmh_3}(\{s : State, ss : \mathbb{P}\, State_\perp \mid ss = \{\perp, s\} \lor ss = \{\perp\}\})$$

$$\{\text{Lemma B.1.12}\}$$

$$= \emptyset$$

$\mathbf{bmh_{0,1,3,2}}$

Lemma B.1.13 ($\mathbf{bmh_{0,1,3,2}}$-idempotent)

$$\mathbf{bmh_{0,1,3,2}} \circ \mathbf{bmh_{0,1,3,2}}(B) = \mathbf{bmh_{0,1,3,2}}(B)$$

Proof.

$$\mathbf{bmh_{0,1,3,2}} \circ \mathbf{bmh_{0,1,3,2}}(B) \qquad \{\text{Definition of } \mathbf{bmh_{0,1,3,2}}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in \mathbf{bmh_{0,1,3,2}}(B) \land (s,\{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B)) \\ \lor \\ \left(\begin{array}{l} (s,\{\perp\}) \notin \mathbf{bmh_{0,1,3,2}}(B) \land (s,\emptyset) \notin \mathbf{bmh_{0,1,3,2}}(B) \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Law B.2.13 and Law B.2.14}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in B \land (s,\{\perp\}) \in B \land (s,\emptyset) \in B \land (s,\{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} \neg\,((s,\emptyset) \in B \land (s,\{\perp\}) \in B) \land \neg\,((s,\emptyset) \in B \land (s,\{\perp\}) \in B) \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Predicate calculus and definition of } \mathbf{bmh_{0,1,3,2}}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in B \land (s,\{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} \neg\,((s,\emptyset) \in B \land (s,\{\perp\}) \in B) \\ \land \\ \exists\, ss_0 \bullet \\ \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s,\emptyset) \in B \land (s,\{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s,\{\perp\}) \notin B \land (s,\emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss\ \right) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right.$$

$$\{\text{Variable renaming and property of sets}\}$$

$$
= \left\{ \begin{array}{l}
s : State, ss : \mathbb{P}\, State_{\perp} \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\lor \\
\left(\begin{array}{l}
\neg\, ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\land \\
\exists\, ss_0 \bullet \\
\quad \left(\begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_1 \bullet \left(\ (s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \perp \notin ss_1 \land \perp \notin ss_0\ \right)
\end{array} \right)
\end{array} \right) \\
\land\ ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l}
s : State, ss : \mathbb{P}\, State_{\perp} \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\lor \\
\left(\begin{array}{l}
(\exists\, ss_0 \bullet ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \\
\lor \\
\exists\, ss_0 \bullet \left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_1 \bullet \left(\ (s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \perp \notin ss_1 \land \perp \notin ss_0\ \right) \\
\land\ ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus: quantifier scope}

$$
= \left\{ \begin{array}{l}
s : State, ss : \mathbb{P}\, State_{\perp} \\
\left| \begin{array}{l}
((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\
\lor \\
\left(\begin{array}{l}
(((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \exists\, ss_0 \bullet ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \\
\lor \\
\left(\begin{array}{l}
(s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\
\land \\
\exists\, ss_1, ss_0 \bullet \left(\begin{array}{l}
(s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \perp \notin ss_1 \land \perp \notin ss_0 \\
\land\ ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss
\end{array} \right)
\end{array} \right)
\end{array} \right)
\end{array} \right.
\end{array} \right\}
$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_1, ss_0 \bullet \left(\begin{array}{l} (s, ss_1) \in B \land ss_1 \subseteq ss_0 \land \perp \notin ss_1 \land \perp \notin ss_0 \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_1 \bullet \left(\; (s, ss_1) \in B \land ss_1 \subseteq ss \land \perp \notin ss_1 \land \perp \notin ss \; \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Definition of $\mathbf{bmh_{0,1,3,2}}$}

$$= \mathbf{bmh_{0,1,3,2}}(B)$$

□

Lemma B.1.14

$$\mathbf{bmh_{0,1,2}} \circ \mathbf{bmh_{0,1,3,2}}(B) = \mathbf{bmh_{0,1,3,2}}(B)$$

Proof.

$\mathbf{bmh_{0,1,2}} \circ \mathbf{bmh_{0,1,3,2}}(B)$ {Definition of $\mathbf{bmh_{0,1,2}}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \lor (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B)) \\ \land\, ((s, \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) \Leftrightarrow (s, \emptyset) \in \mathbf{bmh_{0,1,3,2}}(B)) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Law B.2.13 and Law B.2.14 and predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \\ ((s, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \lor (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B)) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} (s, ss_0) \in \mathbf{bmh_{0,1,3,2}}(B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \lor \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} (s, ss_0 \cup \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Law B.2.11 and Law B.2.12}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \\ \lor \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Definition of $\mathbf{bmh_{0,1,3,2}}(B)$}

$$= \mathbf{bmh_{0,1,3,2}}(B)$$

□

B.2 Auxiliary lemmas

Lemma B.2.1 *Provided B_0 and B_1 are* **BMH0** *and* **BMH1***-healthy.*

$$(B_0 \;\; ;_{BM_\perp} B_2) \sqcup_{BM_\perp} (B_1 \;\; ;_{BM_\perp} B_2) \sqsubseteq_{BM_\perp} ((B_0 \sqcup_{BM_\perp} B_1) \;\; ;_{BM_\perp} B_2)$$

Proof.

$((B_0 \sqcup_{BM_\perp} B_1) \ ;_{BM_\perp} B_2)$

$$\{\text{Assumption: } B_0 \text{ and } B_1 \text{ are } \mathbf{BMH0}\text{-healthy and Law 4.5.7}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in (B_0 \sqcup_{BM_\perp} B_1)\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in (B_0 \sqcup_{BM_\perp} B_1) \end{array}\right\} \end{array}\right)$$

$$\{\text{Definition of } \sqcup_{BM_\perp}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in (B_0 \cap B_1)\} \\ \cup \\ \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in (B_0 \cap B_1) \end{array}\right\} \end{array}\right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \{s_0 : State, ss_0 : \mathbb{P}\, State_\perp \mid (s_0, State_\perp) \in B_0 \land (s_0, State_\perp) \in B_1\} \\ \cup \\ \left\{ s_0 : State, ss_0 : \mathbb{P}\, State_\perp \;\middle|\; \begin{array}{l} (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0 \\ \land \\ (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1 \end{array} \right\} \end{array}\right)$$

$$\{\text{Property of sets}\}$$

$$= \left\{\begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s_0, State_\perp) \in B_0 \land (s_0, State_\perp) \in B_1) \\ \lor \\ \left(\begin{array}{l} (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0 \\ \land \\ (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1 \end{array}\right) \end{array}\right. \end{array}\right\}$$

$$\{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} (s_0, State_\perp) \in B_0 \\ \lor \\ (s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_0 \end{array} \right) \\ \land \\ \left(\begin{array}{l} (s_0, State_\perp) \in B_0 \\ \lor \\ (s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_1 \end{array} \right) \\ \land \\ \left(\begin{array}{l} (s_0, State_\perp) \in B_1 \\ \lor \\ (s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_0 \end{array} \right) \\ \land \\ \left(\begin{array}{l} (s_0, State_\perp) \in B_1 \\ \lor \\ (s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_1 \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Assumption: B_0 and B_1 are **BMH1**-healthy and **BMH0**-healthy}

$$= \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} ((s_0, State_\perp) \in B_0 \land (s_0, State) \in B_0) \\ \lor \\ ((s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_0 \land (s_0, State) \in B_0) \end{array} \right) \\ \land \\ \left(\begin{array}{l} ((s_0, State_\perp) \in B_0 \land (s_0, State) \in B_0) \\ \lor \\ ((s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_1 \land (s_0, State) \in B_1) \end{array} \right) \\ \land \\ \left(\begin{array}{l} ((s_0, State_\perp) \in B_1 \land (s_0, State) \in B_1) \\ \lor \\ ((s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_0 \land (s_0, State) \in B_0) \end{array} \right) \\ \land \\ \left(\begin{array}{l} ((s_0, State_\perp) \in B_1 \land (s_0, State) \in B_1) \\ \lor \\ ((s_0, \{ s_1 : State \mid (s_1, ss_0) \in B_2 \}) \in B_1 \land (s_0, State) \in B_1) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s_0, State) \in B_0 \land (s_0, State) \in B_1 \\ \land \\ ((s_0, State_\perp) \in B_0 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0) \\ \land \\ \left(\begin{array}{l} ((s_0, State_\perp) \in B_0 \land (s_0, State) \in B_0) \\ \lor \\ ((s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1 \land (s_0, State) \in B_1) \end{array} \right) \\ \land \\ \left(\begin{array}{l} ((s_0, State_\perp) \in B_1 \land (s_0, State) \in B_1) \\ \lor \\ ((s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0 \land (s_0, State) \in B_0) \end{array} \right) \\ \land \\ ((s_0, State_\perp) \in B_1 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1) \end{array} \right. \end{array} \right\}$$

{Property of sets and predicate calculus}

$$\sqsupseteq_{BM_\perp} \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s_0, State_\perp) \in B_0 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0) \\ \land \\ ((s_0, State_\perp) \in B_1 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid ((s_0, State_\perp) \in B_0 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_0) \end{array} \right\} \\ \cap \\ \left\{ \begin{array}{l} s_0 : State, ss_0 : \mathbb{P}\, State_\perp \\ \mid ((s_0, State_\perp) \in B_1 \lor (s_0, \{s_1 : State \mid (s_1, ss_0) \in B_2\}) \in B_1) \end{array} \right\} \end{array} \right)$$

{Assumption: B_0 and B_1 are **BMH0**-healthy and Law 4.5.7}

$= (B_0 \ ;_{BM_\perp} B_2) \cap (B_1 \ ;_{BM_\perp} B_2)$ {Definition of $\sqcup_{BM_\perp}$}

$= (B_0 \ ;_{BM_\perp} B_2) \sqcup_{BM_\perp} (B_1 \ ;_{BM_\perp} B_2)$

□

BMH0

Law B.2.1 *Provided B is* **BMH0***-healthy.*

$$\left(\begin{array}{l} \exists\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \\ \bullet\ ((s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \end{array} \right)$$

$$=$$

$$(\exists\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet (s_0, ss_1) \in B \land \perp \in ss_1)$$

Proof. (Implication)

$$\left(\begin{array}{l} \exists\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \\ \bullet ((s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1) \end{array}\right)$$

$$\{\text{Assumption: } B \text{ is } \mathbf{BMH0}\text{-healthy}\}$$

$$= \left(\begin{array}{l} \exists\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \\ \bullet ((s_0, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \in ss_0 \land \perp \in ss_1 \land (s_0, ss_1) \in B) \end{array}\right)$$

$$\{\text{Propositional calculus}\}$$

$$\Rightarrow \exists\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet (\perp \in ss_1 \land (s_0, ss_1) \in B)$$

□

Proof. (Reverse implication)

$$\exists\, s_0 : State, ss_1 : \mathbb{P}\, State_\perp \bullet (\perp \in ss_1 \land (s_0, ss_1) \in B)$$

$$\{\text{Propositional calculus: introduce fresh variable}\}$$

$$= \left(\begin{array}{l} \exists\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ (\perp \in ss_1 \land (s_0, ss_1) \in B \land ss_0 = ss_1 \land (s_0, ss_0) \in B \land \perp \in ss_0) \end{array}\right)$$

$$\{\text{Propositional calculus: weaken predicate}\}$$

$$\Rightarrow \left(\begin{array}{l} \exists\, s_0 : State, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet \\ (\perp \in ss_1 \land (s_0, ss_1) \in B \land ss_0 \subseteq ss_1 \land (s_0, ss_0) \in B \land \perp \in ss_0) \end{array}\right)$$

□

$\mathbf{bmh}_{\mathbf{0,1,2}}$

Law B.2.2

$$\mathbf{bmh}_{0,1,2}(B)$$

$$=$$

$$\left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \vee \\ \left(\begin{array}{l} ((s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B) \\ \wedge \\ \left(\begin{array}{l} (((s, ac') \in B \fatsemi ac \subseteq ss) \wedge \perp \notin ss) \\ \vee \\ ((s, ac' \cup \{\perp\}) \fatsemi ac \subseteq ss) \end{array} \right) \end{array} \right) \end{array} \right. \end{array}\right\}$$

Proof.

$$\begin{array}{ll} \mathbf{bmh}_{0,1,2}(B) & \{\text{Definition of } \mathbf{bmh}_{0,1,2}\} \\ = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \vee (s, ss_0 \cup \{\perp\}) \in B) \\ \wedge ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array}\right\} & \\ & \{\text{Predicate calculus}\} \\ = \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \in ss_0 \wedge \perp \in ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \\ \wedge\, ss_0 \subseteq ss \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right) \end{array} \right. \end{array}\right\} & \{\text{Lemma B.3.1}\} \end{array}$$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \in ss_0 \wedge \perp \in ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\} \qquad \{\text{Property of sets}\}
$$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \in ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

$$
\{\text{Predicate calculus: absorption law}\}
$$

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\} \qquad \{\text{Property of sets}\}
$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, (ss_0 \setminus \{\perp\}) \subseteq ss \wedge \perp \in ss_0 \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Introduce fresh variable}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, t, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, t = (ss_0 \setminus \{\perp\}) \wedge t \subseteq ss \wedge \perp \in ss_0 \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Lemma B.3.2}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \exists\, t, ss_0 \bullet (s, ss_0) \in B \\ \wedge\, (t \cup \{\perp\}) = ss_0 \wedge t \subseteq ss \wedge \perp \notin t \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{One-point rule}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} (\ \exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss\ ) \\ \vee \\ (\ \exists\, t \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss \wedge \perp \notin t\ ) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Type: $\perp \notin ss_0, t$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Variable renaming and substitution}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Predicate calculus}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \vee \\ ((s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B) \end{array} \right) \\ \wedge \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Instatiation: consider case where } t = \emptyset\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \left(\begin{array}{l} ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \vee \\ ((s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B) \end{array} \right) \\ \wedge \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \wedge t \subseteq ss) \\ \vee \\ (s, \emptyset) \in B \end{array} \right) \end{array} \right. \end{array} \right\}$$

$$\{\text{Predicate calculus: absorption law and distribution}\}$$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ \left(\begin{array}{l} ((s, \{\perp\}) \notin B \land (s, \emptyset) \notin B) \\ \land \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss) \\ \lor \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \land t \subseteq ss) \\ \lor \\ (s, \emptyset) \in B \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Instatiation: consider case where $t = \emptyset$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ \left(\begin{array}{l} ((s, \{\perp\}) \notin B \land (s, \emptyset) \notin B) \\ \land \\ \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss) \\ \lor \\ (\exists\, t : \mathbb{P}\, State \bullet (s, t \cup \{\perp\}) \in B \land t \subseteq ss) \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Variable renaming and definition of sequential composition}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ \left(\begin{array}{l} ((s, \{\perp\}) \notin B \land (s, \emptyset) \notin B) \\ \land \\ \left(\begin{array}{l} (((s, ac') \in B \,;\, ac \subseteq ss) \land \perp \notin ss) \\ \lor \\ ((s, ac' \cup \{\perp\}) \,;\, ac \subseteq ss) \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

$\square$

Law B.2.3

$$(s, ss) \in \mathbf{bmh_{0,1,2}}(B) =$$

Proof.

$(s, ss) \in \mathbf{bmh_{0,1,2}}(B)$ {Definition of $\mathbf{bmh_{0,1,2}}(B)$}

$$= (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \quad \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \quad \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \end{array} \right)$$

□

Law B.2.4

$$\exists\, ss_1 \bullet (s, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss)$$
$$=$$
$$\left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

Proof.

$$\exists\, ss_1 \bullet (s, ss_1) \in \mathbf{bmh_{0,1,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss)$$

{Definition of $\mathbf{bmh_{0,1,2}}$}

$$= \exists\, ss_1 \bullet \left(\begin{array}{l} (s, ss_1) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \quad \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \quad \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right\} \\ \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets}

$$= \exists\, ss_1 \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land ss_0 \subseteq ss_1 \land (\perp \in ss_0 \Leftrightarrow \perp \in ss_1) \end{array} \right) \\ \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Predicate calculus: quantifier scope}

$$= \left(\begin{array}{l} ((s,\{\bot\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\ \wedge \\ \exists\, ss_1 \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet ((s,ss_0) \in B \vee (s, ss_0 \cup \{\bot\}) \in B) \\ \wedge\, ss_0 \subseteq ss_1 \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss_1) \end{array} \right) \\ \wedge\, ss_1 \subseteq ss \wedge (\bot \in ss_1 \Leftrightarrow \bot \in ss) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} ((s,\{\bot\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \exists\, ss_0, ss_1 \bullet \left(\begin{array}{l} (s,ss_0) \in B \\ \wedge\, ss_0 \subseteq ss_1 \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss_1) \\ \wedge\, ss_1 \subseteq ss \wedge (\bot \in ss_1 \Leftrightarrow \bot \in ss) \end{array} \right) \\ \vee \\ \exists\, ss_0, ss_1 \bullet \left(\begin{array}{l} (s, ss_0 \cup \{\bot\}) \in B \\ \wedge\, ss_0 \subseteq ss_1 \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss_1) \\ \wedge\, ss_1 \subseteq ss \wedge (\bot \in ss_1 \Leftrightarrow \bot \in ss) \end{array} \right) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} ((s,\{\bot\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \exists\, ss_0 \bullet ((s,ss_0) \in B \wedge ss_0 \subseteq ss \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss)) \\ \vee \\ \exists\, ss_0 \bullet ((s, ss_0 \cup \{\bot\}) \in B \wedge ss_0 \subseteq ss \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss)) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} ((s,\{\bot\}) \in B \Leftrightarrow (s,\emptyset) \in B) \\ \wedge \\ \exists\, ss_0 \bullet ((s,ss_0) \in B \vee (s, ss_0 \cup \{\bot\}) \in B) \wedge ss_0 \subseteq ss \wedge (\bot \in ss_0 \Leftrightarrow \bot \in ss) \end{array} \right)$$

$\square$

Law B.2.5

$$(s,\emptyset) \in \mathbf{bmh_{0,1,2}}(B) = (s,\emptyset) \in B \wedge (s,\{\bot\}) \in B$$

Proof.

$(s,\emptyset) \in \mathbf{bmh_{0,1,2}}(B)$ {Definition of $\mathbf{bmh_{0,1,2}}$}

$$= (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq \emptyset \land (\perp \in ss_0 \Leftrightarrow \perp \in \emptyset) \end{array} \right) \qquad \text{\{Predicate calculus\}}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq \emptyset \land \perp \notin ss_0 \end{array} \right)$$

{Case analysis on ss_0 and one-point rule}

$$= ((s, \emptyset) \in B \lor (s, \emptyset \cup \{\perp\}) \in B) \land ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B)$$

{Property of sets and predicate calculus}

$$= (s, \{\perp\}) \in B \land (s, \emptyset) \in B$$

□

Law B.2.6

$$(s, \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) = (s, \emptyset) \in B \land (s, \{\perp\}) \in B$$

Proof.

$$(s, \{\perp\}) \in \mathbf{bmh_{0,1,2}}(B) \qquad \text{\{Definition of } \mathbf{bmh_{0,1,2}}\text{\}}$$

$$= (s, \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq \{\perp\} \land (\perp \in ss_0 \Leftrightarrow \perp \in \{\perp\}) \end{array} \right) \qquad \text{\{Predicate calculus\}}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land\, ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\, ss_0 \subseteq \{\perp\} \land \perp \in ss_0 \end{array} \right)$$

{Case analysis on ss_0 and one-point rule}

$= ((s, \{\bot\}) \in B \lor (s, \{\bot\} \cup \{\bot\}) \in B) \land ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B)$

{Property of sets and predicate calculus}

$= (s, \{\bot\}) \in B \land (s, \emptyset) \in B$

□

Law B.2.7

$$B_1 \subseteq B_0$$
$$\Leftrightarrow$$
$$\forall\, s : State, ss : \mathbb{P}\, State \bullet \left(\begin{array}{l} (s, ss) \in B_1 \Rightarrow (s, ss) \in B_0 \\ \land \\ (s, ss \cup \{\bot\}) \in B_1 \Rightarrow (s, ss \cup \{\bot\}) \in B_0 \end{array} \right)$$

Proof.

$B_1 \subseteq B_0$ {Definition of subset inclusion}

$\Leftrightarrow \forall\, s : State, ss : \mathbb{P}\, State_\bot \bullet (s, ss) \in B_1 \Rightarrow (s, ss) \in B_0$

{Predicate calculus}

$\Leftrightarrow \forall\, s : State, ss : \mathbb{P}\, State_\bot \bullet ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \land (\bot \in ss \lor \bot \notin ss)$

{Predicate calculus}

$$\Leftrightarrow \forall\, s : State, ss : \mathbb{P}\, State_\bot \bullet \left(\begin{array}{l} (\bot \in ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \\ \land \\ (\bot \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \end{array} \right)$$

{Introduce fresh variable}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\bot \bullet \\ \left(\begin{array}{l} \left(\begin{array}{l} (\exists\, t : \mathbb{P}\, State_\bot \bullet \bot \in ss \land t = ss \setminus \{\bot\}) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \\ \land \\ (\bot \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \end{array} \right) \end{array} \right)$$

{Lemma B.3.2}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State}, ss : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} \left(\begin{array}{l} (\exists\, t : \mathbb{P}\, \mathit{State}_\perp \bullet \perp \notin t \land ss = t \cup \{\perp\}) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \\ \land \\ (\perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \end{array}\right) \end{array}\right)$$

{Predicate calculus: quantifier scope}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State};\ ss, t : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} ((\perp \notin t \land ss = t \cup \{\perp\}) \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \\ \land \\ (\perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \end{array}\right) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State};\ ss, t : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} (\perp \notin t \Rightarrow (ss = t \cup \{\perp\} \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \\ \land \\ (\perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \end{array}\right) \end{array}\right)$$

{Variable renaming}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State};\ ss, t : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} (\perp \notin t \Rightarrow (ss = t \cup \{\perp\} \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0)) \\ \land \\ (\perp \notin t \Rightarrow ((s, t) \in B_1 \Rightarrow (s, t) \in B_0)) \end{array}\right) \end{array}\right)$$

{Predicate calculus: quantifier scope}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State}, t : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} (\perp \notin t \Rightarrow \forall\, ss : \mathbb{P}\, \mathit{State}_\perp \bullet \left(\begin{array}{l} (ss = t \cup \{\perp\}) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \\ \land \\ (\perp \notin t \Rightarrow ((s, t) \in B_1 \Rightarrow (s, t) \in B_0)) \end{array}\right) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \forall\, s : \mathit{State}, t : \mathbb{P}\, \mathit{State}_\perp \bullet \\ \left(\begin{array}{l} (\perp \notin t \Rightarrow ((s, t \cup \{\perp\}) \in B_1 \Rightarrow (s, t \cup \{\perp\}) \in B_0)) \\ \land \\ (\perp \notin t \Rightarrow ((s, t) \in B_1 \Rightarrow (s, t) \in B_0)) \end{array}\right) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \forall s : State, t : \mathbb{P}\, State \bullet \left(\begin{array}{l} (s, t \cup \{\bot\}) \in B_1 \Rightarrow (s, t \cup \{\bot\}) \in B_0 \\ \land \\ (s, t) \in B_1 \Rightarrow (s, t) \in B_0 \end{array} \right)$$

□

$\mathbf{bmh_{0,1,3}}$

Law B.2.8

$$\begin{array}{l} \mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_3}(B) \\ = \\ \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\bot\}) \in B) \\ \land \\ (s, \emptyset) \in B \land ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \end{array} \right) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss) \end{array} \right. \end{array} \right\} \end{array}$$

Proof.

$\mathbf{bmh_0} \circ \mathbf{bmh_1} \circ \mathbf{bmh_3}(B)$ {Definition of $\mathbf{bmh_0} \circ \mathbf{bmh_1}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in \mathbf{bmh_3}(B) \lor (s, ss_0 \cup \{\bot\}) \in \mathbf{bmh_3}(B)) \\ \land\, ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \end{array} \right. \end{array} \right\}$$

{Definition of $\mathbf{bmh_3}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid ((s, \emptyset) \in B \lor \bot \notin ss) \land (s, ss) \in B \end{array} \right\} \\ \lor \\ (s, ss_0 \cup \{\bot\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid ((s, \emptyset) \in B \lor \bot \notin ss) \land (s, ss) \in B \end{array} \right\} \end{array} \right) \\ \land\, ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} (((s, \emptyset) \in B \lor \perp \notin ss_0) \land (s, ss_0) \in B) \\ \lor \\ (((s, \emptyset) \in B \lor \perp \notin (ss_0 \cup \{\perp\})) \land (s, ss_0 \cup \{\perp\}) \in B) \end{array} \right) \\ \land\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets and predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} (((s, \emptyset) \in B \lor \perp \notin ss_0) \land (s, ss_0) \in B) \\ \lor \\ (((s, \emptyset) \in B \lor \mathit{false}) \land (s, ss_0 \cup \{\perp\}) \in B) \end{array} \right) \\ \land\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, ss_0) \in B) \\ \lor \\ (\perp \notin ss_0 \land (s, ss_0) \in B) \\ \lor \\ ((s, \emptyset) \in B \land (s, ss_0 \cup \{\perp\}) \in B) \end{array} \right) \\ \land\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land \\ (s, \emptyset) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \lor \\ \exists\, ss_0 \bullet (\perp \notin ss_0 \land (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_{\perp} \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land \\ (s, \emptyset) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right) \\ \lor \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right. \end{array} \right\}$$

$\square$

Law B.2.9

$$\exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land \\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right)$$
$$=$$
$$\exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land \\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right) \lor (s, \{\perp\}) \in B$$

Proof.

$$\exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\perp\}) \in B) \\ \land \\ ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right) \qquad \{\text{Predicate calculus}\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array}\right)$$

$$\{\text{Instantiation of existential quantification for } ss_0 = \{\perp\} \text{ and } ss_0 = \emptyset\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ ((s, \{\perp\} \cup \{\perp\}) \in B \land \{\perp\} \subseteq ss \land (\perp \in \{\perp\} \Leftrightarrow \perp \in ss)) \\ \lor \\ ((s, \emptyset \cup \{\perp\}) \in B \land \emptyset \subseteq ss \land (\perp \in \emptyset \Leftrightarrow \perp \in ss)) \end{array}\right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ ((s, \{\perp\}) \in B \land \{\perp\} \subseteq ss \land \perp \in ss) \\ \lor \\ ((s, \{\perp\}) \in B \land \perp \notin ss) \end{array}\right)$$

$$\{\text{Lemma B.3.3 and predicate calculus}\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \\ \lor \\ ((s, \{\bot\}) \in B \land \bot \in ss) \\ \lor \\ ((s, \{\bot\}) \in B \land \bot \notin ss) \end{array} \right)$$

{Predicate calculus}

$$= \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, ss_0) \in B \lor (s, ss_0 \cup \{\bot\}) \in B) \\ \land \\ ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \end{array} \right) \lor (s, \{\bot\}) \in B$$

□

$\mathbf{bmh_{0,1,3,2}}$

Law B.2.10

$$\begin{array}{l} (s, ss) \in \mathbf{bmh_{0,1,3,2}}(B) \\ = \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss) \end{array} \right) \end{array} \right) \end{array}$$

Proof.

$(s, ss) \in \mathbf{bmh_{0,1,3,2}}(B)$ {Definition of $\mathbf{bmh_{0,1,3,2}}$}

$$= (s, ss) \in \left\{ \begin{array}{l|l} \multicolumn{2}{l}{s : State, ss : \mathbb{P}\, State_\bot} \\ & ((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \\ & \lor \\ & \left(\begin{array}{l} (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss) \end{array} \right) \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right) \end{array} \right)$$

$\square$

Law B.2.11

$$\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet (s, ss_1 \cup \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \\ \Leftrightarrow \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \end{array}$$

Proof.

$\exists\, ss_1 : \mathbb{P}\, State_\perp \bullet (s, ss_1 \cup \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss)$

{Definition of $\mathbf{bmh_{0,1,3,2}}$}

$$\Leftrightarrow \left(\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet \\ (s, ss_1 \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \land \\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets}

$$\Leftrightarrow \left(\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet \left(\begin{array}{l} (s, ss_0) \in B \land ss_0 \subseteq (ss_1 \cup \{\perp\}) \\ \land \perp \notin ss_0 \land \perp \notin (ss_1 \cup \{\perp\}) \end{array} \right) \end{array} \right) \end{array} \right) \\ \land \\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets and predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \exists ss_1 : \mathbb{P}\, State_\perp \bullet \\ ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right)$$

{Predicate calculus: instatiation of existential quantifier for $ss_1 = ss$}

$$\Leftrightarrow ((s, \emptyset) \in B \land (s, \{\perp\}) \in B)$$

□

Law B.2.12

$$\begin{array}{l} \exists ss_1 : \mathbb{P}\, State_\perp \bullet (s, ss_1) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \\ \Leftrightarrow \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array}\right) \end{array}\right) \end{array}$$

Proof.

$$\exists ss_1 : \mathbb{P}\, State_\perp \bullet (s, ss_1) \in \mathbf{bmh_{0,1,3,2}}(B) \land ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss)$$

{Definition of $\mathbf{bmh_{0,1,3,2}}$}

$$\Leftrightarrow \left(\begin{array}{l} \exists ss_1 : \mathbb{P}\, State_\perp \bullet \\ (s, ss_1) \in \left\{\begin{array}{l|l} s : State, ss : \mathbb{P}\, State_\perp & \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \end{array}\right) \end{array} \end{array}\right\} \\ \land \\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right)$$

{Property of sets}

$$\Leftrightarrow \left(\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet \\ \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet ((s, ss_0) \in B \land ss_0 \subseteq ss_1 \land \perp \notin ss_0 \land \perp \notin ss_1) \end{array}\right) \end{array}\right) \\ \land \\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \land\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \left(\begin{array}{l} \exists\, ss_0, ss_1 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss_1 \\ \land \perp \notin ss_0 \land \perp \notin ss_1 \\ \land\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right) \end{array}\right) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_1 : \mathbb{P}\, State_\perp \bullet ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \land\ ss_1 \subseteq ss \land (\perp \in ss_1 \Leftrightarrow \perp \in ss) \end{array}\right) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array}\right) \end{array}\right)$$

{Predicate calculus: instatiation of existential quantifier for $ss_1 = ss$}

$$\Leftrightarrow \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array}\right) \end{array}\right)$$

$\square$

Law B.2.13

$$(s, \emptyset) \in \mathbf{bmh_{0,1,3,2}}(B) = (s, \emptyset) \in B \land (s, \{\perp\}) \in B$$

Proof.

$$
\begin{array}{ll}
(s, \emptyset) \in \mathbf{bmh_{0,1,3,2}}(B) & \{\text{Definition of } \mathbf{bmh_{0,1,3,2}}\} \\
= (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\} & \\
& \{\text{Property of sets}\} \\
= \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq \emptyset \land \perp \notin ss_0 \land \perp \notin \emptyset \ \right) \end{array} \right) \end{array} \right) & \\
& \{\text{Property of sets and one-point rule}\} \\
= \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ ((s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \land (s, \emptyset) \in B) \end{array} \right) & \{\text{Predicate calculus}\} \\
= (s, \emptyset) \in B \land (s, \{\perp\}) \in B &
\end{array}
$$

□

Law B.2.14

$$(s, \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) = (s, \emptyset) \in B \land (s, \{\perp\}) \in B$$

Proof.

$$
\begin{array}{ll}
(s, \{\perp\}) \in \mathbf{bmh_{0,1,3,2}}(B) & \{\text{Definition of } \mathbf{bmh_{0,1,3,2}}\} \\
= (s, \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\} & \\
& \{\text{Property of sets}\}
\end{array}
$$

$$= \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq \{\bot\} \land \bot \notin ss_0 \land \bot \notin \{\bot\} \ \right) \end{array} \right) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\bot\}) \in B) \\ \lor \\ ((s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \land \mathit{false}) \end{array} \right)$$

{Predicate calculus}

$$= (s, \emptyset) \in B \land (s, \{\bot\}) \in B$$

□

Law B.2.15 *Provided B is* **BMH0** *and* **BMH2**-*healthy.*

$$B = (B \rhd \{ss : \mathbb{P}\, State_\bot \mid \bot \in ss\}) \cup \{s_0 : State, ss : \mathbb{P}\, State_\bot \mid (s_0, \emptyset) \in B\}$$
$$\Leftrightarrow$$
$$\mathbf{BMH3}$$

Proof.

$$B = (B \rhd \{ss : \mathbb{P}\, State_\bot \mid \bot \in ss\}) \cup \{s_0 : State, ss : \mathbb{P}\, State_\bot \mid (s_0, \emptyset) \in B\}$$

{Property of sets}

$$\Leftrightarrow (B = \{s : State, ss : State_\bot \mid ((s, ss) \in B \land \bot \notin ss) \lor (s, \emptyset) \in B\})$$

{Property of sets}

$$\Leftrightarrow \forall s, ss \bullet \left(\begin{array}{l} (s, ss) \in B \Rightarrow (((s, ss) \in B \land \bot \notin ss) \lor (s, \emptyset) \in B) \\ \land \\ ((((s, ss) \in B \land \bot \notin ss) \lor (s, \emptyset) \in B) \Rightarrow (s, ss) \in B) \end{array} \right)$$

{Propositional calculus}

$$\Leftrightarrow \forall s, ss \bullet \left(\begin{array}{l} ((s, \emptyset) \notin B \Rightarrow ((s, ss) \notin B \lor ((s, ss) \in B \land \bot \notin ss))) \\ \land \\ ((((s, ss) \notin B \lor \bot \in ss) \land (s, \emptyset) \notin B) \lor (s, ss) \in B) \end{array} \right)$$

{Propositional calculus: absorption law}

$$\Leftrightarrow \forall s, ss \bullet \left(\begin{array}{l} ((s, \emptyset) \notin B \Rightarrow ((s, ss) \notin B \vee \bot \notin ss)) \\ \wedge \\ ((s, \emptyset) \notin B \vee (s, ss) \in B) \end{array} \right)$$

{Propositional calculus}

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet (s, \emptyset) \in B \Rightarrow (s, ss) \in B \end{array} \right)$$

{Propositional calculus: introduce term}

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet (s, \emptyset) \in B \Rightarrow ((s, ss) \in B \vee (\bot \in ss \wedge \bot \notin ss)) \end{array} \right)$$

{Propositional calculus}

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet (s, \emptyset) \in B \Rightarrow ((s, ss) \in B \vee \bot \in ss) \\ \wedge \\ \forall s, ss \bullet (s, \emptyset) \in B \Rightarrow ((s, ss) \in B \vee \bot \notin ss) \end{array} \right)$$

{Propositional calculus}

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet ((s, \emptyset) \in B \wedge \bot \notin ss) \Rightarrow (s, ss) \in B \\ \wedge \\ \forall s, ss \bullet ((s, \emptyset) \in B \wedge \bot \in ss) \Rightarrow (s, ss) \in B \end{array} \right) \quad \text{\{Property of sets\}}$$

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet ((s, \emptyset) \in B \wedge \emptyset \subseteq ss \wedge \bot \notin \emptyset \wedge \bot \notin ss) \Rightarrow (s, ss) \in B \\ \wedge \\ \forall s, ss \bullet ((s, \emptyset) \in B \wedge \bot \in ss) \Rightarrow (s, ss) \in B \end{array} \right)$$

{Assumption: B is **BMH2**-healthy and Lemma B.3.3}

$$\Leftrightarrow \left(\begin{array}{l} \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss) \\ \wedge \\ \forall s, ss \bullet ((s, \emptyset) \in B \wedge \emptyset \subseteq ss \wedge \bot \notin \emptyset \wedge \bot \notin ss) \Rightarrow (s, ss) \in B \\ \wedge \\ \forall s, ss \bullet ((s, \{\bot\}) \in B \wedge \{\bot\} \subseteq ss \wedge \bot \in \{\bot\} \wedge \bot \in ss) \Rightarrow (s, ss) \in B \end{array}\right)$$

{Assumption: B is **BMH0**-healthy}

$$\Leftrightarrow \forall s, ss \bullet (s, \emptyset) \notin B \Rightarrow ((s, ss) \in B \Rightarrow \bot \notin ss)$$

{Propositional calculus: move quantifier}

$$\Leftrightarrow \forall s \bullet (s, \emptyset) \notin B \Rightarrow \forall ss \bullet ((s, ss) \in B \Rightarrow \bot \notin ss)$$

{Definition of **BMH3**}

$$\Leftrightarrow \mathbf{BMH3}$$

□

bmb2bm

Law B.2.16

$$bm2bmb(\mathbf{bmh_{upclosed}}(B))$$
$$=$$
$$\left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \exists ss_0 \bullet (s, ss_0) \in B \wedge \bot \notin ss_0 \wedge ss_0 \subseteq ss \wedge \bot \notin ss \\ \vee \\ (s, \emptyset) \in B \end{array}\right. \end{array}\right\}$$

Proof.

$bm2bmb(\mathbf{bmh_{upclosed}}(B))$ {Definition of $bm2bmb$}

$$= \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid ((s, ss) \in \mathbf{bmh_{upclosed}}(B) \wedge \bot \notin ss) \vee (s, \emptyset) \in \mathbf{bmh_{upclosed}}(B) \end{array}\right\}$$

{Definition of $\mathbf{bmh_{upclosed}}$}

$$= \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \left| \begin{array}{l} \left((s, ss) \in \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid \exists ss_0 \bullet (s, ss_0) \in B \wedge \bot \notin ss_0 \wedge ss_0 \subseteq ss \wedge \bot \notin ss \end{array}\right\} \wedge \bot \notin ss \right) \\ \vee \\ (s, \emptyset) \in \left\{\begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid \exists ss_0 \bullet (s, ss_0) \in B \wedge \bot \notin ss_0 \wedge ss_0 \subseteq ss \wedge \bot \notin ss \end{array}\right\} \end{array}\right. \end{array}\right\}$$

{Property of sets and predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq \emptyset \end{array} \right. \end{array} \right\}$$

{Case-analysis on ss_0 and one-point rule}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\}$$

$\square$

Theorem B.2.1

$$\mathbf{bmh_{0,1,3,2}} \circ bm2bmb(\mathbf{bmh_{upclosed}}(B)) = bm2bmb(\mathbf{bmh_{upclosed}}(B))$$

Proof.

$\mathbf{bmh_{0,1,3,2}} \circ bm2bmb(\mathbf{bmh_{upclosed}}(B))$ {Definition of $\mathbf{bmh_{0,1,3,2}}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \land (s, \{\perp\}) \in bm2bmb(\mathbf{bmh_{upclosed}}(B))) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin bm2bmb(\mathbf{bmh_{upclosed}}(B)) \land (s, \emptyset) \notin bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Law B.2.19 and Law B.2.18}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \emptyset) \in B) \\ \lor \\ \left(\begin{array}{l} ((s, \emptyset) \notin B \land (s, \emptyset) \notin B) \\ \land \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in bm2bmb(\mathbf{bmh_{upclosed}}(B)) \\ \land\, ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus and definition of $bm2bmb(\mathbf{bmh_{upclosed}}(B))$ (Law B.2.16)}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \wedge \perp \notin ss_0 \wedge ss_0 \subseteq ss \wedge \perp \notin ss \\ \vee \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\} \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Variable renaming and property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ \exists\, ss_0 \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_0 \\ \vee \\ (s, \emptyset) \in B \end{array} \right) \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ \left(\begin{array}{l} \exists\, ss_0, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_0 \\ \wedge\, ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\exists\, ss_0 \bullet (s, \emptyset) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \right) \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ (\exists\, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss \wedge \perp \notin ss) \\ \vee \\ ((s, \emptyset) \in B \wedge \exists\, ss_0 \bullet ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss) \end{array} \right. \end{array} \right\}$$

{Predicate calculus: absorption law}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, \emptyset) \in B \\ \vee \\ (\exists\, ss_1 \bullet (s, ss_1) \in B \wedge \perp \notin ss_1 \wedge ss_1 \subseteq ss \wedge \perp \notin ss) \end{array} \right. \end{array} \right\}$$

$\{\text{Law B.2.16}\}$

$$= bm2bmb(\mathbf{bmh_{upclosed}}(B))$$

$\square$

Law B.2.17

$$bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$$
$$=$$
$$\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \wedge \perp \notin ss \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

Proof.

$bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$ $\{\text{Definition of } bmb2bm\}$

$= \{s : State, ss : \mathbb{P}\, State_\perp \mid ((s, ss) \in \mathbf{bmh_{0,1,3,2}}(B) \wedge \perp \notin ss)\}$

$\{\text{Definition of } \mathbf{bmh_{0,1,3,2}}\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (s, ss) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \wedge ss_0 \subseteq ss \wedge \perp \notin ss_0 \wedge \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \wedge \perp \notin ss \end{array} \right. \end{array} \right\}$$

$\{\text{Property of sets}\}$

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right) \\ \land \perp \notin ss \end{array} \right. \end{array} \right\}$$

{Predicate calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin ss \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\}$$

□

Theorem B.2.2 (bmb2bm-is-bmh$_{\mathbf{upclosed}}$)

$$\mathbf{bmh_{upclosed}} \circ bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) = bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$$

Proof.

$\mathbf{bmh_{upclosed}} \circ bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$ {Definition of $\mathbf{bmh_{upclosed}}$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \mid \exists\, ss_0 \bullet (s, ss_0) \in bmb2bm(\mathbf{bmh_{0,1,3,2}}(B)) \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right\}$$

{Law B.2.17}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \land (s, \{\perp\}) \in B) \land \perp \notin ss \\ \lor \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \\ \land \\ \exists\, ss_0 \bullet \left(\ (s, ss_0) \in B \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \ \right) \end{array} \right) \end{array} \right. \end{array} \right\} \\ \land \\ \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \end{array} \right. \end{array} \right\}$$

{Variable renaming and property of sets}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet \left(\begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \wedge \perp \notin ss_0 \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_1 \bullet \left(\ (s, ss_1) \in B \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_1 \wedge \perp \notin ss_0\ \right) \end{array} \right) \end{array} \right) \\ \wedge \\ \perp \notin ss_0 \wedge ss_0 \subseteq ss \wedge \perp \notin ss \end{array} \right. \end{array} \right\}
$$

{Predicate calculus: distributivity and quantifier scope}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \wedge \perp \notin ss \wedge \exists\, ss_0 \bullet \perp \notin ss_0 \wedge ss_0 \subseteq ss \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_1, ss_0 \bullet \left(\begin{array}{l} (s, ss_1) \in B \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_1 \wedge \perp \notin ss_0 \\ \wedge \\ \perp \notin ss_0 \wedge ss_0 \subseteq ss \wedge \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Predicate calculus: case-analysis on ss_0}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \wedge \perp \notin ss \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_1, ss_0 \bullet \left(\begin{array}{l} (s, ss_1) \in B \wedge ss_1 \subseteq ss_0 \wedge \perp \notin ss_1 \wedge \perp \notin ss_0 \\ \wedge \\ \perp \notin ss_0 \wedge ss_0 \subseteq ss \wedge \perp \notin ss \end{array} \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Predicate calculus}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((s, \emptyset) \in B \wedge (s, \{\perp\}) \in B) \wedge \perp \notin ss \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \\ \wedge \\ \exists\, ss_1 \bullet \left(\ (s, ss_1) \in B \wedge ss_1 \subseteq ss \wedge \perp \notin ss_1 \wedge \perp \notin ss\ \right) \end{array} \right) \end{array} \right. \end{array} \right\}
$$

{Law B.2.17}

$= bmb2bm(\mathbf{bmh_{0,1,3,2}}(B))$

□

Law B.2.18

$$(s, \emptyset) \in bmb2bm(\mathbf{bmh_{upclosed}}) = (s, \emptyset) \in B$$

Proof.

$(s, \emptyset) \in bmb2bm(\mathbf{bmh_{upclosed}})$

$$\{\text{Definition of } bmb2bm(\mathbf{bmh_{upclosed}}) \text{ (Law B.2.16)}\}$$

$$= (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq \emptyset \land \perp \notin \emptyset \\ \lor \\ (s, \emptyset) \in B \end{array} \right)$$

$$\{\text{Predicate calculus and one-point rule}\}$$

$$= (s, \emptyset) \in B$$

□

Law B.2.19

$$(s, \{\perp\}) \in bmb2bm(\mathbf{bmh_{upclosed}}) = (s, \emptyset) \in B$$

Proof.

$(s, \{\perp\}) \in bmb2bm(\mathbf{bmh_{upclosed}})$

$$\{\text{Definition of } bmb2bm(\mathbf{bmh_{upclosed}}) \text{ (Law B.2.16)}\}$$

$$= (s, \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \perp \notin ss_0 \land ss_0 \subseteq ss \land \perp \notin ss \\ \lor \\ (s, \emptyset) \in B \end{array} \right. \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land \bot \notin ss_0 \land ss_0 \subseteq \{\bot\} \land \bot \notin \{\bot\} \\ \lor \\ (s, \emptyset) \in B \end{array} \right)$$

{Property of sets and predicate calculus}

$$= (s, \emptyset) \in B$$

□

B.3 Set theory

Lemma B.3.1

$$\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss) \\ \Leftrightarrow \\ \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\bot\}) \land \bot \in ss_0 \end{array}$$

Proof.

$$\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land (\bot \in ss_0 \Leftrightarrow \bot \in ss)$$

{Predicate calculus}

$$= \exists\, ss_0 \bullet \left(\begin{array}{l} (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land \bot \in ss_0 \land \bot \in ss \\ \lor \\ (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss \end{array} \right)$$

{Predicate calculus and property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\bot\}) \land \bot \in ss_0 \land \bot \in ss \\ \lor \\ \exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss \end{array} \right)$$

{Introduce fresh variable t and substitution}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\bot\}) \land \bot \in ss_0 \land \bot \in ss \\ \lor \\ \exists\, t, ss_0 \bullet (s, t) \in B \land t = ss_0 \cup \{\bot\} \land ss_0 \subseteq ss \land \bot \notin ss_0 \land \bot \notin ss \end{array} \right)$$

{Property of sets (Lemma B.3.2)}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ss \cup \{\bot\}) \land \bot \in ss_0 \land \bot \in ss \\ \lor \\ \exists\, t, ss_0 \bullet (s, t) \in B \land t \setminus \{\bot\} = ss_0 \land ss_0 \subseteq ss \land \bot \in t \land \bot \notin ss \end{array} \right)$$

{One-point rule and subsitutiton}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \in ss \\ \vee \\ \exists\, t \bullet (s, t) \in B \wedge (t \setminus \{\perp\}) \subseteq ss \wedge \perp \in t \wedge \perp \notin ss \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \in ss \\ \vee \\ \exists\, t \bullet (s, t) \in B \wedge t \subseteq (ss \cup \{\perp\}) \wedge \perp \in t \wedge \perp \notin ss \end{array} \right)$$

{Rename variables}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \in ss \\ \vee \\ \exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \notin ss \end{array} \right)$$

{Predicate calculus}

$$= \exists\, ss_0 \bullet ((s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0) \wedge (\perp \in ss \vee \perp \notin ss)$$

{Propositional calculus}

$$= \exists\, ss_0 \bullet ((s, ss_0) \in B \wedge ss_0 \subseteq (ss \cup \{\perp\}) \wedge \perp \in ss_0)$$

□

Lemma B.3.2 (A-setminus-x)

$$(A = B \cup \{x\} \wedge x \notin B) \Leftrightarrow (A \setminus \{x\} = B \wedge x \in A)$$

Proof.

$A = B \cup \{x\} \wedge x \notin B$ {Set equality}

$= (\forall\, y \bullet y \in A \Leftrightarrow y \in (B \cup \{x\})) \wedge x \notin B$ {Propositional calculus}

$= (\forall\, y \bullet (y \in A \Rightarrow y \in (B \cup \{x\})) \wedge (y \in (B \cup \{x\})) \Rightarrow y \in A) \wedge x \notin B$

{Property of sets}

$$= \left(\begin{array}{l} (\forall\, y \bullet (y \in A \Rightarrow (y \in B \vee y \in \{x\})) \wedge ((y \in B \vee y \in \{x\}) \Rightarrow y \in A)) \\ \wedge\, x \notin B \end{array} \right)$$

{Propositional calculus}

$$= \left(\begin{array}{l} \forall\, y \bullet \left(\begin{array}{l} ((y \in A \wedge y \notin \{x\}) \Rightarrow y \in B) \\ \wedge\, (y \in B \Rightarrow y \in A) \wedge (y \in \{x\} \Rightarrow y \in A) \end{array} \right) \\ \wedge\, x \notin B \end{array} \right)$$

{Lemma B.3.4 and propositional calculus}

$$= \left(\forall\, y \bullet \left(\begin{array}{l} ((y \in A \wedge y \notin \{x\}) \Rightarrow y \in B) \\ \wedge\, (y \in B \Rightarrow y \in A) \wedge (y \in \{x\} \Rightarrow y \in A) \wedge (y \in B \Rightarrow y \notin \{x\}) \end{array} \right) \right)$$

{Propositional calculus}

$= \big(\forall y \bullet ((y \in A \wedge y \notin \{x\}) \Leftrightarrow (y \in B)) \wedge (y \in \{x\} \Rightarrow y \in A) \big)$ {Property of sets}

$= (A \setminus \{x\} = B \wedge \{x\} \subseteq A)$ {Lemma B.3.3 and propositional calculus}

$= (A \setminus \{x\} = B \wedge x \in A)$

□

Lemma B.3.3 (set-membership-subset-1)

$$\{x\} \subseteq A \Leftrightarrow x \in A$$

Proof.

$\{x\} \subseteq A$ {Definition of subset inclusion}

$= \forall y \bullet y \in \{x\} \Rightarrow y \in A$ {Propositional calculus}

$= \forall y \bullet \neg (y \in \{x\} \wedge y \notin A)$ {Propositional calculus}

$= \neg \exists y \bullet y = x \wedge y \notin A$ {One-point rule}

$= \neg (x \notin A)$ {Propositional calculus}

$= x \in A$

□

Lemma B.3.4 (set-membership-subset-2)

$$x \notin A \Leftrightarrow (\forall y \bullet y \in A \Rightarrow y \notin \{x\})$$

Proof.

$x \notin A$ {Propositional calculus}

$= \neg (x \in A)$ {Introduce fresh variable}

$= \neg (\exists y \bullet y = x \wedge y \in A)$ {Property of sets}

$= \neg (\exists y \bullet y \in \{x\} \wedge y \in A)$ {Propositional cauclus}

$= \forall y \bullet y \in A \Rightarrow y \notin \{x\}$

□

Lemma B.3.5

$$(A = (B \cup \{x\}) \wedge x \in B) \Leftrightarrow (A = B \wedge x \in B)$$

Proof. (Implication)

$$
\begin{aligned}
& A = B \cup \{x\} \wedge x \in B && \{\text{Property of sets}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge (B \cup \{x\}) \subseteq A \wedge x \in B) && \{\text{Lemma B.3.3}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge (B \cup \{x\}) \subseteq A \wedge \{x\} \subseteq B) && \{\text{Property of sets}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge B \subseteq A \wedge \{x\} \subseteq A \wedge \{x\} \subseteq B) && \{\text{Property of sets}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge B \subseteq A \wedge \{x\} \subseteq A \wedge (\{x\} \cup B = B)) \\
& && \{\text{Propositional calculus}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge B \subseteq A \wedge \{x\} \subseteq A \wedge (\{x\} \cup B) \subseteq B) \wedge B \subseteq (\{x\} \cup B) \\
& && \{\text{Transitivity of subset inclusion and propositional calculus}\} \\
& = (A \subseteq (B \cup \{x\}) \wedge B \subseteq A \wedge \{x\} \subseteq A \wedge (\{x\} \cup B) \subseteq B \wedge A \subseteq B \wedge B \subseteq (\{x\} \cup B)) \\
& && \{\text{Propositional calculus}\} \\
& \Rightarrow B \subseteq A \wedge A \subseteq B \wedge (\{x\} \cup B) \subseteq B \wedge B \subseteq (\{x\} \cup B) && \{\text{Property of sets}\} \\
& = (B = A \wedge \{x\} \cup B) && \{\text{Lemma B.3.3}\} \\
& = (B = A \wedge x \in B)
\end{aligned}
$$

□

Proof. (Reverse implication)

$$
\begin{aligned}
& (B = A \wedge x \in B) && \{\text{Lemma B.3.3}\} \\
& (B = A \wedge \{x\} \subseteq B) && \{\text{Property of sets}\} \\
& = (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B) \\
& && \{\text{Transitivity of subset inclusion and propositional calculus}\} \\
& = (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B \wedge \{x\} \subseteq A) && \{\text{Property of sets}\} \\
& = (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B \wedge \{x\} \subseteq A \wedge (B \cup \{x\}) \subseteq A \wedge (A \cup \{x\}) \subseteq B \\
& && \{\text{Property of sets}\} \\
& = (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B \wedge (\{x\} \cup B = B) \wedge \{x\} \subseteq A \wedge (B \cup \{x\}) \subseteq A \wedge (A \cup \{x\}) \subseteq B \\
& && \{\text{Property of sets and weaken predicate}\} \\
& \Rightarrow (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B \wedge B \subseteq (\{x\} \cup B) \wedge \{x\} \subseteq A \wedge (B \cup \{x\}) \subseteq A \wedge (A \cup \{x\}) \subseteq B \\
& && \{\text{Transitivity of subset inclusion and propositional calculus}\} \\
& \Rightarrow (A \subseteq B \wedge B \subseteq A \wedge \{x\} \subseteq B \wedge B \subseteq (\{x\} \cup B) \wedge A \subseteq (\{x\} \cup B) \wedge \{x\} \subseteq A \wedge (B \cup \{x\}) \subseteq A \\
& && \{\text{Property of sets}\} \\
& = (A = B \wedge B \subseteq (\{x\} \cup B) \wedge \{x\} \subseteq B \wedge \{x\} \subseteq A \wedge (B \cup \{x\}) = A \\
& && \{\text{Propositional calculus}\}
\end{aligned}
$$

$\Rightarrow (\{x\} \subseteq B \wedge (B \cup \{x\}) = A)$ {Lemma B.3.3}
$= ((B \cup \{x\}) = A \wedge x \in B)$

$\square$

Lemma B.3.6

$$((A \cup \{x\}) \subseteq (B \cup \{x\}) \wedge x \notin A \wedge x \notin B) \Leftrightarrow (A \subseteq B \wedge x \notin A \wedge x \notin B)$$

Proof.

$(A \cup \{x\}) \subseteq (B \cup \{x\}) \wedge x \notin A \wedge x \notin B$ {Definition of subset inclusion}
$= \forall y \bullet y \in (A \cup \{x\}) \Rightarrow y \in (B \cup \{x\}) \wedge x \notin A \wedge x \notin B$
{Property of sets}
$= \forall y \bullet (y \in A \vee y \in \{x\}) \Rightarrow (y \in B \vee y \in \{x\}) \wedge x \notin A \wedge x \notin B$
{Propositional calculus}
$= \forall y \bullet y \in A \Rightarrow (y \in B \vee y \in \{x\}) \wedge x \notin A \wedge x \notin B$ {Lemma B.3.4}
$= \left(\begin{array}{l} \forall y \bullet y \in A \Rightarrow (y \in B \vee y \in \{x\}) \\ \wedge \\ \forall y \bullet y \in A \Rightarrow y \notin \{x\} \\ \wedge \\ \forall y \bullet y \in B \Rightarrow y \notin \{x\} \end{array}\right)$ {Propositional calculus}
$= \forall y \bullet \left(\begin{array}{l} y \in A \Rightarrow (y \in B \wedge y \notin \{x\}) \\ \wedge \\ y \in B \Rightarrow y \notin \{x\} \end{array}\right)$ {Propositional calculus}
$= \forall y \bullet (y \in A \Rightarrow y \in B) \wedge ((y \in A \vee y \in B) \Rightarrow (y \notin \{x\}))$
{Propositional calculus and definition of subset inclusion}
$= A \subseteq B \wedge \forall y \bullet ((y \in A \vee y \in B) \Rightarrow (y \notin \{x\}))$
{Property of sets and Lemma B.3.4}
$= A \subseteq B \wedge x \notin (A \cup B)$ {Propositional calculus and property of sets}
$= A \subseteq B \wedge x \notin A \wedge x \notin B$

$\square$

Appendix C

Predicative model

C.1 $d2bmb$

Lemma C.1.1 ($d2bmb$-A-healthy)

$$
\begin{array}{l}
d2bmb(\mathbf{A}(P)) \\
= \\
\left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land \perp \notin ss \land ss \neq \emptyset)) \land ac_0 \subseteq ss \end{array} \right) \right. \end{array} \right\}
\end{array}
$$

Proof.

$d2bmb(\mathbf{A}(P))$ {Definition of $\mathbf{A}$}

$= d2bmb(\neg\, \mathbf{PBMH1}(P^f) \vdash \mathbf{PBMH1}(P^t) \land ac' \neq \emptyset)$

{Definition of **PBMH1**}

$= d2bmb(\neg\, (P^f \,;\, ac \subseteq ac') \vdash (P^t \,;\, ac \subseteq ac') \land ac' \neq \emptyset)$

{Definition of $d2bmb$ (Definition 55)}

$$
= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((\neg\, (P^f \,;\, ac \subseteq ac') \Rightarrow ((P^t \,;\, ac \subseteq ac') \land ac' \neq \emptyset))[ss/ac'] \land \perp \notin ss) \\ \lor \\ ((P^f \,;\, ac \subseteq ac')[ss \setminus \{\perp\}/ac'] \land \perp \in ss) \end{array} \right. \end{array} \right\}
$$

{Definition of sequential composition}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} \neg\, (\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac') \\ \Rightarrow \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet P^t[ac_0/ac'] \wedge ac_0 \subseteq ac' \wedge ac' \neq \emptyset) \end{array} \right) [ss/ac'] \\ \wedge \perp \notin ss \end{array} \right) \\ \vee \\ ((\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac')[ss \setminus \{\perp\}/ac'] \wedge \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Type: $\perp \notin ac'$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} \neg \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac' \\ \wedge \perp \notin ac_0 \wedge \perp \notin ac' \end{array} \right) \\ \Rightarrow \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \wedge ac_0 \subseteq ac' \\ \wedge \perp \notin ac_0 \wedge \perp \notin ac' \wedge ac' \neq \emptyset \end{array} \right) \end{array} \right) [ss/ac'] \\ \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ac' \\ \wedge \perp \notin ac_0 \wedge \perp \notin ac' \end{array} \right) [ss \setminus \{\perp\}/ac'] \\ \wedge \perp \in ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Substitution}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \left(\begin{array}{l} \neg \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq ss \\ \wedge \perp \notin ac_0 \wedge \perp \notin ss \end{array} \right) \\ \Rightarrow \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \wedge ac_0 \subseteq ss \\ \wedge \perp \notin ac_0 \wedge \perp \notin ss \wedge ss \neq \emptyset \end{array} \right) \end{array} \right) \\ \wedge \perp \notin ss \end{array} \right) \\ \vee \\ \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \wedge ac_0 \subseteq (ss \setminus \{\perp\}) \\ \wedge \perp \notin ac_0 \wedge \perp \notin (ss \setminus \{\perp\}) \end{array} \right) \\ \wedge \perp \in ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Propositional calculus and property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \\ \land \perp \notin ac_0 \land \perp \notin ss \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss \\ \land \perp \notin ac_0 \land \perp \notin ss \land ss \neq \emptyset \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq (ss \setminus \{\perp\}) \\ \land \perp \notin ac_0 \land \perp \in ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \\ \land \perp \notin ac_0 \land \perp \notin ss \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss \\ \land \perp \notin ac_0 \land \perp \notin ss \land ss \neq \emptyset \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land \\ (\forall\, x : \mathbb{P}\, State_\perp \bullet x \in ac_0 \Rightarrow x \in ss) \land \\ (\forall\, x : \mathbb{P}\, State_\perp \bullet x \in ac_0 \Rightarrow x \notin \{\perp\}) \\ \land \perp \notin ac_0 \land \perp \in ss \end{array} \right) \end{array} \right. \end{array} \right\}$$

{Propositional calculus, property of sets and Lemma B.3.4}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ac_0 \land \perp \notin ss) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ac_0 \land \perp \notin ss \land ss \neq \emptyset) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ac_0 \land \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Propositional calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ac_0) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State_\perp \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ac_0 \land \perp \notin ss \land ss \neq \emptyset) \end{array} \right. \end{array} \right\}$$

{Propositional calculus}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \; \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State_\perp \bullet (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land \perp \notin ss \land ss \neq \emptyset)) \\ \land\, ac_0 \subseteq ss \land \perp \notin ac_0 \end{array} \right) \right. \end{array} \right\}$$

{Type restriction: $\perp \notin ac_0$}

$$= \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \; \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land \perp \notin ss \land ss \neq \emptyset)) \land ac_0 \subseteq ss \end{array} \right) \right. \end{array} \right\}$$

□

Lemma C.1.2

$$\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in d2bmb(\mathbf{A}(P)) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ = \\ \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \end{array}$$

Proof.

$\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0 \cup \{\perp\}) \in d2bmb(\mathbf{A}(P)) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)$

{Definition of $d2bmb(\mathbf{A}(P))$}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \\ \bullet\, (s, ss_0 \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \perp \notin ss)) \\ \land\, ac_0 \subseteq ss \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land (ss_0 \cup \{\perp\}) \neq \emptyset \land \perp \notin (ss_0 \cup \{\perp\}))) \\ \land\, ac_0 \subseteq (ss_0 \cup \{\perp\}) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets and predicate calculus}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet \\ P^f[ac_0/ac'] \land ac_0 \subseteq (ss_0 \cup \{\perp\}) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet \\ P^f[ac_0/ac'] \land (ac_0 \setminus \{\perp\}) \subseteq ss_0 \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Type of ac' : $\perp \notin ac'$, and property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet \\ P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \perp \in ss_0 \land \perp \in ss \\ \lor \\ \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \in ss \\ \lor \\ \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ss \end{array} \right)$$

{Predicate calculus}

$$= \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss$$

□

Lemma C.1.3

$$\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in d2bmb(\mathbf{A}(P)) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ = \\ \exists\, ac_0 : \mathbb{P}\, State \bullet (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \perp \notin ss)) \land ac_0 \subseteq ss \end{array}$$

Proof.

$\exists\, ss_0 : \mathbb{P}\, State_\perp \bullet (s, ss_0) \in d2bmb(\mathbf{A}(P)) \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss)$

{Definition of $d2bmb(\mathbf{A}(P))$}

$$= \left(\begin{array}{l} \exists\, ss_0 \bullet \\ (s, ss_0) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} \exists\, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \perp \notin ss)) \\ \land\, ac_0 \subseteq ss \end{array} \right. \end{array} \right\} \\ \land\, ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet \\ (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss_0 \neq \emptyset \land \perp \notin ss_0)) \\ \land\, ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land (\perp \in ss_0 \Leftrightarrow \perp \in ss) \\ \lor \\ \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \\ (\perp \in ss_0 \Leftrightarrow \perp \in ss) \land ss_0 \neq \emptyset \land \perp \notin ss_0 \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} (\exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \perp \in ss_0 \land \perp \in ss) \\ \lor \\ (\exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \perp \notin ss_0 \land \perp \notin ss) \\ \lor \\ \left(\begin{array}{l} \exists\, ss_0 : \mathbb{P}\, State_\perp, ac_0 : \mathbb{P}\, State \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss_0 \land ss_0 \subseteq ss \land \\ \land\, ss_0 \neq \emptyset \land \perp \notin ss_0 \land \perp \notin ss \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} (\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \in ss) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq ss \land \perp \notin ss) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet P^t[ac_0/ac'] \land ac_0 \subseteq ss \land ss \neq \emptyset \land \perp \notin ss) \end{array} \right)$$

{Predicate calculus}

$$= \left(\exists\, ac_0 : \mathbb{P}\, State \bullet (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \perp \notin ss)) \land ac_0 \subseteq ss \right)$$

□

Lemma C.1.4

$$(s, \{\perp\}) \in d2bmb(\mathbf{A}(P)) \Leftrightarrow (s, \emptyset) \in d2bmb(\mathbf{A}(P))$$

Proof.

$(s, \{\perp\}) \in d2bmb(\mathbf{A}(P)) \Leftrightarrow (s, \emptyset) \in d2bmb(\mathbf{A}(P))$

{Lemma C.1.5 and Lemma C.1.6}

$= true$

□

Lemma C.1.5

$$(s, \{\bot\}) \in d2bmb(\mathbf{A}(P)) = P^f[\emptyset/ac']$$

Proof.

$$
\begin{aligned}
&(s, \{\bot\}) \in d2bmb(\mathbf{A}(P)) && \{\text{Lemma C.1.1}\}\\
&= (s, \{\bot\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid \exists\, ac_0 : \mathbb{P}\, State \bullet \\ \quad (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \bot \notin ss)) \\ \quad \land\, ac_0 \subseteq ss \end{array} \right\} \\
& && \{\text{Property of sets}\}\\
&= \exists\, ac_0 : \mathbb{P}\, State \bullet (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land \{\bot\} \neq \emptyset \land \bot \notin \{\bot\})) \land ac_0 \subseteq \{\bot\} \\
& && \{\text{Property of sets and predicate calculus}\}\\
&= \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq \{\bot\} \\
& && \{\text{Case-analysis on } ac_0 \text{ and one-point rule}\}\\
&= P^f[\emptyset/ac']
\end{aligned}
$$

□

Lemma C.1.6

$$(s, \emptyset) \in d2bmb(\mathbf{A}(P)) = P^f[\emptyset/ac']$$

Proof.

$$
\begin{aligned}
&(s, \emptyset) \in d2bmb(\mathbf{A}(P)) && \{\text{Definition of } d2bmb \text{ for } P \text{ that is } \mathbf{A}\text{-healthy}\}\\
&= (s, \emptyset) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\bot \\ \mid \exists\, ac_0 : \mathbb{P}\, State \bullet \\ \quad (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land ss \neq \emptyset \land \bot \notin ss)) \\ \quad \land\, ac_0 \subseteq ss \end{array} \right\} \\
& && \{\text{Property of sets}\}\\
&= \exists\, ac_0 : \mathbb{P}\, State \bullet (P^f[ac_0/ac'] \lor (P^t[ac_0/ac'] \land \emptyset \neq \emptyset \land \bot \notin \emptyset)) \land ac_0 \subseteq \emptyset \\
& && \{\text{Property of sets and predicate calculus}\}\\
&= \exists\, ac_0 : \mathbb{P}\, State \bullet P^f[ac_0/ac'] \land ac_0 \subseteq \emptyset \\
& && \{\text{Property of sets and one-point rule}\}\\
&= P^f[\emptyset/ac']
\end{aligned}
$$

□

Lemma C.1.7

$$(s, \emptyset) \in d2bmb(\mathbf{A}(P)) \Leftrightarrow (s, \{\bot\}) \in d2bmb(\mathbf{A}(P)) = true$$

Proof.

$$
\begin{aligned}
&(s, \emptyset) \in d2bmb(\mathbf{A}(P)) \Leftrightarrow (s, \{\bot\}) \in d2bmb(\mathbf{A}(P)) \\
&\hfill \{\text{Lemma C.1.6 and Lemma C.1.5}\} \\
&= true
\end{aligned}
$$

□

C.2 $bmb2d$

Lemma C.2.1

$$
\begin{aligned}
&((s, ac') \in B \mathbin{;} ac \subseteq ac') \land (s, \emptyset) \notin B \\
&\Leftrightarrow \\
&((s, ac') \in B \mathbin{;} ac \subseteq ac') \land ac' \neq \emptyset \land (s, \emptyset) \notin B
\end{aligned}
$$

Proof.

$$
\begin{aligned}
&((s, ac') \in B \mathbin{;} ac \subseteq ac') \land (s, \emptyset) \notin B \\
&\hfill \{\text{Definition of sequential composition}\} \\
&\Leftrightarrow (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac') \land (s, \emptyset) \notin B \\
&\hfill \{\text{Predicate calculus}\} \\
&\Leftrightarrow \left(\begin{array}{l} (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac' \land (ac' = \emptyset \lor ac' \neq \emptyset)) \\ \land \\ (s, \emptyset) \notin B \end{array} \right) \\
&\hfill \{\text{Predicate calculus}\} \\
&\Leftrightarrow \left(\begin{array}{l} (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac' \land ac' = \emptyset) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac' \land ac' \neq \emptyset) \end{array} \right) \land (s, \emptyset) \notin B \\
&\hfill \{\text{Property of sets and case analysis on } ac_0\} \\
&\Leftrightarrow \left(\begin{array}{l} ((s, \emptyset) \in B \land ac' = \emptyset) \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac' \land ac' \neq \emptyset) \end{array} \right) \land (s, \emptyset) \notin B \\
&\hfill \{\text{Predicate calculus}\}
\end{aligned}
$$

$$\Leftrightarrow (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0) \in B \land ac_0 \subseteq ac' \land ac' \neq \emptyset) \land (s, \emptyset) \notin B$$

{Definition of sequential composition}

$$\Leftrightarrow ((s, ac') \in B \mathbin{;} ac \subseteq ac' \land ac' \neq \emptyset) \land (s, \emptyset) \notin B$$

□

Lemma C.2.2 *Provided B satisfies* $\mathbf{bmh_{0,1,2}}$.

$$bmb2d(B) = \left(\begin{array}{l} \neg\, ((s, ac' \cup \{\bot\}) \in B \mathbin{;} ac \subseteq ac') \\ \vdash \\ ((s, ac') \in B \mathbin{;} ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

Proof.

$bmb2d(B)$ {Assumption: B satisfies $\mathbf{bmh_{0,1,2}}$}

$= bmb2d(\mathbf{bmh_{0,1,2}}(B))$ {Lemma C.2.3}

$$= \left(\begin{array}{l} \left(\begin{array}{l} \neg\, ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \land \\ \neg \left(\begin{array}{l} ((s, ac' \cup \{\bot\}) \in B \mathbin{;} ac \subseteq ac') \\ \land \\ (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right) \end{array} \right) \\ \vdash \\ ((s, ac') \in B \mathbin{;} ac \subseteq ac') \land (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \neg \left(\begin{array}{l} \left(\begin{array}{l} ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ ((s, ac' \cup \{\bot\}) \in B \mathbin{;} ac \subseteq ac') \end{array} \right) \\ \land \\ \left(\begin{array}{l} ((s, \{\bot\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ ((s, \{\bot\}) \notin B \land (s, \emptyset) \notin B) \end{array} \right) \end{array} \right) \\ \vdash \\ ((s, ac') \in B \mathbin{;} ac \subseteq ac') \land (s, \{\bot\}) \notin B \land (s, \emptyset) \notin B \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \neg \left(\begin{array}{l} \left(\begin{array}{l} ((s, \{\perp\}) \in B \land (s, \emptyset) \in B) \\ \lor \\ ((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ac') \end{array} \right) \\ \land \\ ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \end{array} \right) \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \{\perp\}) \notin B \land (s, \emptyset) \notin B \end{array} \right)$$

{B is **BMH2**-healthy, as B satisfies $\mathbf{bmh_{0,1,2}}$ and Theorem 4.3.1}

$$= \left(\begin{array}{l} \neg \left(\begin{array}{l} (s, \{\perp\}) \in B \\ \lor \\ ((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ac') \end{array} \right) \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

{Definition of sequential composition}

$$= \left(\begin{array}{l} \neg \left(\begin{array}{l} (s, \{\perp\}) \in B \\ \lor \\ (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0 \cup \{\perp\}) \in B \land ac_0 \subseteq ac') \end{array} \right) \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

{Instantiation of existential quantifier for $ac_0 = \emptyset$}

$$= \left(\begin{array}{l} \neg\, (\exists\, ac_0 : \mathbb{P}\, State \bullet (s, ac_0 \cup \{\perp\}) \in B \land ac_0 \subseteq ac') \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

{Definition of sequential composition}

$$= \left(\begin{array}{l} \neg\, ((s, ac' \cup \{\perp\}) \in B \ ; \ ac \subseteq ac') \\ \vdash \\ ((s, ac') \in B \ ; \ ac \subseteq ac') \land (s, \emptyset) \notin B \end{array} \right)$$

□

Lemma C.2.3

$bmb2d(\mathbf{bmh_{0,1,2}}(B))$

$=$

$$\left(\begin{array}{l}\left(\begin{array}{l}\neg\,((s,\{\bot\})\in B\wedge(s,\emptyset)\in B)\\ \wedge\\ \neg\,(((s,ac'\cup\{\bot\})\in B\ ;\ ac\subseteq ac')\wedge(s,\{\bot\})\notin B\wedge(s,\emptyset)\notin B)\end{array}\right)\\ \vdash\\ ((s,ac')\in B\ ;\ ac\subseteq ac')\wedge(s,\{\bot\})\notin B\wedge(s,\emptyset)\notin B\end{array}\right)$$

Proof.

$$\begin{array}{l}bmb2d(\mathbf{bmh_{0,1,2}}(B)) \hfill \{\text{Definitifon of } bmb2d\}\\
= ok\Rightarrow\left(\begin{array}{l}((s,ac')\in\mathbf{bmh_{0,1,2}}(B)\wedge\bot\notin ac'\wedge ok')\\ \vee\\ ((s,ac'\cup\{\bot\})\in\mathbf{bmh_{0,1,2}}(B)\wedge\bot\notin ac')\end{array}\right)\\
\hfill \{\text{Definition of } \mathbf{bmh_{0,1,2}}(B)\}\\
= ok\Rightarrow\left(\begin{array}{l}\left(\begin{array}{l}(s,ac')\in\left\{\begin{array}{l}s:State,ss:\mathbb{P}\,State_\bot\\ \mid\ \exists\,ss_0\bullet((s,ss_0)\in B\vee(s,ss_0\cup\{\bot\})\in B)\\ \mid\ \wedge((s,\{\bot\})\in B\Leftrightarrow(s,\emptyset)\in B)\\ \mid\ \wedge\,ss_0\subseteq ss\wedge(\bot\in ss_0\Leftrightarrow\bot\in ss)\end{array}\right\}\\ \wedge\\ \bot\notin ac'\wedge ok'\end{array}\right)\\ \vee\\ \left(\begin{array}{l}(s,ac'\cup\{\bot\})\in\left\{\begin{array}{l}s:State,ss:\mathbb{P}\,State_\bot\\ \mid\ \exists\,ss_0\bullet((s,ss_0)\in B\vee(s,ss_0\cup\{\bot\})\in B)\\ \mid\ \wedge((s,\{\bot\})\in B\Leftrightarrow(s,\emptyset)\in B)\\ \mid\ \wedge\,ss_0\subseteq ss\wedge(\bot\in ss_0\Leftrightarrow\bot\in ss)\end{array}\right\}\\ \wedge\bot\notin ac'\end{array}\right)\end{array}\right)\\
\hfill \{\text{Property of sets and predicate calculus}\}\\
= ok\Rightarrow\left(\begin{array}{l}\left(\begin{array}{l}\exists\,ss_0\bullet((s,ss_0)\in B\vee(s,ss_0\cup\{\bot\})\in B)\\ \wedge((s,\{\bot\})\in B\Leftrightarrow(s,\emptyset)\in B)\\ \wedge\,ss_0\subseteq ac'\wedge(\bot\in ss_0\Leftrightarrow\bot\in ac')\\ \wedge\bot\notin ac'\wedge ok'\end{array}\right)\\ \vee\\ \left(\begin{array}{l}\exists\,ss_0\bullet((s,ss_0)\in B\vee(s,ss_0\cup\{\bot\})\in B)\\ \wedge((s,\{\bot\})\in B\Leftrightarrow(s,\emptyset)\in B)\\ \wedge\,ss_0\subseteq(ac'\cup\{\bot\})\wedge(\bot\in ss_0\Leftrightarrow\bot\in(ac'\cup\{\bot\}))\\ \wedge\bot\notin ac'\end{array}\right)\end{array}\right)\\
\hfill \{\text{Predicate calculus}\}\end{array}$$

$$= ok \Rightarrow \left(\begin{array}{l} \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\bot\}) \in B) \\ \land\ ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\ ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \end{array} \right) \\ \lor \\ \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ss_0) \in B \lor (s, ss_0 \cup \{\bot\}) \in B) \\ \land\ ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land\ ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)$$

{Property of sets}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists\, ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)$$

{Predicate calculus: introduce fresh variable}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists t, ss_0 \bullet (s, t) \in B \land t = ss_0 \cup \{\bot\} \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Lemma B.3.2}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists t, ss_0 \bullet (s, t) \in B \land t \setminus \{\bot\} = ss_0 \land ss_0 \subseteq ac' \land \bot \in t \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{One-point rule and substitution}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists t \bullet (s, t) \in B \land (t \setminus \{\bot\}) \subseteq ac' \land \bot \in t \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Property of sets and variable renaming}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\bot\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \land \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq ac' \land \bot \notin ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \land ok' \right) \\ \lor \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \land ss_0 \subseteq (ac' \cup \{\bot\}) \land \bot \in ss_0 \land \bot \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Predicate calculus: absorption law}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \wedge ok' \right) \\ \vee \\ \left(\exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq (ac' \cup \{\perp\}) \wedge \perp \in ss_0 \wedge \perp \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Lemma B.3.1}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \wedge ok' \right) \\ \vee \\ \left(\exists ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge (\perp \in ss_0 \Leftrightarrow \perp \in ac') \wedge \perp \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Predicate calculus}

$$
= ok \Rightarrow \left(\begin{array}{l} ((s, \{\perp\}) \in B \Leftrightarrow (s, \emptyset) \in B) \\ \wedge \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \wedge ok' \right) \\ \vee \\ \left(\exists ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Predicate calculus}

$$
= ok \Rightarrow \left(\begin{array}{l} (((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \vee ((s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B)) \\ \wedge \\ \left(\begin{array}{l} \left(\exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \wedge ok' \right) \\ \vee \\ \left(\exists ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \wedge \perp \notin ac' \right) \end{array} \right) \end{array} \right)
$$

{Instantiation: consider case where $ss_0 = \emptyset$}

$$
= ok \Rightarrow \left(\begin{array}{l} (((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \vee ((s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B)) \\ \wedge \\ \left(\begin{array}{l} \left(\begin{array}{l} (s, \emptyset) \\ \vee \\ \exists ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \end{array} \right) \wedge \perp \notin ac' \wedge ok' \\ \vee \\ \left(\begin{array}{l} (s, \{\perp\}) \\ \vee \\ \exists ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \end{array} \right) \wedge \perp \notin ac' \end{array} \right) \end{array} \right)
$$

{Predicate calculus: distribution}

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \emptyset) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge ok' \wedge (s, \{\bot\}) \in B \wedge (s, \emptyset) \in B
\end{array}\right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \emptyset) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge ok' \wedge (s, \{\bot\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \{\bot\}) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge (s, \{\bot\}) \in B \wedge (s, \emptyset) \in B
\end{array}\right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \{\bot\}) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge (s, \{\bot\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right)
$$

{Predicate calculus: absorption law}

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\bot \notin ac' \wedge ok' \wedge (s, \{\bot\}) \in B \wedge (s, \emptyset) \in B \right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \emptyset) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge ok' \wedge (s, \{\bot\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right) \\
\vee \\
\left(\bot \notin ac' \wedge (s, \{\bot\}) \in B \wedge (s, \emptyset) \in B \right) \\
\vee \\
\left(\begin{array}{l}
\left(\begin{array}{l}
(s, \{\bot\}) \\
\vee \\
\exists\, ss_0 \bullet (s, ss_0 \cup \{\bot\}) \in B \wedge ss_0 \subseteq ac' \wedge \bot \notin ss_0
\end{array}\right) \\
\wedge \bot \notin ac' \wedge (s, \{\bot\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right)
$$

{Instantiation: consider case where $ss_0 = \emptyset$}

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\perp \notin ac' \wedge ok' \wedge (s, \{\perp\}) \in B \wedge (s, \emptyset) \in B \right) \\
\vee \\
\left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \wedge ok' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right) \\
\vee \\
\left(\perp \notin ac' \wedge (s, \{\perp\}) \in B \wedge (s, \emptyset) \in B \right) \\
\vee \\
\left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right)
$$

{Predicate calculus: absorption law}

$$
= ok \Rightarrow \left(\begin{array}{l}
\left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \wedge ok' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right) \\
\vee \\
\left(\perp \notin ac' \wedge (s, \{\perp\}) \in B \wedge (s, \emptyset) \in B \right) \\
\vee \\
\left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right)
$$

{Predicate calculus}

$$
= \left(\begin{array}{l}
\left(\begin{array}{l}
ok \\
\wedge \\
\neg\, (\perp \notin ac' \wedge (s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\
\wedge \\
\neg \left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0 \cup \{\perp\}) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right) \\
\Rightarrow \\
\left(\begin{array}{l}
\exists\, ss_0 \bullet (s, ss_0) \in B \wedge ss_0 \subseteq ac' \wedge \perp \notin ss_0 \\
\wedge \perp \notin ac' \wedge ok' \\
\wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B
\end{array}\right)
\end{array}\right)
$$

{Variable renaming and substitution}

$$= \left(\begin{array}{l} \left(\begin{array}{l} ok \\ \wedge \\ \neg\, (\perp \notin ac' \wedge (s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \wedge \\ \neg \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ac' \cup \{\perp\}) \in B \wedge \perp \notin ac')[ss_0/ac'] \wedge (ac \subseteq ac')[ss_0/ac] \\ \wedge \perp \notin ac' \\ \wedge\, (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \end{array} \right) \end{array} \right) \\ \Rightarrow \\ \left(\begin{array}{l} \exists\, ss_0 \bullet ((s, ac') \in B \wedge \perp \notin ac')[ss_0/ac'] \wedge (ac \subseteq ac')[ss_0/ac] \\ \wedge \perp \notin ac' \wedge ok' \\ \wedge\, (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \end{array} \right) \end{array} \right)$$

$\{$Definition of sequential composition and type of $ac' : \perp \notin ac'\}$

$$= \left(\begin{array}{l} \left(\begin{array}{l} ok \\ \wedge \\ \neg\, ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \wedge \\ \neg \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \in B \,;\, ac \subseteq ac') \\ \wedge\, (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \end{array} \right) \end{array} \right) \\ \Rightarrow \\ \left(\begin{array}{l} ((s, ac') \in B \,;\, ac \subseteq ac') \wedge ok' \\ \wedge\, (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \end{array} \right) \end{array} \right)$$

$\{$Definition of design$\}$

$$= \left(\begin{array}{l} \left(\begin{array}{l} \neg\, ((s, \{\perp\}) \in B \wedge (s, \emptyset) \in B) \\ \wedge \\ \neg\, (((s, ac' \cup \{\perp\}) \in B \,;\, ac \subseteq ac') \wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B) \end{array} \right) \\ \vdash \\ ((s, ac') \in B \,;\, ac \subseteq ac') \wedge (s, \{\perp\}) \notin B \wedge (s, \emptyset) \notin B \end{array} \right)$$

□

Lemma C.2.4

$$(s, \{s_1 : State_\perp \mid true\}) \in d2bmb(P) = P^f[\{s_1 : State \mid true\}/ac']$$

Proof.

$(s, \{s_1 : State_\perp \mid true\}) \in d2bmb(P)$ $\{$Definition of $d2bmb\}$

$$= \left((s, \{s_1 : State_\perp \mid true\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} ((\neg\, P^f \Rightarrow P^t)[ss/ac'] \land \perp \notin ss) \\ \lor \\ (P^f[ss \setminus \{\perp\}/ac'] \land \perp \in ss) \end{array} \right. \end{array} \right\} \right)$$

$$\{\text{Property of sets}\}$$

$$= \left(\begin{array}{l} \left(\begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'][\{s_1 : State_\perp \mid true\}/ss] \\ \land \perp \notin \{s_1 : State_\perp \mid true\} \end{array} \right) \\ \lor \\ \left(\begin{array}{l} P^f[ss \setminus \{\perp\}/ac'][\{s_1 : State_\perp \mid true\}/ss] \\ \land \perp \in \{s_1 : State_\perp \mid true\} \end{array} \right) \end{array} \right)$$

$$\{\text{Property of sets and propositional calculus}\}$$

$$= P^f[ss \setminus \{\perp\}/ac'][\{s_1 : State_\perp \mid true\}/ss] \qquad \{\text{Substitution}\}$$

$$= P^f[\{s_1 : State_\perp \mid true\} \setminus \{\perp\}/ac'] \qquad \{\text{Property of sets}\}$$

$$= P^f[\{s_1 : State \mid true\}/ac']$$

□

Lemma C.2.5 *Provided* $\perp \notin ac'$.

$$\{s : State \mid (s, ac' \cup \{\perp\}) \in d2bmb(P)\} = \{s : State \mid P^f\}$$

Proof.

$$\{s : State \mid (s, ac' \cup \{\perp\}) \in d2bmb(P)\} \qquad \{\text{Definition of } d2bmb\}$$

$$= \left\{ s : State \;\middle|\; (s, ac' \cup \{\perp\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'] \land \perp \notin ss) \\ \lor \\ (P^f[ss \setminus \{\perp\}/ac'] \land \perp \in ss) \end{array} \right. \end{array} \right\} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ s : State \;\middle|\; \begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'][ac' \cup \{\perp\}/ss] \land \perp \notin (ac' \cup \{\perp\})) \\ \lor \\ (P^f[ss \setminus \{\perp\}/ac'][ac' \cup \{\perp\}/ss] \land \perp \in (ac' \cup \{\perp\})) \end{array} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ s : State \;\middle|\; (P^f[ss \setminus \{\perp\}/ac'][ac' \cup \{\perp\}/ss]) \right\} \qquad \{\text{Substitution}\}$$

$$= \left\{ s : State \;\middle|\; (P^f[ac' \cup \{\perp\} \setminus \{\perp\}/ac']) \right\}$$

$$\{\text{Property of sets, and assumption that } \perp \notin ac'\}$$

$$= \{s : State \mid P^f\}$$

$\square$

Lemma C.2.6 *Provided* $\perp \notin ac'$.

$$\{s : State \mid (s, ac') \in d2bmb(P)\} = \{s : State \mid (\neg\, P^f \Rightarrow P^t)\}$$

Proof.

$$\{s : State \mid (s, ac') \in d2bmb(P)\} \qquad \{\text{Definition of } d2bmb\}$$

$$= \left\{ s : State \;\middle|\; (s, ac') \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \quad\left|\; \begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'] \wedge \perp \notin ss) \\ \vee \\ (P^f[ss \setminus \{\perp\}/ac'] \wedge \perp \in ss) \end{array}\right. \end{array} \right\} \right\}$$

$$\{\text{Property of sets}\}$$

$$= \left\{ s : State \;\middle|\; \begin{array}{l} (\neg\, P^f \Rightarrow P^t)[ss/ac'][ac'/ss] \wedge \perp \notin ac') \\ \vee \\ (P^f[ss \setminus \{\perp\}/ac'][ac'/ss] \wedge \perp \in ac') \end{array} \right\}$$

$$\{\text{Subsitutiton}\}$$

$$= \left\{ s : State \;\middle|\; \begin{array}{l} (\neg\, P^f \Rightarrow P^t) \wedge \perp \notin ac') \\ \vee \\ (P^f[ac' \setminus \{\perp\}/ac'] \wedge \perp \in ac') \end{array} \right\}$$

$$\{\text{Assumption: } \perp \notin ac'\}$$

$$= \{s : State \mid (\neg\, P^f \Rightarrow P^t)\}$$

$\square$

Lemma C.2.7

$$(s, \{s : State \mid (s, ac' \cup \{\perp\}) \in d2bmb(P)\}) \in d2bmb(Q)$$
$$=$$
$$(\neg\, Q^f \Rightarrow Q^t)[\{s : State \mid P^f\}/ac']$$

Proof.

$$(s, \{s : State \mid (s, ac' \cup \{\perp\}) \in d2bmb(P)\}) \in d2bmb(Q) \qquad \{\text{Lemma C.2.5}\}$$
$$= (s, \{s : State \mid P^f\}) \in d2bmb(Q) \qquad \{\text{Definition of } d2bmb\}$$

$$= (s, \{s : State \mid P^f\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\neg\, Q^f \Rightarrow Q^t)[ss/ac'] \wedge \perp \notin ss) \\ \vee \\ (Q^f[ss \setminus \{\perp\}/ac'] \wedge \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} ((\neg\, Q^f \Rightarrow Q^t)[ss/ac'][\{s : State \mid P^f\}/ss] \wedge \perp \notin \{s : State \mid P^f\}) \\ \vee \\ (Q^f[ss \setminus \{\perp\}/ac'][\{s : State \mid P^f\}/ss] \wedge \perp \in \{s : State \mid P^f\}) \end{array} \right)$$

{Property of sets}

$$= (\neg\, Q^f \Rightarrow Q^t)[ss/ac'][\{s : State \mid P^f\}/ss] \qquad \{\text{Substitution}\}$$

$$= (\neg\, Q^f \Rightarrow Q^t)[\{s : State \mid P^f\}/ac']$$

□

Lemma C.2.8

$$(s, \{s : State \mid (s, ac') \in d2bmb(P)\}) \in d2bmb(Q)$$
$$=$$
$$(\neg\, Q^f \Rightarrow Q^t)[\{s : State \mid (\neg\, P^f \Rightarrow P^t)\}/ac']$$

Proof.

$$(s, \{s : State \mid (s, ac') \in d2bmb(P)\}) \in d2bmb(Q) \qquad \{\text{Lemma C.2.6}\}$$

$$= (s, \{s : State \mid (\neg\, P^f \Rightarrow P^t)\}) \in d2bmb(Q) \qquad \{\text{Definition of } d2bmb\}$$

$$= (s, \{s : State \mid (\neg\, P^f \Rightarrow P^t)\}) \in \left\{ \begin{array}{l} s : State, ss : \mathbb{P}\, State_\perp \\ \left| \begin{array}{l} (\neg\, Q^f \Rightarrow Q^t)[ss/ac'] \wedge \perp \notin ss) \\ \vee \\ (Q^f[ss \setminus \{\perp\}/ac'] \wedge \perp \in ss) \end{array} \right. \end{array} \right\}$$

{Property of sets}

$$= \left(\begin{array}{l} \left(\begin{array}{l} (\neg\, Q^f \Rightarrow Q^t)[ss/ac'][\{s : State \mid (\neg\, P^f \Rightarrow P^t)\}/ss] \\ \wedge \perp \notin \{s : State \mid (\neg\, P^f \Rightarrow P^t)\} \end{array} \right) \\ \vee \\ \left(\begin{array}{l} Q^f[ss \setminus \{\perp\}/ac'][\{s : State \mid (\neg\, P^f \Rightarrow P^t)\}/ss] \\ \wedge \perp \in \{s : State \mid (\neg\, P^f \Rightarrow P^t)\} \end{array} \right) \end{array} \right)$$

{Property of sets and propositional calculus}

$$= (\neg\, Q^f \Rightarrow Q^t)[ss/ac'][\{s : State \mid (\neg\, P^f \Rightarrow P^t)\}/ss] \qquad \{\text{Substitution}\}$$

$$= (\neg\, Q^f \Rightarrow Q^t)[\{s : State \mid (\neg\, P^f \Rightarrow P^t)\}/ac']$$

$\square$

Lemma C.2.9

$$bmb2d(B_0 \ ; \ B_1) \\ = \\ ok \Rightarrow \left(\begin{array}{l} ((s, \{s_1 : State \mid (s_1, ac') \in B_1\}) \in B_0 \wedge \bot \notin ac' \wedge ok') \\ \vee \\ ((s, \{s_1 : State_\bot \mid true\}) \in B_0 \wedge \bot \notin ac') \\ \vee \\ ((s, \{s_1 : State \mid (s_1, ac' \cup \{\bot\}) \in B_1\}) \in B_0 \wedge \bot \notin ac') \end{array} \right)$$

Proof.

$bmb2d(B_0 \ ; \ B_1)$ {Definition of $bmb2d$}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, ac') \in (B_0 \ ; \ B_1) \wedge \bot \notin ac' \wedge ok') \\ \vee \\ ((s, ac' \cup \{\bot\}) \in (B_0 \ ; \ B_1) \wedge \bot \notin ac') \end{array} \right)$$

{Definition of sequential composition}

$$= ok \Rightarrow \left(\begin{array}{l} \left(\begin{array}{l} (s, ac') \in \left(\begin{array}{l} \{s : State, ss : \mathbb{P}\, State_\bot \mid (s, \{s_1 : State_\bot \mid true\}) \in B_0\} \\ \cup \\ \{s : State, ss : \mathbb{P}\, State_\bot \mid (s, \{s_1 : State \mid (s_1, ss) \in B_1\}) \in B_0\} \end{array} \right) \\ \wedge \bot \notin ac' \wedge ok' \end{array} \right) \\ \vee \\ \left(\begin{array}{l} (s, ac' \cup \{\bot\}) \in \left(\begin{array}{l} \{s : State, ss : \mathbb{P}\, State_\bot \mid (s, \{s_1 : State_\bot \mid true\}) \in B_0\} \\ \cup \\ \{s : State, ss : \mathbb{P}\, State_\bot \mid (s, \{s_1 : State \mid (s_1, ss) \in B_1\}) \in B_0\} \end{array} \right) \\ \wedge \bot \notin ac' \end{array} \right) \end{array} \right)$$

{Property of sets and propositional calculus}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, \{s_1 : State_\bot \mid true\}) \in B_0 \wedge \bot \notin ac' \wedge ok') \\ \vee \\ ((s, \{s_1 : State \mid (s_1, ac') \in B_1\}) \in B_0 \wedge \bot \notin ac' \wedge ok') \\ \vee \\ ((s, \{s_1 : State_\bot \mid true\}) \in B_0 \wedge \bot \notin ac') \\ \vee \\ ((s, \{s_1 : State \mid (s_1, ac' \cup \{\bot\}) \in B_1\}) \in B_0 \wedge \bot \notin ac') \end{array} \right)$$

{Propositional calculus: absorption law}

$$= ok \Rightarrow \left(\begin{array}{l} ((s, \{s_1 : State \mid (s_1, ac') \in B_1\}) \in B_0 \land \bot \notin ac' \land ok') \\ \lor \\ ((s, \{s_1 : State_\bot \mid true\}) \in B_0 \land \bot \notin ac') \\ \lor \\ ((s, \{s_1 : State \mid (s_1, ac' \cup \{\bot\}) \in B_1\}) \in B_0 \land \bot \notin ac') \end{array} \right)$$

□

C.3 Other lemmas

Lemma C.3.1

$$[(\exists\, ac' \bullet P^f) = P^f] \Leftrightarrow [(\exists\, ac' \bullet \neg\, P^f) = \neg\, P^f]$$

Proof.

$[(\exists\, ac' \bullet \neg\, P^f) = \neg\, P^f]$ {Universal quantification}

$\Leftrightarrow \left(\begin{array}{l} (\forall\, ok, ok', ac', s \bullet (\exists\, ac' \bullet \neg\, P^f) \Rightarrow \neg\, P^f) \\ \land \\ (\forall\, ok, ok', ac', s \bullet \neg\, P^f \Rightarrow (\exists\, ac' \bullet \neg\, P^f)) \end{array} \right)$ {Predicate calculus}

$\Leftrightarrow \forall\, ok, ok', ac', s \bullet (\exists\, ac' \bullet \neg\, P^f) \Rightarrow \neg\, P^f$ {Predicate calculus}

$\Leftrightarrow \forall\, ok, s \bullet (\exists\, ac' \bullet \neg\, P^f) \Rightarrow (\forall\, ac' \bullet \neg\, P^f)$ {Predicate calculus}

$\Leftrightarrow \forall\, ok, s \bullet \neg\, (\forall\, ac' \bullet \neg\, P^f) \Rightarrow \neg\, (\exists\, ac' \bullet \neg\, P^f)$ {Predicate calculus}

$\Leftrightarrow \forall\, ok, s \bullet (\exists\, ac' \bullet P^f) \Rightarrow (\forall\, ac' \bullet P^f)$ {Predicate calculus}

$\Leftrightarrow \forall\, ok, s, ac', ok' \bullet (\exists\, ac' \bullet P^f) \Rightarrow P^f$ {Predicate calculus}

$\Leftrightarrow \left(\begin{array}{l} \Leftrightarrow \forall\, ok, s, ac', ok' \bullet (\exists\, ac' \bullet P^f) \Rightarrow P^f \\ \land \\ \Leftrightarrow \forall\, ok, s, ac', ok' \bullet P^f \Rightarrow (\exists\, ac' \bullet P^f) \end{array} \right)$

{Universal quantification}

$= [(\exists\, ac' \bullet P^f) = P^f]$

□

Lemma C.3.2 *Provided B_0 and B_1 are of type $BM_\bot$.*

$$\left[\begin{array}{l} (s, ac') \in B_1 \Rightarrow (s, ac') \in B_0 \\ \land \\ (s, ac' \cup \{\bot\}) \in B_1 \Rightarrow (s, ac' \cup \{\bot\}) \in B_0 \end{array} \right] \Leftrightarrow B_1 \subseteq B_0$$

Proof.

$$
\begin{array}{l}
B_1 \subseteq B_0 \hfill \{\text{Definition of subset inclusion}\} \\
\Leftrightarrow \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet (s, ss) \in B_1 \Rightarrow (s, ss) \in B_0 \\
\hfill \{\text{Predicate calculus}\} \\
\Leftrightarrow \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \wedge (\perp \in ss \vee \perp \notin ss) \end{array} \right) \\
\hfill \{\text{Predicate calculus}\} \\
\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \perp \in ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \\ \wedge \\ \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \end{array} \right) \\
\hfill \{\text{Predicate calculus}\} \\
\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ (\exists\, t : State, ss : \mathbb{P}\, State \bullet t = ss \setminus \{\perp\} \wedge \perp \in ss) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \\ \wedge \\ \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \end{array} \right) \\
\hfill \{\text{Lemma B.3.2}\} \\
\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ (\exists\, t : State, ss : \mathbb{P}\, State \bullet \perp \notin t \wedge t \cup \{\perp\} = ss) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \\ \wedge \\ \left(\begin{array}{l} \forall\, s : State, ss : \mathbb{P}\, State_\perp \bullet \\ \perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array} \right) \end{array} \right) \\
\hfill \{\text{Type: } \perp \notin t\}
\end{array}
$$

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_{\perp} \bullet \\ (\exists t : State, ss : \mathbb{P}\, State \bullet t \cup \{\perp\} = ss) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \\ \wedge \\ \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_{\perp} \bullet \\ \perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \end{array}\right)$$

{Predicate calculus}

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_{\perp}, t : \mathbb{P}\, State \bullet \\ (t \cup \{\perp\} = ss) \\ \Rightarrow \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \\ \wedge \\ \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State_{\perp} \bullet \\ \perp \notin ss \Rightarrow ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \end{array}\right)$$

{Predicate calculus: one-point rule}

$$\Leftrightarrow \left(\begin{array}{l} \left(\begin{array}{l} \forall s : State, t : \mathbb{P}\, State \bullet \\ ((s, t \cup \{\perp\}) \in B_1 \Rightarrow (s, t \cup \{\perp\}) \in B_0) \end{array}\right) \\ \wedge \\ \left(\begin{array}{l} \forall s : State, ss : \mathbb{P}\, State \bullet \\ ((s, ss) \in B_1 \Rightarrow (s, ss) \in B_0) \end{array}\right) \end{array}\right)$$

{Variable renaming and predicate calculus}

$$\Leftrightarrow \forall s : State, ac' : \mathbb{P}\, State \bullet \left(\begin{array}{l} ((s, ac' \cup \{\perp\}) \in B_1 \Rightarrow (s, ac' \cup \{\perp\}) \in B_0) \\ \wedge \\ ((s, ac') \in B_1 \Rightarrow (s, ac') \in B_0) \end{array}\right)$$

{Universal quantification}

$$\Leftrightarrow \left[\begin{array}{l} ((s, ac' \cup \{\perp\}) \in B_1 \Rightarrow (s, ac' \cup \{\perp\}) \in B_0) \\ \wedge \\ ((s, ac') \in B_1 \Rightarrow (s, ac') \in B_0) \end{array}\right]$$

$\square$

Appendix D

PBMH

D.1 Properties

Law D.1.1 (PBMH-idempotent)

$$\mathbf{PBMH} \circ \mathbf{PBMH}(P) = \mathbf{PBMH}(P)$$

Proof.

$$\begin{array}{lr}
\mathbf{PBMH} \circ \mathbf{PBMH}(P) & \{\text{Definition of } \mathbf{PBMH}\} \\
= \mathbf{PBMH}(P \ ; \ ac \subseteq ac') & \{\text{Definition of } \mathbf{PBMH}\} \\
= ((P \ ; \ ac \subseteq ac') \ ; \ ac \subseteq ac') & \{\text{Associativity of sequential composition}\} \\
= (P \ ; \ (ac \subseteq ac' \ ; \ ac \subseteq ac')) & \{\text{Definition of sequential composition}\} \\
= (P \ ; \ (\exists\, ac_0 \bullet ac \subseteq ac_0 \land ac_0 \subseteq ac')) & \{\text{Transitivity of subset inclusion}\} \\
= (P \ ; \ ac \subseteq ac') & \{\text{Definition of } \mathbf{PBMH}\} \\
= \mathbf{PBMH}(P) &
\end{array}$$

□

Lemma D.1.1

$$\mathbf{PBMH}(P) = \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac'$$

Proof.

$$\begin{array}{lr}
\mathbf{PBMH}(P) & \{\text{Definition of } \mathbf{PBMH}\} \\
= P \ ; \ ac \subseteq ac' & \{\text{Definition of sequential composition}\} \\
= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac' &
\end{array}$$

$\square$

D.2 Distribution properties

Law D.2.1 (PBMH-distribute-disjunction)

$$\mathbf{PBMH}(P \lor Q) = \mathbf{PBMH}(P) \lor \mathbf{PBMH}(Q)$$

Proof.

$$\begin{aligned}
&\mathbf{PBMH}(P \lor Q) && \{\text{Definition of } \mathbf{PBMH}\}\\
&= (P \lor Q) \ ;\ ac \subseteq ac' && \{\text{Definition of sequential composition}\}\\
&= \exists\, ac_0 \bullet (P[ac_0/ac'] \lor Q[ac_0/ac']) \land ac_0 \subseteq ac' && \{\text{Predicate calculus}\}\\
&= \exists\, ac_0 \bullet (P[ac_0/ac'] \land ac_0 \subseteq ac') \lor (Q[ac_0/ac'] \land ac_0 \subseteq ac')\\
& && \{\text{Predicate calculus}\}\\
&= \exists\, ac_0 \bullet (P[ac_0/ac'] \land ac_0 \subseteq ac') \lor \exists\, ac_0 \bullet (Q[ac_0/ac'] \land ac_0 \subseteq ac')\\
& && \{\text{Definition of sequential composition}\}\\
&= (P \ ;\ ac \subseteq ac') \lor (Q \ ;\ ac \subseteq ac') && \{\text{Definition of } \mathbf{PBMH}\}\\
&= \mathbf{PBMH}(P) \lor \mathbf{PBMH}(Q)
\end{aligned}$$

$\square$

Law D.2.2 (PBMH-distribute-conjunction) *Provided P and Q satisfy* **PBMH**.

$$\mathbf{PBMH}(P \land Q) = \mathbf{PBMH}(P) \land \mathbf{PBMH}(Q)$$

Proof.

$$\begin{aligned}
&\mathbf{PBMH}(P \land Q) && \{\text{Assumption: } P \text{ and } Q \text{ satisfy } \mathbf{PBMH}\}\\
&= \mathbf{PBMH}((\mathbf{PBMH}(P) \land \mathbf{PBMH}(Q))) && \{\text{Definition of } \mathbf{PBMH}\}\\
&= ((P \ ;\ ac \subseteq ac') \land (Q \ ;\ ac \subseteq ac')) \ ;\ ac \subseteq ac'\\
& && \{\text{Definition of sequential composition}\}\\
&= \exists\, ac_0 \bullet \left(\begin{array}{l} \exists\, ac_1 \bullet (P[ac_1/ac'] \land ac_1 \subseteq ac') \\ \land \\ \exists\, ac_2 \bullet (Q[ac_2/ac'] \land ac_2 \subseteq ac') \end{array}\right)[ac_0/ac'] \land ac_0 \subseteq ac'\\
& && \{\text{Substitution}\}
\end{aligned}$$

$$= \exists\, ac_0 \bullet \left(\begin{array}{l} \exists\, ac_1 \bullet (P[ac_1/ac'] \wedge ac_1 \subseteq ac_0) \\ \wedge \\ \exists\, ac_2 \bullet (Q[ac_2/ac'] \wedge ac_2 \subseteq ac_0) \end{array} \right) \wedge ac_0 \subseteq ac'$$

{Transitivity of subset inclusion}

$$= \left(\begin{array}{l} \exists\, ac_1 \bullet (P[ac_1/ac'] \wedge ac_1 \subseteq ac') \\ \wedge \\ \exists\, ac_2 \bullet (Q[ac_2/ac'] \wedge ac_2 \subseteq ac') \end{array} \right)$$

{Definition of sequential composition}

$= (P \; ; \; ac \subseteq ac') \wedge (Q \; ; \; ac \subseteq ac')$ {Definition of **PBMH**}

$= \mathbf{PBMH}(P) \wedge \mathbf{PBMH}(Q)$

□

D.3 Closure properties

Law D.3.1 (PBMH-disjunction-closure) *Provided P and Q satisfy* **PBMH**.

$$\mathbf{PBMH}(P \vee Q) = P \vee Q$$

Proof.

$\mathbf{PBMH}(P \vee Q)$ {Law D.2.1}

$= \mathbf{PBMH}(P) \vee \mathbf{PBMH}(Q)$ {Assumption: P and Q satisfy **PBMH**}

$= P \vee Q$

□

Law D.3.2 (PBMH-conjunction-closure) *Provided P and Q satisfy* **PBMH**.

$$\mathbf{PBMH}(P \wedge Q) = P \wedge Q$$

Proof.

$\mathbf{PBMH}(P \wedge Q)$ {Assumption: P and Q satisfy **PBMH** and Law D.2.2}

$= \mathbf{PBMH}(P) \wedge \mathbf{PBMH}(Q)$ {Assumption: P and Q satisfy **PBMH**}

$= P \wedge Q$

□

D.4 Lemmas

Lemma D.4.1

$$\mathbf{PBMH}(\mathit{true}) = \mathit{true}$$

Proof.

$$
\begin{aligned}
&\mathbf{PBMH}(\mathit{true}) && \{\text{Definition of } \mathbf{PBMH}\}\\
&= \mathit{true} \,;\, ac \subseteq ac' && \{\text{Definition of sequential composition}\}\\
&= \exists\, ac_0 \bullet \mathit{true}[ac_0/ac'] \wedge ac_0 \subseteq ac'\\
& && \{\text{Property of substitution and predicate calculus}\}\\
&= \mathit{true}
\end{aligned}
$$

□

Lemma D.4.2

$$\mathbf{PBMH}(\mathit{false}) = \mathit{false}$$

Proof.

$$
\begin{aligned}
&\mathbf{PBMH}(\mathit{false}) && \{\text{Definition of } \mathbf{PBMH}\}\\
&= \mathit{false} \,;\, ac \subseteq ac' && \{\text{Definition of sequential composition}\}\\
&= \exists\, ac_0 \bullet \mathit{false}[ac_0/ac'] \wedge ac_0 \subseteq ac'\\
& && \{\text{Property of substitution and predicate calculus}\}\\
&= \mathit{false}
\end{aligned}
$$

□

Lemma D.4.3

$$\mathbf{PBMH}(s \in ac') = s \in ac'$$

Proof.

$$
\begin{aligned}
&\mathbf{PBMH}(s \in ac') && \{\text{Definition of } \mathbf{PBMH}\}\\
&= s \in ac' \,;\, ac \subseteq ac' && \{\text{Definition of sequential composition}\}\\
&= \exists\, ac_0 \bullet s \in ac_0 \wedge ac_0 \subseteq ac' && \{\text{Property of sets}\}\\
&= s \in ac'
\end{aligned}
$$

□

Lemma D.4.4

$$
\begin{array}{l}
(P \wedge ac' \neq \emptyset) \ ;_{\mathcal{A}} (Q \wedge ac' \neq \emptyset) \\
= \\
(P \wedge ac' \neq \emptyset) \ ;_{\mathcal{A}} (Q \wedge ac' \neq \emptyset)) \wedge ac' \neq \emptyset
\end{array}
$$

Proof.

$$
\begin{aligned}
& (P \wedge ac' \neq \emptyset) \ ;_{\mathcal{A}} (Q \wedge ac' \neq \emptyset) \\
& \hspace{20em} \{\text{Definition of } ;_{\mathcal{A}}\} \\
& = (P \wedge ac' \neq \emptyset)[\{z \mid Q \wedge ac' \neq \emptyset)[z/s]\}/ac'] \hspace{6em} \{\text{Substitution}\} \\
& = (P \wedge ac' \neq \emptyset)[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \hspace{6em} \{\text{Substitution}\} \\
& = \left(\begin{array}{l} P[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \\ \wedge \\ \{z \mid Q[z/s] \wedge ac' \neq \emptyset\} \neq \emptyset \end{array} \right) \hspace{4em} \{\text{Propositional calculus}\} \\
& = \left(\begin{array}{l} P[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \\ \wedge \\ \exists\, z \bullet z \in \{z \mid Q[z/s] \wedge ac' \neq \emptyset\} \end{array} \right) \hspace{4em} \{\text{Property of sets}\} \\
& = \left(\begin{array}{l} P[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \\ \wedge \\ \exists\, z \bullet Q[z/s] \wedge ac' \neq \emptyset \end{array} \right) \\
& \hspace{8em} \{\text{Predicate calculus: quantifier scope and duplicate term}\} \\
& = \left(\begin{array}{l} P[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \\ \wedge \\ (\exists\, z \bullet Q[z/s] \wedge ac' \neq \emptyset) \end{array} \right) \wedge ac' \neq \emptyset \hspace{4em} \{\text{Property of sets}\} \\
& = \left(\begin{array}{l} P[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac'] \\ \wedge \\ \{z \mid Q[z/s] \wedge ac' \neq \emptyset\} \neq \emptyset \end{array} \right) \wedge ac' \neq \emptyset \\
& \hspace{14em} \{\text{Re-introduce } ac' \text{ and substitution}\} \\
& = ((P \wedge ac' \neq \emptyset)[\{z \mid Q[z/s] \wedge ac' \neq \emptyset\}/ac']) \wedge ac' \neq \emptyset \hspace{3em} \{\text{Substitution}\} \\
& = ((P \wedge ac' \neq \emptyset)[\{z \mid (Q \wedge ac' \neq \emptyset)[z/s]\}/ac']) \wedge ac' \neq \emptyset \\
& \hspace{20em} \{\text{Definition of } ;_{\mathcal{A}}\} \\
& = ((P \wedge ac' \neq \emptyset) \ ;_{\mathcal{A}} (Q \wedge ac' \neq \emptyset)) \wedge ac' \neq \emptyset
\end{aligned}
$$

□

Lemma D.4.5

$$\mathbf{PBMH}(ac' \neq \emptyset) = ac' \neq \emptyset$$

Proof.

$\mathbf{PBMH}(ac' \neq \emptyset)$ {Definition of **PBMH**}

$= ac' \neq \emptyset \; ; \; ac \subseteq ac'$ {Definition of sequential composition}

$= \exists\, ac_0 \bullet ac_0 \neq \emptyset \land ac_0 \subseteq ac'$ {Property of sets (Lemma D.5.3)}

$= ac' \neq \emptyset$

□

Lemma D.4.6 *Provided ac' is not free in P.*

$$\mathbf{PBMH}(P) = P$$

Proof.

$\mathbf{PBMH}(P)$ {Definition of **PBMH**}

$= P \; ; \; ac \subseteq ac'$ {Definition of sequential composition}

$= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac'$

{Assumption: ac' not free in P and predicate calculus}

$= P \land \exists\, ac_0 \bullet ac_0 \subseteq ac'$ {Case-analysis on ac_0}

$= P$

□

Lemma D.4.7

$$P \Rightarrow \mathbf{PBMH}(P)$$

Proof.

P {Introduce fresh variable}

$= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 = ac'$

{Property of sets and propositional calculus}

$\Rightarrow \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac'$ {Definition of sequential composition}

$= P \; ; \; ac \subseteq ac'$ {Definition of **PBMH**}

$= \mathbf{PBMH}(P)$

□

Lemma D.4.8

$$\mathbf{PBMH}(P \mathbin{;} ac = \emptyset) = P \mathbin{;} ac = \emptyset$$

Proof.

$\mathbf{PBMH}(P \mathbin{;} ac = \emptyset)$ {Definition of **PBMH**}

$= (P \mathbin{;} ac = \emptyset) \mathbin{;} ac \subseteq ac'$ {Associativity of sequential composition}

$= P \mathbin{;} (ac = \emptyset \mathbin{;} ac \subseteq ac')$ {Definition of sequential composition}

$= P \mathbin{;} (\exists\, ac_0 \bullet ac = \emptyset \land ac_0 \subseteq ac')$ {Propositional calculus}

$= P \mathbin{;} (ac = \emptyset \land \exists\, ac_0 \bullet ac_0 \subseteq ac')$ {Choose $ac_0 = \emptyset$}

$= P \mathbin{;} (ac = \emptyset \land \mathit{true})$ {Propositional calculus}

$= P \mathbin{;} ac = \emptyset$

□

D.5 Set theory

Lemma D.5.1 (⊆-transitivity-multiple)

$\exists\, D \bullet (\exists\, A \bullet P(A) \land A \subseteq D) \land (\exists\, B \bullet P(B) \land B \subseteq D) \land D \subseteq E$

$= (\exists\, A \bullet P(A) \land A \subseteq E) \land (\exists\, B \bullet P(B) \land B \subseteq E)$

Proof. (Implication)

$\exists\, D \bullet (\exists\, A \bullet P(A) \land A \subseteq D) \land (\exists\, B \bullet P(B) \land B \subseteq D) \land D \subseteq E$

{Propositional calculus}

$\Rightarrow (\exists\, D, A \bullet P(A) \land A \subseteq D \land D \subseteq E) \land (\exists\, D, B \bullet P(B) \land B \subseteq D \land D \subseteq E)$

{Propositional calculus and transitivity of subset inclusion}

$= (\exists\, A \bullet P(A) \land A \subseteq E) \land (\exists\, B \bullet P(B) \land B \subseteq E)$

□

Proof. (Reverse implication)

$$\left(\begin{array}{l} (\exists\, A \bullet P(A) \land A \subseteq E) \land (\exists\, B \bullet P(B) \land B \subseteq E) \\ \Rightarrow \exists\, D \bullet (\exists\, A \bullet P(A) \land A \subseteq D) \land (\exists\, B \bullet P(B) \land B \subseteq D) \land D \subseteq E \end{array}\right)$$

{Set $D = E$}

$$= \left(\begin{array}{l} (\exists A \bullet P(A) \wedge A \subseteq E) \wedge (\exists B \bullet P(B) \wedge B \subseteq E) \\ \Rightarrow (\exists A \bullet P(A) \wedge A \subseteq E) \wedge (\exists B \bullet P(B) \wedge B \subseteq E) \wedge E \subseteq E \end{array} \right)$$

{Reflexivity of subset inclusion and propositional calculus}

$= true$

□

Lemma D.5.2

$s \in A \Rightarrow A \neq \emptyset$

Proof.

$s \in A \Rightarrow A \neq \emptyset$ {Property of sets}

$= s \in A \Rightarrow \exists z \bullet z \in A$ {Choose $z = s$}

$= s \in A \Rightarrow s \in A$ {Propositional calculus}

$= true$

□

Lemma D.5.3

$\exists B \bullet B \neq \emptyset \wedge B \subseteq C \Leftrightarrow C \neq \emptyset$

Proof. (Implication) By contradiction: Suppose the consequent is false yet the antecedent is true. Then $C = \emptyset$.

$\exists B \bullet B \neq \emptyset \wedge B \subseteq C$ {Assumption: $C = \emptyset$}

$= \exists B \bullet B \neq \emptyset \wedge B \subseteq \emptyset$ {Property of subset inclusion}

$= \exists B \bullet B \neq \emptyset \wedge B = \emptyset$ {Propositional calculus}

$= false$

□

Proof. (Reverse implication)

$C \neq \emptyset \Rightarrow \exists B \bullet B \neq \emptyset \wedge B \subseteq C$ {Choose $B = C$}

$= C \neq \emptyset \Rightarrow C \neq \emptyset \wedge C \subset C$ {Reflexivity of subset inclusion}

$= C \neq \emptyset \Rightarrow C \neq \emptyset$ {Propositional calculus}

$= true$

□

Lemma D.5.4

$\exists\, ac_0 \bullet s \in ac_0 \land ac_0 \subseteq ac' \Leftrightarrow s \in ac'$

Proof. (Implication)

$$\begin{aligned}
&\exists\, ac_0 \bullet s \in ac_0 \land ac_0 \subseteq ac' && \{\text{Definition of subset inclusion}\}\\
&= \exists\, ac_0 \bullet s \in ac_0 \land (\forall\, z \bullet z \in ac_0 \Rightarrow z \in ac')\\
& && \{\text{Assume } s \in ac_0 \text{ then there is a case when } z = s\}\\
&= \exists\, ac_0 \bullet s \in ac_0 \land (\forall\, z \bullet z \in ac_0 \Rightarrow z \in ac') \land (s \in ac_0 \Rightarrow s \in ac')\\
& && \{\text{Assume } s \in ac_0 \text{ and propositional calculus}\}\\
&\Rightarrow s \in ac'
\end{aligned}$$

□

Proof. (Reverse implication)

$$\begin{aligned}
&s \in ac' \Rightarrow (\exists\, ac_0 \bullet s \in ac_0 \land ac_0 \subseteq ac') && \{\text{Choose } ac_0 = ac'\}\\
&= (s \in ac') \Rightarrow (s \in ac' \land ac' \subseteq ac')\\
& && \{\text{Reflexivity of subset inclusion and propositional calculus}\}\\
&= \mathit{true}
\end{aligned}$$

□

Appendix E

Sequential composition ($\mathcal{A}$)

E.1 Algebraic properties

Law E.1.1 (; $_\mathcal{A}$-ac'-not-free) *Provided ac' is not free in P.*

$$P \ ;_\mathcal{A} Q = P$$

Proof.

$$
\begin{aligned}
& P \ ;_\mathcal{A} Q && \{\text{Definition of } ;_\mathcal{A}\} \\
&= P[\{z : State \mid Q[z/s]\}/ac'] && \{\text{Assumption: } ac' \text{ not free in } P\} \\
&= P
\end{aligned}
$$

□

Law E.1.2 (; $_\mathcal{A}$-associativity) *Provided P and Q satisfy* **PBMH**.

$$P \ ;_\mathcal{A} (Q \ ;_\mathcal{A} R) = (P \ ;_\mathcal{A} Q) \ ;_\mathcal{A} R$$

Proof.

$$
\begin{aligned}
& (P \ ;_\mathcal{A} Q) \ ;_\mathcal{A} R && \{\text{Definition of sequential composition, twice}\} \\
&= (P[\{z \mid Q[z/s]\}/ac'])[\{z \mid R[z/s]\}/ac'] \\
& && \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\} \\
&= (P \ ; \ ac \subseteq ac')[\{z \mid Q[z/s]\}/ac'])[\{z \mid R[z/s]\}/ac'] && \{\text{Substitution}\} \\
&= (P \ ; \ ac \subseteq \{z \mid Q[z/s]\})[\{z \mid R[z/s]\}/ac'] \\
& && \{\text{Definition of subset inclusion and property of sets}\}
\end{aligned}
$$

$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow Q[z/s])[\{z \mid R[z/s]\}/ac']$$
{Assumption: Q satisfies **PBMH**}
$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow (Q[z/s] \,;\, ac \subseteq ac'))[\{z \mid R[z/s]\}/ac']$$
{Substitution}
$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow (Q[z/s] \,;\, ac \subseteq \{z \mid R[z/s]\}))$$
{Re-introduce ac'}
$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow (Q[z/s] \,;\, ac \subseteq ac')[\{z \mid R[z/s]\}/ac']))$$
{Assumption: Q satisfies **PBMH**, and definition of sequential composition}
$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow (Q[z/s] \;;_{\mathcal{A}} R))$$
{Re-introduce ac'}
$$= (P \,;\, \forall z \bullet z \in ac \Rightarrow z \in ac')[\{z \mid Q[z/s] \;;_{\mathcal{A}} R\}/ac']$$
{Definition of subset inclusion and sequential composition}
$$= (P \,;\, ac \subseteq ac') \;;_{\mathcal{A}} (Q \;;_{\mathcal{A}} R)$$
{Assumption: P satisfies **PBMH**}
$$= P \;;_{\mathcal{A}} (Q \;;_{\mathcal{A}} R)$$

□

Law E.1.3 ($;_{\mathcal{A}}$-negation)

$$\neg (P \;;_{\mathcal{A}} Q) = (\neg P \;;_{\mathcal{A}} Q)$$

Proof.

$$\neg (P \;;_{\mathcal{A}} Q)$$
{Definition of sequential composition}
$$= \neg (P[\{z \mid Q[z/s]\}/ac'])$$
{Propositional calculus}
$$= (\neg P[\{z \mid Q[z/s]\}/ac'])$$
{Definition of sequential composition}
$$= (\neg P \;;_{\mathcal{A}} Q)$$

□

E.2 Closure properties

Law E.2.1 ($;_{\mathcal{A}}$-closure) *Provided P and Q satisfy* **PBMH**.

$$\mathbf{PBMH}(P \;;_{\mathcal{A}} Q) = P \;;_{\mathcal{A}} Q$$

Proof. (Implication)

$$\mathbf{PBMH}(P \;;_{\mathcal{A}} Q)$$
{Definition of **PBMH**}

$$= (P \ ;_{\mathcal{A}} Q) \ ; \ ac \subseteq ac' \qquad \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\}$$

$$= ((P \ ; \ ac \subseteq ac') \ ;_{\mathcal{A}} Q) \ ; \ ac \subseteq ac'$$

$$\{\text{Definition of sequential composition}\}$$

$$= ((\exists ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \ ;_{\mathcal{A}} Q) \ ; \ ac \subseteq ac'$$

$$\{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\}$$

$$= (\exists ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid Q[z/s]\}) \ ; \ ac \subseteq ac'$$

$$\{\text{Assumption: } Q \text{ satisfies } \mathbf{PBMH}\}$$

$$= (\exists ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid (Q \ ; \ ac \subseteq ac')[z/s]\}) \ ; \ ac \subseteq ac'$$

$$\{\text{Definition of subset inclusion}\}$$

$$= \left(\left(\begin{array}{l} \exists ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall z \bullet z \in ac_0 \Rightarrow ((Q \ ; \ ac \subseteq ac')[z/s]) \end{array} \right) \ ; \ ac \subseteq ac' \right)$$

$$\{\text{Definition of sequential composition}\}$$

$$= \exists ac_1 \bullet \left(\begin{array}{l} \left(\begin{array}{l} \exists ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall z \bullet \left(\begin{array}{l} z \in ac_0 \\ \Rightarrow \\ ((\exists ac_0 \bullet Q[ac_0/ac'] \land ac_0 \subseteq ac')[z/s]) \end{array} \right) \end{array} \right) [ac_1/ac'] \\ \land ac_1 \subseteq ac' \end{array} \right)$$

$$\{\text{Substitution}\}$$

$$= \left(\begin{array}{l} \exists ac_0, ac_1 \bullet P[ac_0/ac'] \\ \land \\ \forall z \bullet \left(\begin{array}{l} z \in ac_0 \\ \Rightarrow \\ (\exists ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac_1) \end{array} \right) \land ac_1 \subseteq ac' \end{array} \right)$$

$$\{\text{Predicate calculus: quantifier scope}\}$$

$$= \left(\begin{array}{l} \exists ac_0 \bullet P[ac_0/ac'] \\ \land \\ \exists ac_1 \bullet \forall z \bullet \left(\begin{array}{l} z \in ac_0 \\ \Rightarrow \\ (\exists ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac_1) \end{array} \right) \land ac_1 \subseteq ac' \end{array} \right)$$

$$\{\text{Predicate calculus}\}$$

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \exists\, ac_1 \bullet \forall\, z \bullet \left(\begin{array}{l} (z \notin ac_0 \land ac_1 \subseteq ac') \\ \lor \\ (\exists\, ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac_1 \land ac_1 \subseteq ac') \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$\Rightarrow \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet \exists\, ac_1 \bullet \left(\begin{array}{l} (z \notin ac_0 \land ac_1 \subseteq ac') \\ \lor \\ (\exists\, ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac_1 \land ac_1 \subseteq ac') \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet \left(\begin{array}{l} (\exists\, ac_1 \bullet z \notin ac_0 \land ac_1 \subseteq ac') \\ \lor \\ (\exists\, ac_0, ac_1 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac_1 \land ac_1 \subseteq ac') \end{array} \right) \end{array} \right)$$

{Property of sets and predicate calculus}

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet \left(\begin{array}{l} (z \notin ac_0) \\ \lor \\ (\exists\, ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac') \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet z \in ac_0 \Rightarrow (\exists\, ac_0 \bullet Q[ac_0/ac'][z/s] \land ac_0 \subseteq ac') \end{array} \right)$$

{Substitution}

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet z \in ac_0 \Rightarrow (\exists\, ac_0 \bullet Q[ac_0/ac'] \land ac_0 \subseteq ac')[z/s] \end{array} \right)$$

{Definition of sequential composition}

$$= \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0/ac'] \\ \land \\ \forall\, z \bullet z \in ac_0 \Rightarrow (Q \;;\; ac \subseteq ac')[z/s] \end{array} \right)$$

{Assumption: Q satisfies **PBMH**}

$= \exists\, ac_0 \bullet P[ac_0/ac'] \wedge \forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]$
{Definition of subset inclusion and re-introduce set comprehension}
$= \exists\, ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq \{z \mid Q[z/s]\}$
{Re-introduce ac', definition of $;_{\mathcal{A}}$ and substitution}
$= (\exists\, ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq ac')[\{z \mid Q[z/s]\}/ac']$
{Definition of sequential composition}
$= (P \,;\, ac_0 \subseteq ac') \;;_{\mathcal{A}} Q$ {Assumption: P satisfies **PBMH**}
$= P \operatorname{seq} Q$

□

Proof. (Reverse implication)

$P \;;_{\mathcal{A}} Q$ {Lemma D.4.7}
$\Rightarrow \mathbf{PBMH}(P \;;_{\mathcal{A}} Q)$

□

E.3 Distributivity with respect to disjunction

Law E.3.1 ($;_{\mathcal{A}}$-right-distributivity-disjunction)

$$(P \vee Q) \;;_{\mathcal{A}} R = (P \;;_{\mathcal{A}} R) \vee (Q \;;_{\mathcal{A}} R)$$

Proof.

$(P \vee Q) \;;_{\mathcal{A}} R$ {Definition of sequential composition}
$= (P \vee Q)[\{z \mid R[z/s]\}/ac']$ {Substitution}
$= (P[\{z \mid R[z/s]\}/ac'] \vee Q[\{z \mid R[z/s]\}/ac'])$
{Definition of sequential composition}
$= (P \;;_{\mathcal{A}} R) \vee (Q \;;_{\mathcal{A}} R)$

□

E.4 Distributivity with respect to conjunction

Law E.4.1 ($;_{\mathcal{A}}$-right-distributivity-conjunction)

$$(P \wedge Q) \;;_{\mathcal{A}} R = (P \;;_{\mathcal{A}} R) \wedge (Q \;;_{\mathcal{A}} R)$$

Proof.

$$\begin{aligned}
&(P \wedge Q) \;;_{\mathcal{A}} R && \{\text{Definition of } ;_{\mathcal{A}}\}\\
&= (P \wedge Q)[\{z \mid R[z/s]\}/ac'] && \{\text{Property of substitution}\}\\
&= (P[\{z \mid R[z/s]\}/ac'] \wedge Q[\{z \mid R[z/s]\}/ac']) && \{\text{Definition of } ;_{\mathcal{A}}\}\\
&= (P \;;_{\mathcal{A}} R) \wedge (Q \;;_{\mathcal{A}} R)
\end{aligned}$$

□

Law E.4.2 *Provided P satisfies* **PBMH2**. *This property does not necessarily hold in the opposite direction (See Example 19).*

$$P \;;_{\mathcal{A}} (Q \wedge R) \Rightarrow (P \;;_{\mathcal{A}} Q) \wedge (P \;;_{\mathcal{A}} R)$$

Proof. (Implication) To be revised. □

Example 19

$$\begin{aligned}
&((ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 1) \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 2)) \Rightarrow (ac' \neq \emptyset \;;_{\mathcal{A}} (s.x = 1 \wedge s.x = 2))\\
&\qquad \{\text{Propositional calculus}\}\\
&= ((ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 1) \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 2)) \Rightarrow (ac' \neq \emptyset \;;_{\mathcal{A}} \mathit{false})\\
&\qquad \{\text{Definition of sequential composition}\}\\
&= ((ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 1) \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 2)) \Rightarrow (ac' \neq \emptyset[\{z \mid \mathit{false}\}/ac'])\\
&\qquad \{\text{Property of sets, substitution and propositional calculus}\}\\
&= ((ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 1) \wedge (ac' \neq \emptyset \;;_{\mathcal{A}} s.x = 2)) \Rightarrow \mathit{false}\\
&\qquad \{\text{Definition of sequential composition and substitution}\}\\
&= ((\{z \mid z.x = 1\} \neq \emptyset) \wedge (\{z \mid z.x = 2\} \neq \emptyset)) \Rightarrow \mathit{false} && \{\text{Property of sets}\}\\
&= ((\exists z \bullet z.x = 1) \wedge (\exists z \bullet z.x = 2)) \Rightarrow \mathit{false} && \{\text{One-point}\}\\
&= \mathit{true} \Rightarrow \mathit{false} && \{\text{Propositional calculus}\}\\
&= \mathit{false}
\end{aligned}$$

E.5 Extreme points

Law E.5.1 (; $_{\mathcal{A}}$-P-sequence-*false*:1) *Provided P satisfies* **PBMH**.

$$P \ ;_{\mathcal{A}} \mathbf{false} = P[\emptyset/ac']$$

Proof.

$$
\begin{array}{lr}
P \ ;_{\mathcal{A}} \mathbf{false} & \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\} \\
= (P \ ; \ ac \subseteq ac') \ ;_{\mathcal{A}} \mathit{false} & \{\text{Definition of sequential composition}\} \\
= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \ ;_{\mathcal{A}} \mathit{false} & \{\text{Definition of } ;_{\mathcal{A}}\} \\
= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \emptyset & \{\text{Property of sets and one-point rule}\} \\
= P[\emptyset/ac']
\end{array}
$$

□

Law E.5.2 (; $_{\mathcal{A}}$-P-sequence-*true*) *Provided P satisfies* **PBMH**.

$$P \ ;_{\mathcal{A}} \mathbf{true} = \exists\, ac' \bullet P$$

Proof.

$$
\begin{array}{lr}
P \ ;_{\mathcal{A}} \mathbf{true} & \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\} \\
= (P \ ; \ ac \subseteq ac') \ ;_{\mathcal{A}} \mathit{true} & \{\text{Definition of sequential composition}\} \\
= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \ ;_{\mathcal{A}} \mathit{true} & \{\text{Definition of } ;_{\mathcal{A}}\} \\
= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid \mathit{true}\} & \{\text{Property of sets}\} \\
= \exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow \mathit{true}) & \{\text{Propositional calculus}\} \\
= \exists\, ac_0 \bullet P[ac_0/ac'] & \{\text{One-point rule}\} \\
= \exists\, ac_0 \bullet (\exists\, ac' \bullet P \land ac' = ac_0) & \{\text{One-point rule: } ac_0 \text{ not free in } P\} \\
= \exists\, ac' \bullet P
\end{array}
$$

□

E.6 Algebraic properties and sequential composition

Law E.6.1 (; $_{\mathcal{A}}$-sequence-left-associativity) *Provided ok and ac are not free in R.*

$$(P \ ; \ Q) \ ;_{\mathcal{A}} R = P \ ; (Q \ ;_{\mathcal{A}} R)$$

Proof.

$(P \mathbin{;} Q) \mathbin{;_{\mathcal{A}}} R$ {Definition of sequential composition}

$= (\exists\, ok_0, ac_0 \bullet P[ok_0, ac_0/ok, ac'] \wedge Q[ok_0, ac_0/ok, ac]) \mathbin{;_{\mathcal{A}}} R$

{Definition of $;_{\mathcal{A}}$}

$= (\exists\, ok_0, ac_0 \bullet P[ok_0, ac_0/ok, ac'] \wedge Q[ok_0, ac_0/ok, ac])[\{z \mid R[z/s]\}/ac']$

{Substitution: ac' not free in ac_0}

$= (\exists\, ok_0, ac_0 \bullet P[ok_0, ac_0/ok, ac'] \wedge Q[ok_0, ac_0/ok, ac][\{z \mid R[z/s]\}/ac'])$

{Assumption: $\{ok, ac\}$ not free in R}

$= (\exists\, ok_0, ac_0 \bullet P[ok_0, ac_0/ok, ac'] \wedge Q[\{z \mid R[z/s]\}/ac'][ok_0, ac_0/ok, ac])$

{Definition of sequential composition}

$= P \mathbin{;} Q[\{z \mid R[z/s]\}/ac']$ {Definition of $;_{\mathcal{A}}$}

$= P \mathbin{;} (Q \mathbin{;_{\mathcal{A}}} R)$

□

E.7 Skip

Definition 68

$$\mathit{II}_{\mathcal{A}} \mathrel{\widehat{=}} s \in ac'$$

Law E.7.1 $\mathit{II}_{\mathcal{A}}$ *is a fixed point of* **PBMH**.

$$\mathbf{PBMH}(\mathit{II}_{\mathcal{A}}) = \mathit{II}_{\mathcal{A}}$$

Proof.

$\mathbf{PBMH}(\mathit{II}_{\mathcal{A}})$ {Definition of $\mathit{II}_{\mathcal{A}}$ and **PBMH**}

$= (s \in ac') \mathbin{;} (ac \subseteq ac')$

{Definition of sequential composition and substitution}

$= \exists\, ac_0 \bullet s \in ac_0 \wedge ac_0 \subseteq ac'$ {Law D.5.4}

$= s \in ac'$

□

Law E.7.2 ($;_{\mathcal{A}}$-$\mathit{II}_{\mathcal{A}}$-left-unit)

$$\mathit{II}_{\mathcal{A}} \mathbin{;_{\mathcal{A}}} P$$

Proof.

$$
\begin{aligned}
& I\!I_{\mathcal{A}} \;\, ;_{\mathcal{A}} P && \{\text{Definition of } I\!I_{\mathcal{A}}\} \\
& = s \in ac' \;\, ;_{\mathcal{A}} P && \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\} \\
& = s \in \{z \mid P[z/s]\} && \{\text{Property of sets}\} \\
& = P[z/s][s/z] && \{\text{Substitution}\} \\
& = P
\end{aligned}
$$

□

Law E.7.3 ($;_{\mathcal{A}}$-$I\!I_{\mathcal{A}}$-right-unit) *Provided P satisfies* **PBMH**.

$$P \;\, ;_{\mathcal{A}} I\!I_{\mathcal{A}}$$

Proof.

$$
\begin{aligned}
& P \;\, ;_{\mathcal{A}} I\!I_{\mathcal{A}} && \{\text{Definition of } I\!I_{\mathcal{A}}\} \\
& = P \;\, ;_{\mathcal{A}} (s \in ac') && \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\} \\
& = (P \; ; \; ac \subseteq ac') \;\, ;_{\mathcal{A}} (s \in ac') && \{\text{Definition of sequential composition}\} \\
& = (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;\, ;_{\mathcal{A}} (s \in ac') && \{\text{Definition of } ;_{\mathcal{A}}\} \\
& = \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid z \in ac'\} && \{\text{Property of sets}\} \\
& = \exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow z \in ac') \\
& && \{\text{Definition of subset inclusion}\} \\
& = \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac' && \{\text{Definition of sequential composition}\} \\
& = P \; ; \; ac \subseteq ac' && \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\} \\
& = P
\end{aligned}
$$

□

E.8 Other lemmas

Lemma E.8.1 *Provided P is* **PBMH**-*healthy and s is not free in e.*

$$P \;\, ;_{\mathcal{A}} (Q \Rightarrow (R \land e)) = (P \;\, ;_{\mathcal{A}} \neg\, Q) \lor ((P \;\, ;_{\mathcal{A}} (Q \Rightarrow R)) \land e)$$

Proof.

$P \;;_{\mathcal{A}} (Q \Rightarrow (R \land e))$

$$
\begin{aligned}
& \{\text{Assumption: } P \text{ is } \mathbf{PBMH}\text{-healthy (Lemma D.1.1)}\} \\
&= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \Rightarrow (R \land e)) \\
& \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\} \\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow (R \land e)\} \\
& \{\text{Property of sets and } s \text{ not free in } e\} \\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow (R[z/s] \land e)) \\
& \{\text{Lemma E.8.4}\} \\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \left(\begin{array}{l} (\forall\, z \bullet z \in ac_0 \Rightarrow \neg\, Q[z/s]) \\ \lor \\ ((\forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow R[z/s])) \land e) \end{array}\right) \\
& \{\text{Predicate calculus}\} \\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow \neg\, Q[z/s])) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ((\forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow R[z/s])) \land e)) \end{array}\right) \\
& \{\text{Property of sets}\} \\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow z \in \{s \mid \neg\, Q\})) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ((\forall\, z \bullet z \in ac_0 \Rightarrow z \in \{s \mid Q \Rightarrow R\}) \land e)) \end{array}\right) \\
& \{\text{Property of sets}\} \\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid \neg\, Q\}) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow R\} \land e) \end{array}\right) \\
& \{\text{Predicate calculus}\} \\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid \neg\, Q\}) \\ \lor \\ ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow R\}) \land e) \end{array}\right) \\
& \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\} \\
&= \left(\begin{array}{l} ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} \neg\, Q) \\ \lor \\ (((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \Rightarrow R)) \land e) \end{array}\right) \\
& \{\text{Assumption: } P \text{ is } \mathbf{PBMH}\text{-healthy (Lemma D.1.1)}\}
\end{aligned}
$$

$$= (P \;;_{\mathcal{A}} \neg\, Q) \lor ((P \;;_{\mathcal{A}} (Q \Rightarrow R)) \land e)$$

□

Lemma E.8.2 *Provided P satisfies* **PBMH**.

$$P \;;_{\mathcal{A}} (Q \land ok') = (P \;;_{\mathcal{A}} \mathit{false}) \lor ((P \;;_{\mathcal{A}} Q) \land ok')$$

Proof.

$$
\begin{aligned}
& P \;;_{\mathcal{A}} (Q \land ok') && \{\text{Assumption: } P \text{ satisfies } \mathbf{PBMH}\}\\
&= \mathbf{PBMH}(P) \;;_{\mathcal{A}} (Q \land ok') && \{\text{Definition of } \mathbf{PBMH}\}\\
&= (P \,;\, ac \subseteq ac') \;;_{\mathcal{A}} (Q \land ok') \\
& && \{\text{Definition of sequential composition and substitution}\}\\
&= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \land ok') \\
& && \{\text{Defintiion of } ;_{\mathcal{A}} \text{ and substitution}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid (Q \land ok')[z/s]\} \\
& && \{\text{Property of substitution}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid Q[z/s] \land ok'\} \\
& && \{\text{Propositional calculus}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land (\forall\, z \bullet z \in ac_0 \Rightarrow ok') \\
& && \{\text{Propositional calculus}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land (\forall\, z \bullet z \notin ac_0 \lor ok') \\
& && \{\text{Predicate calculus: } ok' \text{ not in } z\text{, move quantifier}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land ((\forall\, z \bullet z \notin ac_0) \lor ok') \\
& && \{\text{Predicate calculus: distribution}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \left(\begin{array}{l} ((\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land (\forall\, z \bullet z \notin ac_0)) \\ \lor \\ ((\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land ok') \end{array}\right) \\
& && \{\text{Propositional calculus}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \left(\begin{array}{l} (\forall\, z \bullet (z \in ac_0 \Rightarrow Q[z/s]) \land z \notin ac_0) \\ \lor \\ ((\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land ok') \end{array}\right) \\
& && \{\text{Propositional calculus}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land ((\forall\, z \bullet z \notin ac_0) \lor ((\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land ok')) \\
& && \{\text{Propositional calculus}\}
\end{aligned}
$$

$$= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land \forall\, z \bullet z \notin ac_0) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow Q[z/s]) \land ok') \end{array} \right)$$

{Property of sets and introduce set comprehension}

$$= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 = \emptyset) \\ \lor \\ ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid Q[z/s]\}) \land ok') \end{array} \right)$$

{One-point rule and substitution}

$$= \left(\begin{array}{l} P[\emptyset/ac'] \\ \lor \\ ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{z \mid Q[z/s]\}) \land ok') \end{array} \right)$$

{Re-introduce ac'}

$$= \left(\begin{array}{l} P[\emptyset/ac'] \\ \lor \\ ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac')[\{z \mid Q[z/s]\}/ac'] \land ok') \end{array} \right)$$

{Definition of $\;;_{\mathcal{A}}$}

$$= \left(\begin{array}{l} P[\emptyset/ac'] \\ \lor \\ (((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} Q) \land ok') \end{array} \right)$$

{Definition of sequential composition}

$$= P[\emptyset/ac'] \lor (((P \,;\, ac \subseteq ac') \;;_{\mathcal{A}} Q) \land ok')$$

{Assumption: P satisfies **PBMH**}

$$= P[\emptyset/ac'] \lor ((P \;;_{\mathcal{A}} Q) \land ok')$$ {Law E.5.1}

$$= (P \;;_{\mathcal{A}} \mathit{false}) \lor ((P \;;_{\mathcal{A}} Q) \land ok')$$

□

Lemma E.8.3 *Provided P is* **PBMH***-healthy.*

$$P \;;_{\mathcal{A}} (Q \Rightarrow (R \land ok')) = (P \;;_{\mathcal{A}} \neg\, Q) \lor ((P \;;_{\mathcal{A}} (Q \Rightarrow R)) \land ok')$$

Proof.

$$P \;;_{\mathcal{A}} (Q \Rightarrow (R \land ok'))$$

{Assumption: P is **PBMH**-healthy (Lemma D.1.1)}

$$= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \Rightarrow (R \land ok'))$$

{Definition of $\;;_{\mathcal{A}}$ and substitution}

$$
\begin{aligned}
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow (R \land ok')\} && \{\text{Property of sets}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow (R[z/s] \land ok'))\\
& && \{\text{Lemma E.8.4}\}\\
&= \exists\, ac_0 \bullet P[ac_0/ac'] \land \left(\begin{array}{l} (\forall\, z \bullet z \in ac_0 \Rightarrow \neg\, Q[z/s]) \\ \lor \\ ((\forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow R[z/s])) \land ok') \end{array}\right)\\
& && \{\text{Predicate calculus}\}\\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow \neg\, Q[z/s])) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ((\forall\, z \bullet z \in ac_0 \Rightarrow (Q[z/s] \Rightarrow R[z/s])) \land ok')) \end{array}\right)\\
& && \{\text{Property of sets}\}\\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land (\forall\, z \bullet z \in ac_0 \Rightarrow z \in \{s \mid \neg\, Q\})) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ((\forall\, z \bullet z \in ac_0 \Rightarrow z \in \{s \mid Q \Rightarrow R\}) \land ok')) \end{array}\right)\\
& && \{\text{Property of sets}\}\\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid \neg\, Q\}) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow R\} \land ok') \end{array}\right)\\
& && \{\text{Predicate calculus}\}\\
&= \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid \neg\, Q\}) \\ \lor \\ ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \Rightarrow R\}) \land ok') \end{array}\right)\\
& && \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\}\\
&= \left(\begin{array}{l} ((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} \neg\, Q) \\ \lor \\ (((\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \Rightarrow R)) \land ok') \end{array}\right)\\
& && \{\text{Assumption: } P \text{ is } \mathbf{PBMH}\text{-healthy (Lemma D.1.1)}\}\\
&= (P \;;_{\mathcal{A}} \neg\, Q) \lor ((P \;;_{\mathcal{A}} (Q \Rightarrow R)) \land ok')
\end{aligned}
$$

□

Lemma E.8.4 ✓$_P$ ✓$_A$ *Provided x is not free in e*

$$
\begin{array}{l}
\forall\, x \bullet P \Rightarrow (Q \Rightarrow (R \land e)) \\
= \\
(\forall\, x \bullet P \Rightarrow \neg\, Q) \lor ((\forall\, x \bullet P \Rightarrow (Q \Rightarrow R)) \land e)
\end{array}
$$

Proof.

$$
\begin{array}{ll}
\forall x \bullet P \Rightarrow (Q \Rightarrow (R \wedge e)) & \{\text{Predicate calculus}\} \\
= \forall x \bullet (P \wedge Q) \Rightarrow (R \wedge e) & \{\text{Predicate calculus}\} \\
= \forall x \bullet ((P \wedge Q) \Rightarrow R) \wedge ((P \wedge Q) \Rightarrow e) & \{\text{Predicate calculus}\} \\
= \forall x \bullet ((P \wedge Q) \Rightarrow R) \wedge (\neg (P \wedge Q) \vee e) & \{\text{Predicate calculus}\} \\
= (\forall x \bullet (P \wedge Q) \Rightarrow R) \wedge (\forall x \bullet \neg (P \wedge Q) \vee e) & \\
& \{\text{Predicate calculus: } x \text{ is not free in } e\} \\
= (\forall x \bullet (P \wedge Q) \Rightarrow R) \wedge ((\forall x \bullet \neg (P \wedge Q)) \vee e) & \{\text{Predicate calculus}\} \\
= \left(\begin{array}{l} ((\forall x \bullet (P \wedge Q) \Rightarrow R) \wedge (\forall x \bullet \neg (P \wedge Q))) \\ \vee \\ ((\forall x \bullet (P \wedge Q) \Rightarrow R) \wedge e) \end{array} \right) & \{\text{Predicate calculus}\} \\
= \left(\begin{array}{l} (\forall x \bullet ((P \wedge Q) \Rightarrow R) \wedge \neg (P \wedge Q)) \\ \vee \\ ((\forall x \bullet ((P \wedge Q) \Rightarrow R)) \wedge e) \end{array} \right) & \{\text{Predicate calculus}\} \\
= \left(\begin{array}{l} (\forall x \bullet \neg (P \wedge Q)) \\ \vee \\ ((\forall x \bullet ((P \wedge Q) \Rightarrow R)) \wedge e) \end{array} \right) & \{\text{Predicate calculus}\} \\
= (\forall x \bullet P \Rightarrow \neg Q) \vee ((\forall x \bullet P \Rightarrow (Q \Rightarrow R)) \wedge e) &
\end{array}
$$

□

Lemma E.8.5 $\checkmark_P$ $\checkmark_A$ *Provided P is **PBMH**-healthy.*

$$(P \;;_{\mathcal{A}} Q) \vee (P \;;_{\mathcal{A}} R) \Rightarrow (P \;;_{\mathcal{A}} (Q \vee R))$$

Proof.

$$
\begin{array}{ll}
(P \;;_{\mathcal{A}} Q) \vee (P \;;_{\mathcal{A}} R) & \\
& \{\text{Assumption: } P \text{ is } \mathbf{PBMH}\text{-healthy (Lemma D.1.1)}\} \\
= \left(\begin{array}{l} ((\exists ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq ac') \;;_{\mathcal{A}} Q) \\ \vee \\ ((\exists ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq ac') \;;_{\mathcal{A}} R) \end{array} \right) & \\
& \{\text{Definition of } ;_{\mathcal{A}} \text{ and substitution}\} \\
= \left(\begin{array}{l} (\exists ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq \{s \mid Q\}) \\ \vee \\ (\exists ac_0 \bullet P[ac_0/ac'] \wedge ac_0 \subseteq \{s \mid R\}) \end{array} \right) & \{\text{Predicate calculus}\}
\end{array}
$$

$= \exists\, ac_0 \bullet P[ac_0/ac'] \land (ac_0 \subseteq \{s \mid Q\} \lor ac_0 \subseteq \{s \mid R\})$

{Property of sets and predicate calculus}

$\Rightarrow \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q\} \cup \{s \mid R\}$ {Property of sets}

$= \exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq \{s \mid Q \lor R\}$

{Definition of $\;_{\mathcal{A}}$ and substitution}

$= (\exists\, ac_0 \bullet P[ac_0/ac'] \land ac_0 \subseteq ac') \;;_{\mathcal{A}} (Q \lor R)$

{Assumption: P is **PBMH**-healthy (Lemma D.1.1)}

$= P \;;_{\mathcal{A}} (Q \lor R)$

$\square$

Theorem E.8.1 *Provided P is* **PBMH**-*healthy.*

$$(P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} true) = P \;;_{\mathcal{A}} true$$

Proof.

$(P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} true)$ {Lemma E.8.5}

$= ((P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} true)) \land (P \;;_{\mathcal{A}} (Q \lor true))$

{Predicate calculus}

$= ((P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} true)) \land (P \;;_{\mathcal{A}} true)$

{Predicate calculus: absorption law}

$= (P \;;_{\mathcal{A}} true)$

$\square$

Theorem E.8.2 *Provided P is* **PBMH**-*healthy.*

$$(P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} false) = P \;;_{\mathcal{A}} Q$$

Proof.

$(P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} false)$ {Lemma E.8.5}

$= ((P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} false)) \land (P \;;_{\mathcal{A}} (Q \lor false))$

{Predicate calculus}

$= ((P \;;_{\mathcal{A}} Q) \lor (P \;;_{\mathcal{A}} false)) \land (P \;;_{\mathcal{A}} Q)$

{Predicate calculus: absorption law}

$= P \;;_{\mathcal{A}} Q$

$\square$

Appendix F

Sequential composition ($;_{\mathcal{D}\mathbf{ac}}$)

F.1 Properties

Law F.1.1 *Provided P, Q and R are* **A**-*healthy.*

$$(P \ ;_{\mathcal{D}\mathbf{ac}} Q) \ ;_{\mathcal{D}\mathbf{ac}} R = P \ ;_{\mathcal{D}\mathbf{ac}} (Q \ ;_{\mathcal{D}\mathbf{ac}} R)$$

Proof.

$$(P \ ;_{\mathcal{D}\mathbf{ac}} Q) \ ;_{\mathcal{D}\mathbf{ac}} R =$$

{Assumption: P, D and R are designs}

$$= ((\neg P^f \vdash P^t) \ ;_{\mathcal{D}\mathbf{ac}} (\neg Q^f \vdash Q^t)) \ ;_{\mathcal{D}\mathbf{ac}} (\neg R^f \vdash R^t)$$

{Definition of sequential composition (via Theorem 5.5.1)}

$$= ((\neg P^f \ ;_{\mathcal{A}} \mathit{true}) \land (\neg P^t \ ;_{\mathcal{A}} Q^f) \land (\neg P^t \ ;_{\mathcal{A}} \mathit{false}) \vdash P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{D}\mathbf{ac}} (\neg R^f \vdash R^t)$$

{Definition of sequential composition (via Theorem 5.5.1)}

$$= \left(\begin{array}{l} ((\neg P^f \ ;_{\mathcal{A}} \mathit{true}) \land (\neg P^t \ ;_{\mathcal{A}} Q^f) \land (\neg P^t \ ;_{\mathcal{A}} \mathit{false})) \ ;_{\mathcal{A}} \mathit{true} \\ \land (\neg (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} R^f) \land (\neg (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} \mathit{false}) \\ \vdash (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} R^t \end{array}\right)$$

{Right-distributivity of sequential composition (Law E.4.1)}

$$= \left(\begin{array}{l} (\neg P^f \ ;_{\mathcal{A}} \mathit{true}) \ ;_{\mathcal{A}} \mathit{true} \land (\neg P^t \ ;_{\mathcal{A}} Q^f) \ ;_{\mathcal{A}} \mathit{true} \land (\neg P^t \ ;_{\mathcal{A}} \mathit{false}) \ ;_{\mathcal{A}} \mathit{true} \\ \land (\neg (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} R^f) \land (\neg (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} \mathit{false}) \\ \vdash (P^t \ ;_{\mathcal{A}} Q^t) \ ;_{\mathcal{A}} R^t \end{array}\right)$$

{Negation (Law E.1.3)}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} true) \;;_{\mathcal{A}} true \wedge \neg ((P^t \;;_{\mathcal{A}} Q^f) \;;_{\mathcal{A}} true) \wedge \neg ((P^t \;;_{\mathcal{A}} false) \;;_{\mathcal{A}} true) \\ \wedge \neg ((P^t \;;_{\mathcal{A}} Q^t) \;;_{\mathcal{A}} R^f) \wedge \neg ((P^t \;;_{\mathcal{A}} Q^t) \;;_{\mathcal{A}} false) \\ \vdash (P^t \;;_{\mathcal{A}} Q^t) \;;_{\mathcal{A}} R^t \end{pmatrix}$$

{Associativity (Law E.1.2)}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} (true \;;_{\mathcal{A}} true)) \wedge \neg (P^t \;;_{\mathcal{A}} (Q^f \;;_{\mathcal{A}} true)) \wedge \neg (P^t \;;_{\mathcal{A}} (false \;;_{\mathcal{A}} true)) \\ \wedge \neg (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^f)) \wedge \neg (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} false)) \\ \vdash (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^t)) \end{pmatrix}$$

{Property of $;_{\mathcal{A}}$}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} (Q^f \;;_{\mathcal{A}} true)) \wedge \neg (P^t \;;_{\mathcal{A}} (false \;;_{\mathcal{A}} true)) \\ \wedge \neg (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^f)) \wedge \neg (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} false)) \\ \vdash (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^t)) \end{pmatrix}$$

{Propositional calculus}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} false) \\ \wedge \neg \big((P^t \;;_{\mathcal{A}} (Q^f \;;_{\mathcal{A}} true)) \vee (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^f)) \vee (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} false))\big) \\ \vdash (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^t)) \end{pmatrix}$$

{Distributivity of sequential composition over disjunction (Law **??**)}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} false) \\ \wedge \neg \big(P^t \;;_{\mathcal{A}} ((Q^f \;;_{\mathcal{A}} true) \vee (Q^t \;;_{\mathcal{A}} R^f) \vee (Q^t \;;_{\mathcal{A}} false))\big) \\ \vdash (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^t)) \end{pmatrix}$$

{Propositional calculus}

$$= \begin{pmatrix} (\neg P^f \;;_{\mathcal{A}} true) \wedge \neg (P^t \;;_{\mathcal{A}} false) \\ \wedge \neg \big(P^t \;;_{\mathcal{A}} ((\neg Q^f \;;_{\mathcal{A}} true) \wedge (\neg Q^t \;;_{\mathcal{A}} R^f) \wedge (\neg Q^t \;;_{\mathcal{A}} false))\big) \\ \vdash (P^t \;;_{\mathcal{A}} (Q^t \;;_{\mathcal{A}} R^t)) \end{pmatrix}$$

{Definition of $;_{\mathcal{D}\mathbf{ac}}$}

$$= (\neg P^f \vdash P^t) \;;_{\mathcal{D}\mathbf{ac}} (P^t \;;_{\mathcal{A}} ((\neg Q^f \;;_{\mathcal{A}} true) \wedge (\neg Q^t \;;_{\mathcal{A}} R^f) \wedge (\neg Q^t \;;_{\mathcal{A}} false)) \vdash Q^t \;;_{\mathcal{A}} R^t)$$

{Definition of $;_{\mathcal{D}\mathbf{ac}}$}

$$= (\neg P^f \vdash P^t) \;;_{\mathcal{D}\mathbf{ac}} ((\neg Q^f \vdash Q^t) \;;_{\mathcal{D}\mathbf{ac}} (\neg R^f \vdash R^t))$$

{Property of designs}

$$= P \;;_{\mathcal{D}\mathbf{ac}} (Q \;;_{\mathcal{D}\mathbf{ac}} R)$$

$\square$

Appendix G

Linking theories

G.1 *acdash2ac* and *ac2acdash*

Law G.1.1 (*acdash2ac*-**subset)**

$$acdash2ac(t) \subseteq acdash2ac(z) \Leftrightarrow t \subseteq z$$

Proof. To be established. $\square$

Law G.1.2 (*ac2acdash*-**subset)**

$$ac2acdash(t) \subseteq ac2acdash(z) \Leftrightarrow t \subseteq z$$

Proof. To be established. $\square$

Law G.1.3 (*acdash2ac*-$\emptyset$**)**

$$acdash2ac(\emptyset) = \emptyset$$

Proof.

$$\begin{aligned}
& acdash2ac(\emptyset) && \{\text{Definition of } acdash2ac\} \\
& = \left\{ \begin{array}{l} z_0 : S_{in\alpha P}, z_1 : S_{out\alpha P} \\ \mid z_0 \in \emptyset \land (\bigwedge x : \alpha P \bullet z_0.x = z_1.(x')) \bullet z_1 \end{array} \right\} && \{\text{Property of sets}\} \\
& = \{z_0 : S_{in\alpha P}, z_1 : S_{out\alpha P} \mid false\} && \{\text{Property of sets}\} \\
& = \emptyset
\end{aligned}$$

$\square$

Law G.1.4 ($ac2acdash$-$\emptyset$)

$$ac2acdash(\emptyset) = \emptyset$$

Proof. Similar to that of Law G.1.3 □

G.2 PBMH

Law G.2.1 *Provided P satisfies* **PBMH**.

$$P[\emptyset/ac'] \lor P = P$$

Proof.

$$\begin{array}{ll}
P[\emptyset/ac'] \lor P & \{\text{Assumption: } P \text{ is } \textbf{PBMH}\text{-healthy}\} \\
= (P \mathbin{;} ac \subseteq ac')[\emptyset/ac'] \lor (P \mathbin{;} ac \subseteq ac') & \{\text{Substitution}\} \\
= (P \mathbin{;} ac \subseteq \emptyset) \lor (P \mathbin{;} ac \subseteq ac') & \\
& \{\text{Distributivity of sequential composition w.r.t. disjunction}\} \\
= P \mathbin{;} (ac \subseteq \emptyset \lor ac \subseteq ac') & \{\text{Property of subset inclusion}\} \\
= P \mathbin{;} (ac \subseteq ac') & \{\text{Assumption: } P \text{ is } \textbf{PBMH}\text{-healthy}\} \\
= P &
\end{array}$$

□

G.3 Lemmas

Lemma G.3.1

$$pbmh2d(P) = \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \end{array}\right)$$

Proof.

$$\begin{array}{ll}
pbmh2d(P) & \{\text{Definition of } pbmh2d\} \\
= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \end{array}\right) & \\
& \{\text{Predicate calculus}\}
\end{array}$$

$$= \left(\begin{array}{l} \neg\, P[\emptyset / ac'][s / in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \\ \wedge\, (ac' = \emptyset \vee ac' \neq \emptyset) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \neg\, P[\emptyset / ac'][s / in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' \neq \emptyset) \\ \vee \\ (\exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' = \emptyset) \end{array} \right) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \neg\, P[\emptyset / ac'][s / in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' \neq \emptyset) \\ \vee \\ \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge \\ \forall\, z \bullet (z \in acdash2ac(ac_0) \Rightarrow z \in ac') \wedge \forall\, z \bullet z \notin ac' \end{array} \right) \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} \neg\, P[\emptyset / ac'][s / in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' \neq \emptyset) \\ \vee \\ \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge \\ \forall\, z \bullet (z \notin acdash2ac(ac_0) \wedge z \notin ac') \end{array} \right) \end{array} \right) \end{array} \right)$$

{Property of sets}

$$= \left(\begin{array}{l} \neg\, P[\emptyset / ac'][s / in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge acdash2ac(ac_0) \subseteq ac' \wedge ac' \neq \emptyset) \\ \vee \\ \left(\begin{array}{l} \exists\, ac_0 \bullet P[ac_0 / ac'][s / in\alpha] \wedge \\ ac2acdash \circ acdash2ac(ac_0) = ac2acdash(\emptyset) \wedge ac' = \emptyset \end{array} \right) \end{array} \right) \end{array} \right)$$

{Law G.1.4 and Law 5.7.1}

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset) \\ \lor \\ (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land ac_0 = \emptyset \land ac' = \emptyset) \end{array} \right) \end{array} \right)$$

{One-point rule}

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset) \\ \lor \\ (P[\emptyset/ac'][s/in\alpha] \land ac' = \emptyset) \end{array} \right) \end{array} \right)$$

{Definition of design}

$$= \left(\begin{array}{l} (ok \land \neg\, P[\emptyset/ac'][s/in\alpha]) \\ \Rightarrow \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset \land ok') \\ \lor \\ (P[\emptyset/ac'][s/in\alpha] \land ac' = \emptyset \land ok') \end{array} \right) \end{array} \right)$$

{Predicate calculus}

$$= \left(\begin{array}{l} ok \Rightarrow \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset \land ok') \\ \lor \\ (P[\emptyset/ac'][s/in\alpha] \land ac' = \emptyset \land ok') \\ \lor \\ P[\emptyset/ac'][s/in\alpha] \end{array} \right) \end{array} \right)$$

{Predicate calculus: absorption law}

$$= \left(\begin{array}{l} ok \Rightarrow \\ \left(\begin{array}{l} (\exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset \land ok') \\ \lor \\ P[\emptyset/ac'][s/in\alpha] \end{array} \right) \end{array} \right)$$

{Predicate calculus and definition of design}

$$= \left(\begin{array}{l} \neg\, P[\emptyset/ac'][s/in\alpha] \\ \vdash \\ \exists\, ac_0 \bullet P[ac_0/ac'][s/in\alpha] \land acdash2ac(ac_0) \subseteq ac' \land ac' \neq \emptyset \end{array} \right)$$

$\square$