A Stepwise Approach to Linking Theories

Pedro Ribeiro, Ana Cavalcanti, and Jim Woodcock

Department of Computer Science, University of York, UK
{pedro.ribeiro,ana.cavalcanti,jim.woodcock}@york.ac.uk

Abstract Formal modelling of complex systems requires catering for a variety of aspects. The Unifying Theories of Programming (UTP) distinguishes itself as a semantic framework that promotes unification of results across different modelling paradigms via linking functions. The naive composition of theories, however, may yield unexpected or undesirable semantic models. Here, we propose a stepwise approach to linking theories where we deal separately with the definition of the relation between the variables in the different theories and the identification of healthiness conditions. We explore this approach by deriving healthiness conditions for *Circus Time* via calculation, based on the healthiness conditions of CSP and a small set of principles underlying the timed model.

Keywords: theory engineering, *Circus*, CSP, UTP

1 Introduction

Systems exhibit several aspects of interest, including, for instance, state, behaviour, concurrency, object-orientation, time, and others. Several modelling paradigms capture one or a few of these aspects. The UTP of Hoare and He [1] is distinctive as a relational semantic framework that supports unification of results across different paradigms. Individual models can be studied in isolation using different UTP theories, while their combinations can be studied by composing theories. Of central importance to composition of theories are: a standard notion of refinement across the theories, and the definition of pairs of monotonic linking functions between them, usually Galois connections.

For example, in the UTP, functional total correctness is characterised by the theory of designs, while reactive behaviour is captured using the theory of reactive processes. Their combination yields a theory for the process algebra Communicating Sequential Processes (CSP) [2]. Additions to that theory yield theories in the *Circus* [3] family, where not only can state and behaviour be captured together, but also time [4,5], object-orientation [6], and so on [7].

Combining paradigms is not trivial as their naive combination may produce unexpected or undesirable semantic models. For example, it is often desirable for the operators of the combined theory to preserve the semantics of the corresponding operators of the original theories, in the sense that, when they are applied to predicates that correspond to those of the original theory, their behaviours are also in correspondence. To establish such a result, we need to identify Galois connections between the original and the combined theories.

We consider, for example, the theory of *Circus Time* [4], a discrete-time version of *Circus* that combines Z [8] and Timed CSP [9]. In *Circus Time*, data operations are instantaneous, and so every time property is explicitly specified: this is crucial to facilitate modelling and reasoning. It is not clear how to establish that the *Circus Time* theory preserves the semantics of the CSP operators, so that, when *Circus Time* operators are applied to (untimed) CSP processes, the resulting behaviour is consistent with that of the corresponding CSP operators.

Identifying a Galois connection that supports the proof that the operators in the *Circus Time* and CSP theories are consistent with each other is important, for example, to study external choice. The current definition [4] is not satisfactory: as pointed out in [10], external choice in *Circus Time* does not handle termination appropriately. We consider, for instance, $Wait\ d \Box Wait\ (d{+}m)$, a choice between terminating after d or after $d + m$ time units. Since, like in CSP, termination is not under the control of the environment, the choice should be resolved in favour of $Wait\ d$. However, this is not the case with the definition proposed in [4]. Finding an appropriate definition is challenging [11,12].

A Galois connection (L,R) is defined in [4]. L maps *Circus Time* processes to untimed *Circus* processes, while R is defined as the weakest inverse of L. For example, the application of L to $Wait\ d$ yields $Skip \sqcap Stop$, a process that may choose nondeterministically to terminate or deadlock. The results obtained for operators mapped through this linking function are not satisfactory. It is not clear how *Skip* can be mapped into its *Circus Time* counterpart as a terminating process taking no time, at the same time that the timed counterpart of *Stop* takes any amount of time. These desirable properties of the timed model make it less than obvious how to define an appropriate Galois connection.

In this paper, we present a general stepwise approach to linking theories that, by providing for a clear separation of concerns when linking theories, gives guidance as to how theories can be linked. We take inspiration from the calculational approach to data refinement based on auxiliary variables [13]. Accordingly, we use an intermediate super-theory with variables of both theories of interest.

In our approach, the link between the source and the super-theory adds the variables of the target theory. Another important component of a UTP theory are healthiness conditions, which identify the valid predicates over the theory variables. In our approach, healthiness conditions that the desired target theory must satisfy and coupling invariants relating variables of both theories are used to characterise the super-theory. The target theory is reached by removing the starting theory's variables. The opposite links can be constructed similarly.

We have applied our approach to *Circus Time* to construct a Galois connection that can justify its healthiness conditions and operators. In this example, we split the healthiness conditions in two categories: those that refer exclusively to concerns of the timed model are identified separately from those carried over from (untimed) CSP. The healthiness conditions of the original *Circus Time* theory are explained as combinations of these. We also justify the relationship between the observation variables of the two theories by considering separately the removal and introduction of variables, and the relationship between variables

in the different theories. Our super-theory allows us to derive the healthiness conditions and operators of *Circus Time* as induced from the untimed model.

The remainder of this paper is organised as follows. In Section 2 we introduce the required UTP theories, including the theory of CSP and *Circus Time*. In Section 3 we discuss the stepwise linking approach. In Section 4 we use the proposed approach to build a super-theory of timed reactive processes, and ultimately derive a model for *Circus Time*. Finally, we conclude in Section 5 by summarizing our findings and discussing future work.

2 Preliminaries

UTP theories include relations defined by predicates P. They are characterised by three components: an alphabet, a set of healthiness conditions and a set of operators. The alphabet defines the free variables that can be used in the predicates. Also, the alphabet $\alpha(P)$ of a relation P is split into $in\alpha(P)$, which contains undashed variables corresponding to the initial observations, and $out\alpha(P)$ containing the dashed counterparts for after or final observations. The healthiness conditions are defined by monotonic idempotent functions; the theory contains only the healthy predicates: the fixed points of the healthiness conditions. The predicates can be defined using the operators of the theory.

Refinement is defined in all theories as universal reverse implication. In the UTP, total correctness is characterised through the theory of designs [1,14], whose healthiness conditions are named **H1** and **H2**. Every design P can be expressed in terms of pre and postcondition pairs, $(\neg P^f \vdash P^t)$, where $P^o = P[o/ok']$ and t and f correspond to *true* and *false*, respectively.

2.1 CSP

Programs characterised by continuous interactions with their environment are modelled in the UTP using the theory of reactive processes [1,15]. In addition to the variables, ok and ok' of the theory of designs, this theory includes the variables $wait$, tr, ref and their dashed counterparts, that record information about interactions with the environment.

This is a theory where observations of intermediate states of programs are recorded. The boolean variable $wait$ records whether the previous process is waiting for an interaction from the environment or, alternatively, has terminated. Similarly, $wait'$ ascertains this for the current process. The boolean variable ok indicates whether the previous process is in a stable state, while ok' records this information for the current process. If a process is not in a stable state, it is said to have diverged. A process starts executing only in states where ok and $\neg wait$ are *true*. Successful termination is characterised by ok' and $\neg wait'$ being *true*.

The actual interactions with the environment are represented using sequences of events, recorded by tr and tr'. The variable tr records the sequence of events that took place before the current process started, while tr' records the intermediate or final sequence of events that can be observed. Finally, ref and ref'

record the set of events that may be refused by the process. Refusal sets allow the appropriate modelling of deadlock and nondeterminism [2].

The theory of reactive processes **R** is characterised by the functional composition ($\circ$) of three healthiness conditions [1,15] below, where function application binds stronger than function composition.

Definition 1 (Healthiness Conditions of Reactive Processes).

$$\mathbf{R1}(P) \mathrel{\widehat{=}} P \wedge tr \leq tr' \qquad \mathbf{R2}(P) \mathrel{\widehat{=}} P[\langle\rangle, (tr'-tr)/tr, tr']$$
$$\mathbf{R3}(P) \mathrel{\widehat{=}} \mathbb{I}_{rea} \lhd wait \rhd P \qquad \mathbf{R}(P) \mathrel{\widehat{=}} \mathbf{R3} \circ \mathbf{R1} \circ \mathbf{R2}(P)$$

R1 requires that in all circumstances the only change that can be observed in the final trace of events tr' is an extension of the initial sequence tr, while **R2** requires that a process must not impose any restriction on the initial value of tr. Finally, **R3** requires that if the previous process is waiting for an interaction with the environment, that is, *wait* is *true*, then the process behaves as the identity of the theory $\mathbb{I}_{rea}$ [1,15].

The theory of CSP can be described by reactive processes that in addition satisfy the healthiness conditions **CSP1** and **CSP2** reproduced below [1,15].

Definition 2 (CSP).

$$\mathbf{CSP1}(P) \mathrel{\widehat{=}} P \vee \mathbf{R1}(\neg ok)$$
$$\mathbf{CSP2}(P) \mathrel{\widehat{=}} P \ ; ((ok \Rightarrow ok') \wedge tr' = tr \wedge ref' = ref \wedge wait' = wait)$$

The first healthiness condition **CSP1** requires that if the previous process has diverged, that is, *ok* is *false*, then extension of the trace is the only guarantee. **CSP2** is **H2** restated with the extended alphabet of reactive processes.

A process that is **R**, **CSP1** and **CSP2**-healthy can be described in terms of a design [1,15]. We reproduce this result below, where $P^o_w = P[o, w/ok', wait]$.

Theorem 1 (Reactive Design). *For every CSP process* P, $\mathbf{R}(\neg P^f_f \vdash P^t_f) = P$

This result is important as it allows CSP processes to be specified in terms of pre and postconditions, such as is the case for sequential programs, while the healthiness condition **R** enforces the required reactive behaviour.

2.2 *Circus Time*

Circus is a combination of Z, CSP and Dijkstra's language of guarded commands. Its semantics is also defined using reactive designs. The timed version *Circus Time* [4,5] provides facilities to explicitly model and reason about discrete time state-rich reactive systems. Observations are timed, so the trace of events and the set of refusals are recorded as pairs in a non-empty timed sequence tr_T, whose dashed counterpart is tr'_T, and where Σ is the set of all possible events. This is analogous to untimed CSP where tr and tr' are defined as sequences whose elements are drawn from Σ.

Definition 3. $tr_T, tr'_T : \mathrm{seq}_1(\mathrm{seq}\, \Sigma \times \mathbb{P}\, \Sigma)$

Here we use seq_1 following the Z notation [16] to denote a finite non-empty sequence. The variables *ok*, ok', *wait* and $wait'$ retain the same meaning as in

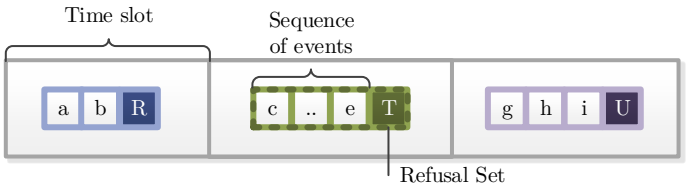

Figure 1: A timed sequence.

the untimed theory, and in that of CSP. For the purpose of our discussion, we adopt a model based on that of [4], but without considering state directly.

An illustration of a timed sequence consisting of three time slots is presented in Figure 1. Each slot contains a pair, whose first component is a sequence of events, such as a followed by b, and whose second component is a refusal set (shaded in Figure 1) such as R. This is useful to illustrate the intuition behind the healthiness conditions that we discuss in the sequel.

Healthiness Conditions The first healthiness condition $\mathbf{R1_T}$ of the *Circus Time* theory ensures that the trace of events across time cannot be undone. It is the counterpart to **R1** and is defined as follows.

Definition 4. $\mathbf{R1_T}(P) \mathrel{\widehat{=}} P \wedge \mathcal{E}(tr_T, tr'_T)$

It is a conjunctive healthiness condition [7] defined using the predicate $\mathcal{E}$.

Definition 5. $\mathcal{E}(s,t) \mathrel{\widehat{=}} (front(s) < t) \wedge fst \circ last(s) \leq fst \circ head(t - front(s))$

Given two timed traces s and t, $\mathcal{E}$ requires the *front* (which for a given sequence yields all the elements except the last) of s to be a strict prefix of t, and in addition that the first component (as given by *fst*) of the *last* pair of s is a prefix of the first component of the *head* of the difference between t and $front(s)$. If we consider s and t to be tr_T and tr'_T, respectively, then the strict prefixing $front(tr_T) < tr'_T$ requires that not only are the traces of previous time slots kept unchanged, but also the refusal sets. In addition, the difference $tr'_T - front(tr_T)$ yields the timed sequence corresponding to the current and future observations, and so the *head* corresponds to the first after observation in the current time slot. A pair of sequences satisfying $\mathbf{R1_T}$ is illustrated in Figure 2. The functions *front*, *last*, *head*, *fst* and *snd* are those of Z [16] with expected meanings.

The counterpart to **R2** is $\mathbf{R2_T}$, which requires processes to be insensitive to events in the initial timed sequence tr_T.

Definition 6. $\mathbf{R2_T}(P) \mathrel{\widehat{=}} P[\langle(\langle\rangle, snd \circ last(tr_T))\rangle, dif_T(tr'_T, tr_T)/tr_T, tr'_T]$

It is defined by considering the substitution of tr_T by the timed sequence whose only element is a pair, where the trace is empty and the refusal set is the last observed in tr_T. The sequence tr'_T is substituted by the application of the function dif_T that captures the difference in events during the current time slot.

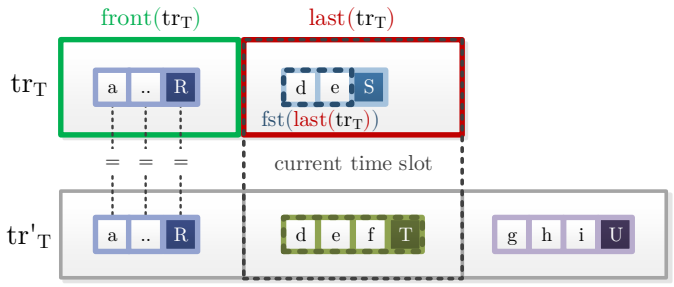

Figure 2: Example of a pair of sequences tr_T and tr'_T satisfying $\mathbf{R1_T}$.

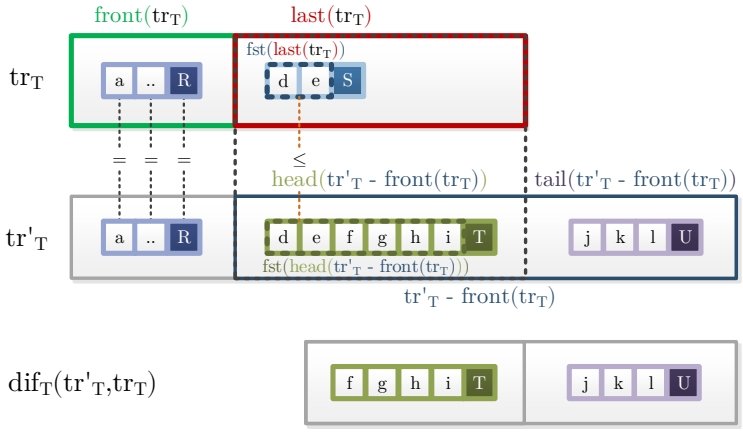

Figure 3: Example application of dif_T.

The function dif_T takes two timed traces tr'_T and tr_T, and yields a sequence whose first element is a pair containing the trace actually observed during that time slot, and the refusal set observed at the end of the time slot.

Definition 7.

$$dif_T(tr'_T, tr_T) \mathrel{\widehat{=}} \left(\begin{array}{l} \langle \left(\begin{array}{l} fst \circ head(tr'_T - front(tr_T)) - fst \circ last(tr_T)), \\ snd \circ head(tr'_T - front(tr_T)) \end{array} \right) \rangle \\ \frown \\ tail(tr'_T - front(tr_T)) \end{array} \right)$$

The current sequence of time slots is obtained by the difference $tr'_T - front(tr_T)$. The actual events occurring during the first of those slots are obtained by the difference between $fst \circ head(tr'_T - front(tr_T))$ and $fst \circ last(tr_T)$. An illustration of an application of dif_T to timed traces satisfying $\mathbf{R1_T}$ is shown in Figure 3.

The counterpart to **R3** is $\mathbf{R3_T}$ below. Instead of $\mathbb{I}_{rea}$, the identity of the theory of reactive processes, $\mathbb{I}_T$, the identify of the timed theory is employed.

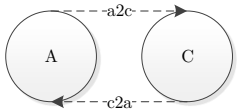

Figure 4: Linking between theories.

Definition 8.

$$\mathbf{R3_T}(P) \mathrel{\widehat{=}} \mathbb{I}_T \lhd wait \rhd P$$
$$\mathbb{I}_T \mathrel{\widehat{=}} \mathbf{R1_T}(\neg ok) \lor (ok' \land tr'_T = tr_T \land wait' = wait)$$

If the process is in an unstable state, that is, ok is *false*, then expansion of the timed sequence tr_T is the only guarantee. Otherwise, the process is stable, that is, ok is *true*, the timed sequence tr_T is kept intact and so is the value of *wait*. The functional composition of $\mathbf{R1_T}$, $\mathbf{R2_T}$ and $\mathbf{R3_T}$ is $\mathbf{R_T}$.

This concludes the overview of *Circus Time*. We next explore an approach to find Galois connections between theories, which leads to the definition of a new Galois connection between *Circus* and *Circus Time*.

3 Linking Theories via Super-Theories

The definition of linking functions between UTP theories with different alphabets involves introduction of variables of the target theory and removal of variables of the source theory (essentially a data refinement), while at the same time enforcing the healthiness conditions of the target theory. In other words, in addition to a data refinement, there is an application of the healthiness condition of the target theory. This is illustrated for two arbitrary theories A and C in Figure 4, where a pair of linking functions $a2c$ and $c2a$ is shown.

When defining $a2c$ and $c2a$, a problem arises if the complete set of healthiness conditions of the target theory C is not known a priori. This is often the case when developing a new theory. An appealing approach is to calculate the healthiness conditions via application of $a2c$ to healthy predicates of A. If, however, finding a Galois connection, that is, defining $a2c$ and $c2a$ in the first place, is not immediately obvious, then this is not a solution.

For example, in the case of the link from *Circus* to *Circus Time* two choices arise naturally: every trace of events takes place in a single time slot, and so no time is actually added; or any amount of time can pass for any given trace. The latter violates $\mathbf{R1_T}$, while the former does not capture an interesting correspondence between the models. The right approach lies between these two extremes.

We propose that, instead of exploring the links between the theories directly, we break down the linking functions into a series of functions that, when composed, achieve the same goal. We consider again the arbitrary theories depicted

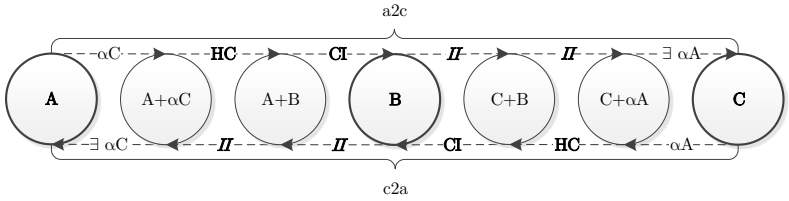

Figure 5: Stepwise linking between theories.

in Figure 4, and suppose that we know only partially the set of healthiness conditions of the theory C, denoted by the function **HC**. To calculate those induced from theory A, we can proceed as depicted in Figure 5.

The theories A and C are related through an intermediate super-theory B. The alphabet of B is the union of the alphabets of A and C: $\alpha B = \alpha A \cup \alpha C$. To relate the values of variables in αA and αC we introduce a coupling invariant **CI**, which is applied after **HC**, the known healthiness condition of the theory C that must be satisfied irrespective of those induced from A.

In what follows we define coupling invariants and characterise the properties required of **HC** to ensure that $a2c$ and $c2a$ form a Galois connection between the theories of interest. Finally, we present formal definitions for $a2c$ and $c2a$, and show that they form a Galois connection.

A coupling invariant is a monotonic and idempotent function **CI** defined by the general form below, where Q is a predicate relating variables.

Definition 9 (Coupling Invariant). $\mathbf{CI}(P) \mathrel{\widehat{=}} P \wedge Q$

If Q does not depend on P, then **CI** is a conjunctive healthiness condition [7]. A coupling invariant and the identity function $\mathbb{I}$ form a Galois connection as established by the following Lemma 1, following the result of Lemma 4.2.3 in [1].

Lemma 1. CI *and* $\mathbb{I}$ *form a Galois connection in the domain of* **CI***-healthy predicates.*

Proof. $\mathbb{I} \circ \mathbf{CI}(P) \sqsupseteq P$ {By definition of $\mathbb{I}$ and **CI** and predicate calculus}
and $\mathbf{CI} \circ \mathbb{I}(Q) \sqsubseteq Q$ {By definition of $\mathbb{I}$ and predicate calculus, Q satisfies **CI**}
□

We observe that in the proof of Lemma 1, we assume that $\mathbb{I}$ is applied to a **CI**-healthy predicate. This is because the Galois connection is established with the subset of interest of theory B that is **CI**-healthy.

Similarly, links related to the data refinement, in which one function introduces variables, and another function hides them, also form a Galois connection as established by the following Lemma 2. We use the operator $_{+C}$, an alphabet extension with no particular value specified for variables in the set C.

Definition 10. $P_{+C} \mathrel{\widehat{=}} P$, *with* $\alpha(P_{+C}) = \alpha(P) \cup C$

In words, the alphabet of P_{+C} is augmented with the variables in C, but the

values of these new variables are not restricted.

Lemma 2. *Provided variables in αC are not free in P, $\exists\, \alpha C \bullet P$ and Q_{+C} form a Galois connection.*

Proof.

$$
\begin{array}{ll}
\exists\, \alpha C \bullet (P_{+C}) & \{\text{Theory alphabet extension}\} \\
= \exists\, \alpha C \bullet (P) & \{\text{Assumption: } c \text{ and } c' \text{ not free in } P\} \\
= P &
\end{array}
$$

$$
\begin{array}{ll}
(\exists\, \alpha C \bullet Q)_{+C} & \{\text{Theory alphabet extension}\} \\
= (\exists\, \alpha C \bullet Q) & \{\text{Predicate calculus}\} \\
\sqsubseteq Q &
\end{array}
$$

□

The remaining Galois connection to be established lies between the theory with variables of both A and B (depicted as $A + \alpha C$ in Figure 5), and the theory whose predicates satisfy **HC** (depicted as $A + B$ in Figure 5).

Lemma 3. **HC** *and* $\mathbb{I}$ *form a Galois connection in the domain of* **HC***-healthy predicates, provided* **HC** *is a monotonic and idempotent function, and, for all P, either* $\mathbf{HC}(P) \sqsupseteq P$ *(strengthening) or* $\mathbf{HC}(P) \sqsubseteq P$ *(weakening), or both.*

Proof. $\mathbf{HC} \circ \mathbb{I}(Q) = Q$ {Def. of $\mathbb{I}$, and assumption: Q is **HC**-healthy}
(Case: **HC** *is strengthening)*

$$
\begin{array}{ll}
\mathbb{I} \circ \mathbf{HC}(P) & \{\text{Definition of } \mathbb{I}\} \\
= \mathbf{HC}(P) & \{\text{Assumption: } \mathbf{HC}(P) \sqsupseteq P\} \\
\sqsupseteq P &
\end{array}
$$

(Case: **HC** *is weakening)*

$$
\begin{array}{ll}
\mathbb{I} \circ \mathbf{HC}(P) & \{\text{Definition of } \mathbb{I}\} \\
= \mathbf{HC}(P) & \{\text{Assumption: } \mathbf{HC}(P) \sqsubseteq P\} \\
\sqsubseteq P &
\end{array}
$$

□

When **HC** is applied to a predicate P that is **HC**-healthy, then $\mathbf{HC}(P) = P$, and the proviso of Lemma 3 requiring strengthening or weakening is trivially satisfied. In the context of our approach, however, the proviso must also be satisfied when **HC** is applied to a predicate P that results from the application of P_{+C}, that is, when the variables of set C are allowed to take arbitrary values by P. For example, we consider the case where **HC** is defined by a function like **R2**. This function is neither strengthening nor weakening when applied to unhealthy-predicates. We consider the following counter-example: $\mathbf{R2}(tr = \langle\rangle) = \mathit{true}$ and $\mathbf{R2}(tr \neq \langle\rangle) = \mathit{false}$. However, the application of $\mathbf{R2}(P)$ to a predicate P where tr and tr' take arbitrary values yields an equality.

As illustrated in Figure 5 the linking function $a2c$ from A to C is the composition of several functions: a function that introduces the variables of C; the healthiness condition **HC**; the coupling invariant **CI**; followed by two applications of the identity function, and an existential quantification over all variables in A. We have a similar composition for $c2a$. Formally, we can describe $a2c$ and $c2a$ as follows: the identities do not need to be included in the composition.

Definition 11.

$$a2c(P) \mathrel{\widehat{=}} \exists\, \alpha A \bullet \mathbf{CI} \circ \mathbf{HC}(P_{+C}) \qquad c2a(P) \mathrel{\widehat{=}} \exists\, \alpha C \bullet \mathbf{CI} \circ \mathbf{HC}(P_{+A})$$

For variables that are simply aliases, the existential quantification at either end of the link is over the subset of those variables not present in the target theory. The relation established between variables could alternatively be defined using the data refinement approach of the UTP. However, to satisfy **HC** the invariants would need to be strengthened, and it is not clear how functions like **R2** could be justified purely by data refinement. Here we deal with these concerns piecewise.

The functions $a2c$ and $c2a$ form a Galois connection. This is our main result in this paper, established by the following Theorem 2.

Theorem 2. *$a2c$ and $c2a$ form a Galois connection, provided* **HC** *is idempotent and monotonic, and* **HC** *is either strengthening or weakening, or both.*

Proof. Follows from Lemmas 1 to 3 and Theorem 4.2.5 in [1] (Galois connections compose).

Our approach provides for a systematic way of studying the relationship between theories. As long as the known healthiness condition **HC** is weakening or strengthening, or both, then a Galois connection can be established. The coupling invariant can be tweaked as required to yield different Galois connections. Links between theories can be non-trivial due to the underlying differences in paradigm. The intermediate super-theory enables constructs from multiple theories to be considered together within the same alphabetized relation space, while still providing a Galois connection with the constituent theories.

This concludes our discussion on building super-theories. In the next section, we illustrate our approach by discussing how it can be used to build a model for *Circus Time* starting with the (untimed) CSP theory.

4 A Stepwise Approach Towards *Circus Time*

Here, we build a super-theory of timed reactive processes based on the CSP theory. Section 4.1 defines the alphabet and healthiness conditions of the super-theory, and coupling invariants that characterise the valid timed traces. The instance of **HC** in this example is the composition of the healthiness conditions we identify; similarly, the instance of **CI** is the composition of the coupling invariants. Defining **HC** and **CI** by composition of simpler functions gives a piecewise characterisation of properties of interest. This method is suggested by

Theorem 2 and is an illustration the main feature of our stepwise approach to connecting theories. In Section 4.2 we calculate an explicit description of the linking function from (untimed) CSP to *Circus Time*, using Definition 11, and present the results obtained with our Galois connection.

4.1 Constructing the Super-Theory

The alphabet of the super-theory includes the union of alphabets of the theories of CSP and *Circus Time*, defined in Section 2; ok and ok' are common to both theories. Furthermore, we also add auxiliary variables tr_C and tr'_C to the super-theory to facilitate reasoning about traces in the current time slot.

Definition 12 (Alphabet).

$$tr, tr' : \operatorname{seq} \Sigma \qquad tr_T, tr'_T : \operatorname{seq}(\operatorname{seq} \Sigma \times \mathbb{P}\, \Sigma)$$
$$wait, wait', ok, ok' : Boolean \qquad wait_T, wait'_T : Boolean$$
$$ref, ref' : \mathbb{P}\, \Sigma \qquad tr_C, tr'_C : \operatorname{seq} \Sigma$$

In contrast with the treatment in [4], we require timed traces not to be empty by using a healthiness condition, defined in the sequel, rather than using the type system directly. This obviates the need to check intermediate calculations for type correctness with regards to this property.

Healthiness Conditions Here, we identify minimal restrictions that are later used to justify the original healthiness conditions of *Circus Time*. With this approach, we consider issues related to time in isolation from those already captured by the healthiness condition of CSP.

TR0 The first condition of the super-theory requires that no sequence of events is empty: the length $\# tr_T$ of the initial trace tr_T is greater than zero.

Definition 13. $\mathbf{TR0}(P) \mathrel{\widehat{=}} P \land \# tr_T > 0$

This makes operations on traces, such as *front*, well-defined. The corresponding restriction on tr'_T arises as a consequence of **TR0** and **TR1** defined next.

TR1 The second healthiness condition requires that time increases monotonically, that is, the length of the after timed trace tr'_T must be greater than or equal to the length of the current timed trace tr_T.

Definition 14. $\mathbf{TR1}(P) \mathrel{\widehat{=}} P \land \# tr_T \leq \# tr'_T$

In the original *Circus Time* model [4], **TR0** and **TR1** are implicit.

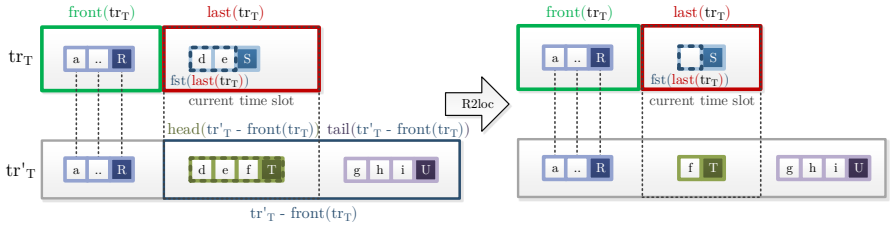

Figure 6: Example of applying $\mathbf{R2loc_T}$.

TR2 The third healthiness condition requires that previous observations across time cannot be changed and is defined as follows.

Definition 15. $\mathbf{TR2}(P) \mathrel{\widehat{=}} P \wedge \mathit{front}(tr_T) \leq tr'_T$

In words, the *front* of the current timed sequence tr_T must be a prefix of tr'_T. In [4], this requirement is part of $\mathbf{R1_T}$, but here it is studied in isolation.

An example of a relation that is **TR0**, **TR1** and **TR2**-healthy is depicted in Figure 2. The healthiness conditions considered so far guarantee preservation of history before the current time slot, however, they are not sufficient to guarantee that **R1** is observed within the current time slot. Later in this section, we tackle this aspect by using coupling invariants related to tr_C and tr'_C.

TR3 The next healthiness condition defines for $wait_T$ part of what is established by **R3** for *wait*. It states that if the previous process is waiting in a stable state, then no explicit time is added and it continues waiting.

Definition 16. $\mathbf{TR3}(P) \mathrel{\widehat{=}} P \wedge ((ok \wedge wait_T) \Rightarrow (\#tr'_T = \#tr_T \wedge wait'_T))$

This healthiness condition is essential to justify the definition of the timed identity $\mathbb{I}_T$. Further aspects of **R3**, including behaviour in the presence of divergence of the previous process are considered separately.

$\mathbf{R2loc_T}$ The following healthiness condition captures part of **R2**, in that, if we ignore time and the events that happened in the previous time slot, then the counterpart to applying **R2** in the current time slot is $\mathbf{R2loc_T}$.

Definition 17. $\mathbf{R2loc_T}(P) = P \begin{bmatrix} \mathit{front}(tr_T) \frown \langle(\langle\rangle, \mathit{snd} \circ \mathit{last}(tr_T))\rangle / tr_T \\ \mathit{front}(tr_T) \frown \mathit{dif}_T(tr'_T, tr_T) / tr'_T \end{bmatrix}$

A pictorial description of the application of $\mathbf{R2loc_T}$ is shown in Figure 6. In the current time slot, the *front* of tr_T is maintained, while the last sequence of events is replaced by the empty sequence. Similarly, the subsequent observation of tr'_T is replaced with *front* of tr_T ($\mathit{front}(tr_T)$ is guaranteed to be a prefix of tr'_T when we consider relations that satisfy **TR0** and **TR1**) followed by the corresponding difference in events observed during the current time slots as given by dif_T.

For dif_T to be well-defined $\mathit{fst}(\mathit{last}(tr_T))$ must be a prefix of the sequence $\mathit{fst}(\mathit{head}(tr'_T - \mathit{front}(tr_T)))$. This is not an issue in the original *Circus Time* theory

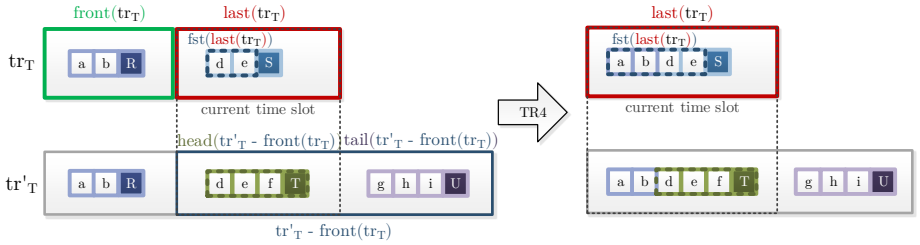

Figure 7: Example of applying **TR4**.

as it includes $\mathbf{R1_T}$, but the healthiness conditions above do not address this issue. One option is to consider this as a requirement for dif_T to be well-defined. Another option, which we choose to follow, is to enforce the counterpart to **R1** in the current time slot with the following healthiness condition.

Definition 18. $\mathbf{R1_C}(P) \mathrel{\widehat{=}} tr_C \leq tr'_C$

This is a modelling decision: both options can justify $\mathbf{R1_T}$. Later, a coupling invariant relates the values of tr_C and tr'_C with those of tr'_T and tr'_T.

TR4 The second requirement of $\mathbf{R2_T}$ is captured by the following healthiness condition **TR4** that requires processes not to depend on the time elapsed before them, irrespective of events that have happened. $\mathbf{R2loc_T}$ above captures insensitivity to events, whereas **TR4** captures insensitivity to time. A fixed point of **TR4** must allow the timed traces tr_T and tr'_T to be replaced with traces whose first time slots contain all events that have happened before, concatenated with any current events. In other words, the behaviour of a fixed point must be the same, even if no time had elapsed before.

Definition 19.

$$\mathbf{TR4}(P) \mathrel{\widehat{=}} P\left[\begin{array}{l} \langle (Flat(tr_T), snd \circ last(tr_T)) \rangle / tr_T \\ \left(\begin{array}{l} \langle \left(\begin{array}{l} Flat(front(tr_T) \frown head(tr'_T - front(tr_T))), \\ snd \circ head(tr'_T - front(tr_T)) \end{array}\right) \rangle \\ \frown \\ tail(tr'_T - front(tr_T)) \end{array}\right) / tr'_T \end{array}\right]$$

The sequence tr_T is replaced by a sequence with only one element: a pair whose first component is $Flat(tr_T)$, a projection on tr_T that yields the sequence of events in every first component of the pairs in tr_T, that is, all events that happened by the beginning of the current observation. Similarly, tr'_T is replaced by a sequence whose first pair has as first component the sequence of events observed up until the end of the current time slot. This includes the events in $front(tr_T)$ concatenated with those in the current time slot, given by $head(tr'_T - front(tr_T))$. An example of applying **TR4** is shown in Figure 7.

The combination of $\mathbf{R2loc_T}$ and **TR4** corresponds to $\mathbf{R2_T}$ as established by the following Lemma 4. Proof of this and other results to follow that are not included in this paper are available in [17], with essential results having been checked using Isabelle/UTP [18].

Lemma 4. *Provided* $\#tr_T > 0$, $\mathbf{TR4} \circ \mathbf{R2loc_T}(P) = \mathbf{R2_T}(P)$.

This equality with $\mathbf{R2_T}$ holds only when **TR4** is applied after $\mathbf{R2loc_T}$. Although this may seem counter-intuitive, this requirement is a consequence of the order in which the substitutions of **TR4** and $\mathbf{R2loc_T}$ are applied.

TR The healthiness condition corresponding to the functional composition of all the previous healthiness conditions is **TR**.

Definition 20.

$$\mathbf{TR}(P) \mathrel{\widehat{=}} \mathbf{TR0} \circ \mathbf{TR1} \circ \mathbf{TR2} \circ \mathbf{TR3} \circ \mathbf{TR4} \circ \mathbf{R2loc_T} \circ \mathbf{R1_C}(P)$$

This function is strengthening as established by the following Theorem 3.

Theorem 3. *Provided* tr_T *and* tr'_T *are not free in* P, $\mathbf{TR}(P) \sqsupseteq P$.

Proof.

$$\begin{array}{ll}
\mathbf{TR}(P) & \{\text{Definition of } \mathbf{TR}\} \\
= \mathbf{TR0123} \circ \mathbf{TR4} \circ \mathbf{R2loc_T} \circ \mathbf{R1_C}(P) & \{\text{Lemma 4}\} \\
= \mathbf{TR0123} \circ \mathbf{R2_T} \circ \mathbf{R1_C}(P) & \{\text{Assumption: } tr_T \text{ and } tr'_T \text{ not free in } P\} \\
= \mathbf{TR0123} \circ \mathbf{R1_C}(P) & \{\text{Definition of } \mathbf{TR0} \text{ to } \mathbf{TR3}\text{, predicate calculus}\} \\
\sqsupseteq \mathbf{R1_C}(P) & \{\text{Definition of } \mathbf{R1_C} \text{ and predicate calculus}\} \\
\sqsupseteq P &
\end{array}$$

Following the approach outlined in Section 3, this result ensures that a linking function including **TR** as healthiness condition, yields a Galois connection.

This concludes the discussion of the healthiness conditions governing the timed aspects of the super-theory and *Circus Time*.

Coupling Invariants In this section we define the coupling invariants that relate the value of variables in the super-theory.

CI0 The first coupling invariant relates the timed traces, tr_T and tr'_T, with their untimed counterparts, tr and tr'. The difference in traces in the untimed model $tr' - tr$ must be in agreement with the difference in events observed over all time units as given by the difference $Flat(tr'_T) - Flat(tr_T)$.

Definition 21.

$$\begin{array}{l}
\mathbf{CI0}(P) \mathrel{\widehat{=}} \\
\quad P \wedge (tr' - tr) = Flat(tr'_T) - Flat(tr_T) \wedge Flat(tr_T) \leq Flat(tr'_T) \wedge tr \leq tr'
\end{array}$$

For the differences to be well-defined we require $Flat(tr_T)$ to be a prefix of $Flat(tr'_T)$, and tr of tr'. While a direct equality could be used, rather than an equality between differences, it poses problems if **R2** were applied to **CI0** as it forbids insisting on a particular value for tr. Therefore, here we only consider the relationship between differences, an approach also followed in [19].

CI1 The second invariant requires refusals in the untimed *ref* variables and the timed traces variables tr_T to be in agreement.

Definition 22. $\mathbf{CI1}(P) \mathrel{\widehat=} P \land ref = snd \circ last(tr_T) \land ref' = snd \circ last(tr'_T)$

The value of *ref* must be the same as the refusal in the last time slot $last(tr_T)$ of tr_T, as given by the second component of $last(tr_T)$, whereas ref' must be the refusal in the last time slot $last(tr'_T)$, as given by the second component of $last(tr'_T)$, which may or may not be the same time slot as $last(tr_T)$.

CI2 The next invariant requires that termination without visible events in a stable state in the untimed model does not allow any time to pass.

Definition 23.

$$\mathbf{CI2}(P) \mathrel{\widehat=} P \land ((\neg wait' \land \neg P^f_f \land ok \land ok' \land tr' = tr) \Rightarrow \#tr'_T = \#tr_T)$$

That is, when $wait'$ is *false*, the precondition $\neg P^f_f$ of P is satisfied, and stability is preserved with ok and ok', and no event is observed $tr' = tr$, then no time must pass. Consequently, the CSP process *Skip* in the context of the super-theory allows no time to pass. As previously indicated, this is required in *Circus Time* to ensure that time passage is explicitly modelled. We note that data operations in *Circus Time*, like *Skip*, do not engage in events, and so, if not divergent, are instantaneous. So, time budgets and deadlines need to be explicitly defined.

CI3 The next invariant relates termination of interactions in both theories.

Definition 24. $\mathbf{CI3}(P) \mathrel{\widehat=} P \land wait_T = wait \land (\neg wait' \Rightarrow \neg wait'_T)$

It requires that termination, or not, of the previous process is the same in both models as $wait_T$ is equal to $wait$. On the other hand, termination of interactions in the untimed model, for the current process, implies termination in the timed model, but not vice-versa. If we were to admit $wait' = wait'_T$, then it would be impossible to define a process such as *Wait* d, since, when it terminates, **CI2** requires no time to pass, and thus d could never be greater than zero. On the negative side, if we consider the CSP process *Stop* in the context of the super-theory, then it does not necessarily wait forever in the timed model. This is, however, unavoidable: if we were to admit $wait' \Rightarrow wait'_T$, then in the context of the super-theory *Stop* would require non-termination appropriately, but *Skip* would no longer require termination, and similarly *Wait* d could still never terminate with d greater than zero due to **CI2**. We, therefore, need to provide a new definition of *Stop* in the super-theory, which is not related to the CSP process *Stop* by our Galois connection. It was the study of the super-theory, including both the *wait* and $wait_T$ variables, that revealed the difficulties.

CI4 The final coupling invariant **CI4** relates the values of tr_C and tr'_C, and the values of tr_T and tr'_T, respectively.

Definition 25.

$$\mathbf{CI4}(P) \mathrel{\widehat=} \begin{pmatrix} fst \circ head(tr'_T - front(tr_T)) - fst \circ last(tr_T) = tr'_C - tr_C \wedge \\ P \wedge \\ tr_C \leq tr'_C \wedge fst \circ last(tr_T) \leq fst \circ head(tr'_T - front(tr_T)) \end{pmatrix}$$

The difference in traces between the variables tr'_C and tr_C, and the difference in events observed in the timed traces during the current time slot, as given by $fst \circ head(tr'_T - front(tr_T))$ and $fst \circ last(tr_T)$, must be in agreement. Finally tr_C must be a prefix of tr'_C in order for the difference to be well-defined. Similarly, we also require the differences in the timed model to be prefixes. As discussed before, this aspect is part of $\mathbf{R1_T}$ in the original *Circus Time* theory.

CI The complete relationship between timed and untimed variables is established by the coupling invariant **CI**, the composition of the previous invariants.

Definition 26. $\mathbf{CI}(P) = \mathbf{CI0} \circ \mathbf{CI1} \circ \mathbf{CI3} \circ \mathbf{CI2} \circ \mathbf{CI4}(P)$

We observe that **CI3** needs to be applied before **CI2** as the functions are not commutative; the others commute with each other.

Having defined both the healthiness condition **TR** and the coupling invariant **CI** of the super-theory, we now define the resulting Galois connection as described in Section 3. We have a pair of functions $csp2t$, mapping from untimed CSP to *Circus Time*, and $t2csp$ mapping in the opposite direction.

Definition 27.

$$csp2t(P) \mathrel{\widehat=} \exists\, U\alpha \bullet \mathbf{CI} \circ \mathbf{TR}(P_{+T}) \;\; t2csp(P) \mathrel{\widehat=} \exists\, T\alpha \bullet \mathbf{CI} \circ \mathbf{TR}(P_{+U})$$
$$where\ T = \{tr_T, tr'_T, tr_C, tr'_C\},\ U = \{tr, tr', ref, ref', wait, wait', tr_C, tr'_C\}$$

That we have a Galois connection follows from Theorem 3 and Lemma 3.

This concludes the construction of the super-theory. In the following we explore the mapping of CSP operators into the super-theory and into *Circus Time*.

4.2 Using the Super-Theory

In this section we use the super-theory to relate CSP processes and their *Circus Time* counterparts. To that end, we first observe that the application of **TR** and **CI** to a reactive design yields a timed reactive process in the context of the super-theory of the form established by the following Theorem 4, where the function **S**, defined below, is used instead of **R**.

Definition 28. $\mathbf{S}(P) = \mathbf{R012_T} \circ \mathbf{CI0134} \circ \mathbf{R3_T} \circ \mathbf{R2}(P)$

The function **S** is the result of composing the healthiness conditions of the original *Circus Time* theory ($\mathbf{R0_T}$ to $\mathbf{R2_T}$, and $\mathbf{R3_T}$), together with our coupling invariants and **R2**. In the resulting design of Theorem 4, the conjunction in the postcondition is due to **CI2**: if the process terminates successfully in the untimed model, and without communicating any event, then time must not pass.

Theorem 4. *Provided ok' and $wait$ are not free in P,*

$$\mathbf{CI} \circ \mathbf{TR} \circ \mathbf{R}(P \vdash Q) = \mathbf{S}(P \vdash ((\neg wait' \land tr' = tr) \Rightarrow \#tr'_T = \#tr_T) \land Q)$$

We recall that all predicates of the CSP theory can be described as reactive designs, and so Theorem 4 describes all predicates of the super-theory. Similarly to CSP processes, they are the image of a design through a healthiness function. We observe that the proviso is standard for CSP processes, since their preconditions do not depend on the value of $wait$ as a result of **R3**.

Using Theorem 4, we can give a general characterisation of the result of applying $csp2t$ to a reactive design as established by the following Theorem 5, where $\phi \mathrel{\widehat{=}} (Flat(tr'_T) = Flat(tr_T))$ and $f = false$ and $t = true$.

Theorem 5. *Provided tr_C and tr'_C are not free in P and Q, and ok' and $wait$ are not free in P,*

$$csp2t \circ \mathbf{R}(P \vdash Q) = \mathbf{R_T}\begin{pmatrix} (\psi(P)[f/wait'] \lor wait'_T) \land \psi(P)[t/wait'] \\ \vdash \\ ((\phi \Rightarrow \#tr'_T = \#tr_T) \land \neg wait'_T \land \psi(Q)[f/wait']) \lor \psi(Q)[t/wait'] \end{pmatrix}$$

The proviso is satisfied by CSP processes as tr_C and tr'_C are not free in a reactive design. We obtain a timed reactive design with $\mathbf{R_T}$ applied. The design mentions the original pre and postconditions, P and Q, with ψ applied to them.

Definition 29. $\psi(P) \mathrel{\widehat{=}} P\begin{bmatrix} \langle\rangle, Flat(tr'_T) - Flat(tr_T), wait_T/tr', tr', wait \\ snd \circ last(tr_T), snd \circ last(tr'_T)/ref, ref' \end{bmatrix}$

These substitutions are a consequence of the definition of the coupling invariants and the healthiness condition **R2** of the original reactive design.

In a CSP process $\mathbf{R}(P \vdash Q)$, we expect $wait'$ not to be constrained, or even free, in P. In this case, the precondition of $csp2t \circ \mathbf{R}(P \vdash Q)$ is simply $\psi(P)$. We do not have, however, a healthiness condition that ensures that $wait'$ is not free in P. So, the actual precondition of $csp2t \circ \mathbf{R}(P \vdash Q)$ requires that $wait'_T$ must hold if P requires $wait'$ to be true.

The postcondition considers two cases. The second case is simpler: $wait'$ is admitted to be $true$ in Q, and so the postcondition is Q with the appropriate substitutions of ψ. The first case is when $wait'$ is admitted to be $false$: if no events are observed, that is, $Flat(tr_T) = Flat(tr'_T)$, then no time can pass, and termination also occurs in the timed model, with $wait'_T$ being $false$.

Having established the general results of mapping (untimed) CSP processes into the super-theory, and into *Circus Time*, in the remainder of this section we discuss the mapping of *Skip*, *Stop* and external choice.

Skip The result of mapping *Skip* into the timed theory is established by the following Theorem 6.

Theorem 6. $csp2t(Skip) = \mathbf{R_T}(true \vdash \#tr'_T = \#tr_T \land \phi \land \neg wait'_T)$

The precondition is also *true*, while the postcondition requires termination in the timed model $\neg wait'_T$, that no events are observed, and that no time must pass. This is the original definition of *Skip* in *Circus Time* [4].

Stop The result of mapping *Stop* through $csp2t$ is established by Theorem 7.

Theorem 7. $csp2t(Stop) = \mathbf{R_T}(true \vdash Flat(tr'_T) = Flat(tr_T))$

Like in CSP the precondition is *true*, while the postcondition is rather different: it only states that no events are observed, but termination is not guaranteed. This is unlike the definition of timed $Stop_T$ [4] reproduced below.

Definition 30. $Stop_T \mathrel{\widehat=} \mathbf{R_T}(true \vdash Flat(tr'_T) = Flat(tr_T) \land wait'_T)$

The application of $t2csp$ to $Stop_T$, however, yields the *Stop* of CSP as required, since $wait'_T \Rightarrow wait'$ is enforced by **CI3**.

External Choice Following from the result of Theorem 4, the next Theorem 8 establishes the induced definition of external choice in the super-theory.

Theorem 8. *Provided ok' and wait are not free in P and Q,*

$$\mathbf{TR} \circ \mathbf{CI} \circ \mathbf{R}((P \vdash R) \,\Box_{CSP}\, (Q \vdash S)) =$$
$$\mathbf{S}\left((P \land Q) \vdash \begin{pmatrix} ((R \land S) \lhd tr' = tr \land wait' \rhd (R \lor S)) \\ \land \\ ((\neg wait' \land tr' = tr) \Rightarrow \#tr'_T = \#tr_T) \end{pmatrix}\right)$$

The precondition is the conjunction of both preconditions just like in CSP, whereas the postcondition requires, in addition to that of CSP, immediate termination in the untimed model to become instantaneous. For example, in the case of the untimed process $Skip \,\Box\, a \rightarrow Skip$, there is no agreement on waiting, so either *Skip* terminates instantaneously, or the prefixing on the event a terminates at any time, without any waiting period observed. So, we have unexpected behaviour in a timed setting: although *Skip* terminates immediately, it does not resolve the choice, and although $a \rightarrow Skip$ can take time, we cannot observe its stable waiting states. We note that, in (untimed) CSP, termination also does not resolve a choice, and the above is the definition in the super-theory. We consider next the result of mapping external choice through the super-theory into the timed model is established by the following Theorem 9.

Theorem 9. *Provided tr_C and tr'_C, and ok' and $wait$, are not free in P and Q,*

$$csp2t(\mathbf{R}(P \vdash S) \,\Box_{CSP}\, \mathbf{R}(Q \vdash R)) = \mathbf{R_T}\left(\begin{array}{l}(\psi(P \wedge Q)[f/wait'] \vee wait'_T) \wedge \psi(P \wedge Q)[t/wait'] \\ \vdash \\ \left(\begin{array}{l}\psi(R \wedge S)[t/wait'] \lhd \phi \rhd \psi(R \vee S)[t/wait']) \\ \vee \\ ((\phi \Rightarrow \#tr'_T = \#tr_T) \wedge \psi(R \vee S)[f/wait'] \wedge \neg wait'_T)\end{array}\right)\end{array}\right)$$

This result closely follows that of Theorems 5 and 8. The precondition retains the conjunction of the original reactive design with appropriate substitutions.

In the postcondition there is a disjunction between, roughly, the usual conditional that characterises the choice, and an extra disjunct that stems from **CI**. It covers the possibility that one of the processes terminates, with $wait'$ being *false* in R or S, and termination also takes place in the timed theory, with $wait'_T$ being *false*, but it is instantaneous if no event is observed. The conditional considers the cases where R and S agree on waiting in the untimed model and, either no event is observed (ϕ) and R and S agree, or either process performs some visible event ($\neg\phi$). In any case, waiting in the untimed model does not lead to waiting in the timed model because of **CI3**. For example, the process $Skip \,\Box\, Wait\ 1$ has only one possible behaviour: immediate and instantaneous termination. We note that $Skip = Wait\ 0$, and so $Skip \,\Box\, Wait\ 1$ is a process of the form $Wait\ d \,\Box\, Wait\ (d + m)$ mentioned in Section 1.

We consider another example: $Wait\ 1 \,\Box\, Wait\ 2$. In this case, the only possible agreement between the processes is to wait 1 time unit. Termination of either process with no visible events cannot be instantaneous and so the behaviour after 1 time unit is miraculous. Finally, we consider $Wait\ 1 \,\Box\, (Wait\ 2\ ;\ a \rightarrow Skip)$, where there is a choice between terminating after 1 time unit, or performing the event a after 2 time units. In this case, and following Theorem 9, the processes can only agree on waiting for 1 time unit. After 2 time units, the event a can still be observed, but between 1 and 2 time units the process is miraculous.

Ultimately the definition of external choice induced from (untimed) CSP does not satisfy the timed properties of interest, namely, that early termination of one of the processes leads to termination. The definition considered in [4] does not correspond to this induced definition either. The approach we propose allows the study of different timed models, and, consequently, different definitions of timed external choice, through Galois connections which preserve the properties of untimed CSP. These variations can be obtained by adjusting the coupling variants piecewise. Further work is necessary to explore other possibilities.

5 Conclusion

The composition of theories is crucial for the unification of results in the UTP. Galois connections are an essential tool for the theory engineer as part of studying multiple aspects and relating definitions amongst different models.

The approach we propose promotes separation of concerns: healthiness conditions are defined separately to the relation between variables of the theories. The coupling invariants can be adjusted to yield models satisfying different properties, and provided the healthiness conditions are strengthening or weakening, or both, then Galois connections can be established. Although, we have used this technique to study only *Circus* and *Circus Time*, we expect it to be of more general use because it is based on general ideas of data refinement. Confirmation of this generality, however, is still to be established.

We have applied our approach to find a Galois connection between CSP and *Circus Time* that can justify the definition of the healthiness conditions and operators of *Circus Time*. This is different to that proposed in [4]. Our construction relies on a set of principles underlying the timed model and the appropriate definition of coupling invariants. This approach provides a way to study the induced definitions of operators, such as *Skip*, *Stop* and external choice.

The definition obtained for timed external choice is not entirely satisfactory in light of desired properties. Different versions of this operator are considered in [12,11]. In pursuit of a suitable treatment of external choice, it remains for us to study the relationship between untimed CSP and those models through a super-theory construction that preserves the semantics of untimed CSP.

Acknowledgments We would like to thank Simon Foster for his support regarding Isabelle/UTP. This work is funded by EPSRC grants EP/H017461/1 and EP/M025756/1.

References

1. Hoare, C.A.R., He, J.: Unifying Theories of Programming. Prentice Hall International Series in Computer Science (1998)
2. Roscoe, A.W.: Understanding concurrent systems. Springer (2010)
3. Oliveira, M., Cavalcanti, A., Woodcock, J.: A UTP semantics for Circus. Formal Aspects of Computing **21**(1) (2007) 3–32
4. Sherif, A., Cavalcanti, A.L.C., He, J., Sampaio, A.C.A.: A process algebraic framework for specification and validation of real-time systems. Formal Aspects of Computing **22**(2) (2010) 153–191
5. Wei, K., Woodcock, J., Cavalcanti, A.: *Circus Time* with Reactive Designs. In Wolff, B., Gaudel, M.C., Feliachi, A., eds.: Unifying Theories of Programming. Volume 7681 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 68–87
6. Cavalcanti, A., Sampaio, A., Woodcock, J.: A Refinement Strategy for *Circus*. Formal Aspects of Computing **15** (2003) 146–181
7. Harwood, W., Cavalcanti, A., Woodcock, J.: A Theory of Pointers for the UTP. In Fitzgerald, J., Haxthausen, A., Yenigun, H., eds.: Theoretical Aspects of Computing - ICTAC 2008. Volume 5160 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2008) 141–155
8. Woodcock, J., Davies, J.: Using Z: Specification, Refinement, and Proof. Prentice Hall (1996)

9. Schneider, S.: Concurrent and real-time systems: the CSP approach. Worldwide series in computer science. John Wiley (2000)
10. Wei, K., Woodcock, J., Cavalcanti, A.: New *Circus Time*. Technical report, University of York (February 2012) `https://www.cs.york.ac.uk/circus/publications/techreports/reports/Circus%20Time.pdf`.
11. Butterfield, A., Gancarski, P., Woodcock, J.: State Visibility and Communication in Unifying Theories of Programming. In: Theoretical Aspects of Software Engineering, 2009. TASE 2009. Third IEEE International Symposium on. (July 2009) 47–54
12. Canham, S., Woodcock, J.: Three Approaches to Timed External Choice in UTP. In: Unifying Theories of Programming: 5th International Symposium, UTP 2014, Singapore, May 13, 2014, Revised Selected Papers. Springer International Publishing, Cham (2015) 1–20
13. Morgan, C.: Programming from specifications. Prentice Hall (1994)
14. Woodcock, J., Cavalcanti, A.: A Tutorial Introduction to Designs in Unifying Theories of Programming. In Boiten, E., Derrick, J., Smith, G., eds.: Integrated Formal Methods. Volume 2999 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2004) 40–66
15. Cavalcanti, A., Woodcock, J.: A Tutorial Introduction to CSP in *Unifying Theories of Programming*. In Cavalcanti, A., Sampaio, A., Woodcock, J., eds.: Refinement Techniques in Software Engineering. Volume 3167 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 220–268
16. Spivey, J.M.: The Z notation: A Reference Manual. Prentice Hall (1989)
17. Ribeiro, P.: Super-Theories. Technical report, University of York (2016) `https://www-users.cs.york.ac.uk/pfr/reports/super-theories.pdf`.
18. Foster, S., Zeyda, F., Woodcock, J.: Isabelle/UTP: A Mechanised Theory Engineering Framework. In Naumann, D., ed.: Unifying Theories of Programming. Volume 8963 of Lecture Notes in Computer Science. Springer International Publishing (2015) 21–41
19. Banks, M.J., Jacob, J.L.: On integrating confidentiality and functionality in a formal method. Formal Aspects of Computing **26**(5) (2013) 963–992