Artificial Catalysed Reaction Networks for Search

Andrew Weeks and Susan Stepney

Department of Computer Science, University of York, Heslington, York, YO10 5DD

Abstract. There is a tension between biology's emphasis on perpetuation and the change required by biologically-inspired evolutionary algorithms. But there were radical changes when life first emerged from a prebiotic soup of chemicals. We investigate the possibility of taking ideas about the origin of life to develop a novel general purpose constructive search technique, inspired by ideas from autocatalytic reaction networks. Reactions combine partial solutions into fuller solutions; the reaction rates are catalysed by the products' fitnesses.

1 Introduction

Most biological processes are concerned with the survival and propagation of *existing* life. Evolution is hesitant, taking small, incremental steps in a game played out over thousands or millions of years. Yet evolutionary algorithms put change first, aiming to find answers from a radically different starting point.

By definition, biology did not exist before life emerged. Nature started with a collection of parts, a prebiotic soup of simple chemicals from which the first life forms emerged. A number of theories have been put forward about the origin of life and we investigate how one of them – Kauffman [1993]'s autocatalytic reaction networks – might be used to inspire a new constructive search algorithm.

The idea presented here is simple; the effort is in finding the right abstractions from chemistry to apply. The solution is certainly not (yet) perfected, but hopefully offers a glimpse at what could be possible.

2 Artificial and Real Chemistry

There is much research on various artificial chemistries; for a comprehensive review of the field see Dittrich *et al* [2001]. Much of the research relates to investigation of real chemistry, life, complexity, novelty, self-organisation, emergence, and so on (for example, [Fontana 1992] [Kauffman 1993] [Kauffman 1995] [McMullin 1997] [Gross & McMullin 2002] and many others). Some have tackled the problem of search using a chemistry-inspired approach (for example, [Kanada 1995]), but take a rather different approach from that suggested here.

Consider the chemical reaction $a\mathrm{A} + b\mathrm{B} \rightleftharpoons c\mathrm{C} + d\mathrm{D}$. This has a molecules of reactant A combining with b molecules of B to produce C and D. All reactions are **reversible**, and

in the **back reaction** C and D produce A and B. At **equilibrium**, these two reactions proceed at the same rate. The **equilibrium constant** K captures the ratio of *concentrations* of reactants, denoted [A] and [B], to products, [C] and [D], at equilibrium. The **rate constant** k captures the rate at which the reactions progress: k_1 is the forward reaction rate constant and the reaction rate is k_1[A][B]; k_{-1} is the back reaction rate constant and the reaction rate is k_{-1}[C][D]. (See the Appendix for more details.)

A **catalyst** is a substance that increases the rate of a reaction without being consumed itself; it speeds the rate, increasing k, by lowering the (free energy) barrier to the reaction. The speedup may be only marginal, or may be dramatic, increasing the rate by a factor of 10^{12} or more [Haynie 2001]. An **autocatalytic** process is one that catalyses itself: the products of the reaction catalyse their further creation. Kauffman [1993] [1995]'s suggestion for the origin of life is based on the **autocatalytic network**. He adopts a simple, abstract chemistry to explore his ideas, using strings to represent molecules. Single letters (*monomers*) represent the food supplied to the system. A reaction concatenates the reactants, adding a longer string (*polymer*) to the soup. Alternatively, two products may result from a reaction: back reactions may cleave large strings into smaller fragments. Kauffman also assumes that the products of some reactions catalyse *other* reactions. In his simple model, catalysts for reactions are chosen randomly, with a low probability. A network forms because "*as the length of the molecules increases, the number of kinds of molecules increases exponentially, but the number of reactions by which they convert from one to another rises even faster*" [Kauffman 1995]. So the ratio of catalysed reactions to chemicals eventually exceeds a certain threshold, and then a large collectively-autocatalytic system spontaneously emerges; this *catalysed reaction subgraph* consists of the vast majority of the reactants in the soup. Similar results are observed with more complex catalyst models [Kauffman 1993, 1995].

3 Chemistry-inspired search

Our search model explores the possibility of using some of these ideas and harnessing some of the constructive power that – if Kauffman is right – must surely be in these systems. In particular, we ask: Is it possible to start with a few simple elements and use these to construct a complete solution? Is it possible for a solution simply to emerge?

In these networks the different reaction rates, catalysis, the varying supply of reactants and products, and the interactions between them turn the whole system into a much more complex entity. This complex interplay of factors ultimately determines each molecule's 'fitness', evolves throughout the life of a network, and makes the ground rules through which reaction networks are able develop their emergent behaviour.

To use these ideas to inspire a search algorithm, we must find a *mapping* between the concepts in a reaction network and the sort of representations typically found in a search problem. The search algorithm must also build up a reaction network, directing this building towards solving some external problem. The network development must be simple to model, and it must be possible to apply it to many different problems.

Modelling partial solutions

The reaction network represents the entire solution space, so it must be possible to model all solutions and partial solutions within this space. The representation of a (partial) solution is as a *molecule* within the network. The connections between the molecules are modelled as reactions.

Due to the large size of most search spaces, genetic algorithms (GAs), and many other meta-heuristic search algorithms, model only a small part of it at any given time. We do the same here. Kauffman initialises his networks with small food molecules, which react to form larger molecules. In our reaction network, the food comprises the simplest "atomic" *partial solutions*. Food is reacted to create larger molecules that participate in subsequent reactions. As molecules are created, the associated *reaction links* between the participants are also created. Through a series of reactions, complete solutions can be created. The molecules in the network do not have a set position within the final solution; this is fixed as full solutions are built up.

We look at a (very!) simple case study problem: "*n* 1s". This searches a solution space of the strings drawn from the alphabet $\Sigma = \{0,1\}$ for that string of length *n* that comprises all 1s. Partial solutions are substrings. Reactions produce new molecules by concatenation, such as $01 + 11 \rightarrow 0111$. *n* 1s is by no means a challenging problem for humans, neither is it a difficult problem for computers: hill climbing works admirably. However, it *is* a problem that GAs find relatively tricky; for example, our comparison GA implementation tried over 9 million solutions before finding the answer to 512 1s. Also, it is a simple enough problem that it is a useful basis for analysis.

Fitness-based catalysis

Here, an external fitness function is imposed on the reaction network, to direct reactions to form good solutions. This is achieved by catalysis. Catalysing a reaction increases its rate and produces more product. We want more of the *fitter* products, so the catalysed rate is related to a product's fitness.

The aim of the network is to build up increasingly fit solutions by combining good intermediate molecules already in the network. Good building blocks have a high reaction rate, so quickly appear in large numbers. Ideally, when a new reaction is added to the network, the best possible molecules (partial solutions) should be chosen as reactants. So participants are selected probabilistically based on their concentrations. By integrating catalytic activity over time using concentrations, it smoothes over short term perturbations in the fitness function, making the network less sensitive. This is also consistent with the approach used in most GAs, in which common genes tend to appear in more offspring. The higher chance of selecting good building blocks as participants in new reactions reinforces fit parts of the final solution at an early stage. This is could prove beneficial in directing the search, but does not preclude the network from trying even fitter alternatives if they are found.

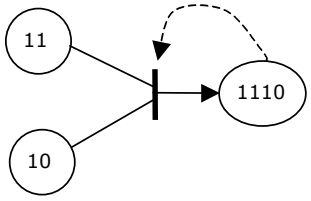

Figure 1 Each molecule is an autocatalyst

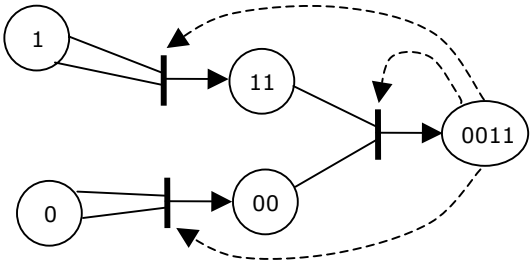

Figure 2 Catalysing back along reaction paths

Catalysis Network

Catalysis is applied to reaction paths in the network. Where a partial fitness function exists, each molecule acts as an autocatalyst for its own production (fig 1). If there is no partial fitness function, any final products on the reaction path catalyse their building blocks (fig 2); we do not address that problem (the *credit assignment problem*) here.

As well as the forward path dependence on good building blocks that already exist within the network, there is a backward path dependence between catalyst and reactants, which introduces a degree of competition and cooperation between elements within the network.

Reaction rates are used to indicate fitness; however, the existence of equilibria is essential in *regulating* reactions. Each individual reaction influences all others on the same path, and by extension in the entire network:

- fit molecules are provided with positive feedback, increasing their rate of production
- the creation of less useful molecules is slowed by their concentration build-up; it could even lead to the back reaction (their destruction) dominating
- the balance of concentrations and reactions shifts continually as new molecules are added to the reaction network, establishing new paths that utilise or largely supersede older ones

In a suitably large network, the effects of all of these interactions should give rise to intricate and complex – yet controlled – emergent behaviour.

4 Implementation

Even for relatively simple sets of reactions, calculating information such as equilibrium positions can be difficult; performing the same sorts of analyses on reaction networks would be almost impossible. Fortunately, the reaction can easily be simulated by devolving responsibility for these calculations down to the molecular level.

We have implemented a simple artificial reaction network system. It comprises molecules representing partial solutions, situated in a reaction network linking them together with fitness-catalysed reaction rates. All (forward) reactions created have two reactants (though the same molecule could be selected twice). While this is a simplification of wet chemistry, "higher order" reactions with more than two reactants are relatively uncommon in nature, too.

Initialise: The simulation is initialised with only food molecules (atoms), and no reactions.

The simulation then loops around the following two steps until a solution is found, or the maximum number of steps has been taken.

Grow network: The network is updated (grown) either by combining two molecules (forward reaction) or by splitting one (back reaction). Reactants are selected for this process probabilistically based on their concentrations. The resulting product is compared to the currently existing molecules. There are three possibilities:

- *new product*: the new product is added to the molecule pool with zero concentration; the new generating reaction is added to the network with a forward reaction rate k_1 given by the fitness of the product molecule, and a back reaction rate $k_{-1} = k_1/K$. K is taken to be a constant of the run; typically $K = 10$ (see later).
- *existing product, new reaction*: the new generating reaction is added to the network with a reaction rate given by the fitness of the product molecule
- *existing product and reaction*: an existing reaction has been duplicated, so the process tries again to create a novel reaction

Update Concentrations: The molecules' concentrations are updated, depending on their old concentrations and the fitness-based reaction rates, using a discretised version of equation 9. The concentrations of the food molecules are maintained at their initial values. A parameter allows the concentrations of the other molecules to be updated m times per network growth step.

5 Results: 8 1s and 16 1s problems

For the 8 1s problem, the run length varies considerably and is quite dependent on which reactants are selected in the early stages of network growth. If these are judiciously chosen, the answer is typically found after the network has grown to 10–15 molecules; however other runs grew to 200–300 molecules before finding the solution, a fair proportion of the search space (search space size is $2^n + 2^{n-1} + \ldots + 2^1 = 510$ for $n = 8$).

About 75% of the runs produced the solution almost immediately. Features typical of most runs that do *not* complete almost immediately are: (i) at first, the fitness of the best solution rises rapidly as good molecules are combined (a short run finds the answer at this

stage); (ii) the next new molecule produced is relatively unfit, but this is followed by another relatively good solution; (iii) this fitness oscillation continues throughout the run.

The results from larger n 1s problems follow a similar pattern: promising results are produced initially, followed by more mediocre solutions. Solutions for 16 1s are typically produced after creating 80–500 molecules (in a search space of size 130,000), although again some runs last much longer.

For reasons examined below, the current version of the program does not yet scale well, so was not tested on significantly larger problems.

Fitness oscillation behaviour can be explained by the concentrations of intermediates. Initially, the network chooses fit building blocks as reactants as they have been produced in greater quantities. However, it does not take particularly long for the concentration of molecules that are less fit (but unused as reactants) to exceed that of more valuable molecules that are now part of reaction chains. The next novel reaction is likely to use one of these less fit molecules because of its high concentration. This quickly reduces its concentration (and that of closely linked molecules, often with similar constituent atoms, through back reactions), allowing the next new reaction to use fitter molecules again.

Network flow: Chemists rely on a number of different reactor types to create the products they need. The two main types are: batch reactors, where chemicals are added and then left to reach equilibrium (a closed system); and flow reactors, where reactants are continually refreshed and products removed (an open system). Systems in nature behave somewhere between these extremes: they are open systems, but products are not uniformly swept away and replaced as soon as they are created.

If the food used by the network is replenished, material continues to flow through the system, preventing it from approaching equilibrium. We achieve this by keeping the stocks of food at the same concentration throughout. The concentrations of the molecules at the end of the reaction chain soon increase well above that of the other molecules. These continue a roughly linear rise in concentration for the remainder of the run. The concentrations of the intermediate products rise initially, then reach an approximately steady state, being produced and consumed at roughly the same rate.

Some difficulties are experienced if this flow model is used when a network is being built. The concentration of the molecules at the end of reaction chains keeps on rising, making them likely candidates for the next novel reaction. Certainly this has some nice properties, such as building up full solutions quickly by tending to combine large molecules. However, some of these large molecules are not useful: this bias against smaller, but essential, building blocks is problematic.

6 Strengths and weaknesses of the algorithm

As is evident from the preliminary results, there are some issues with the current version:

- The system punishes good components. Good molecules often form part of other good molecules, which have high reaction rates. These larger molecules rapidly deplete the stock of their fit constituents. In short, the good building blocks are used up making nearly, but not quite, perfect solutions. Adjusting *K* makes little difference: initially the smaller molecules remain in higher concentration, but this just encourages more reactions to be created using it.
- The system does not remove any unfit components. "The essential requirement for a system to be self-selective is that it has to stabilise certain structures at the expense of others" [Eigen & Schuster 1977]. Almost all GAs do this: a GA would be unproductive if the population grew because there were no 'death' each generation.

However, the approach does have some nice properties.

Lack of sensitivity to partial fitness function: Due to the equilibrium relationships between molecules, the algorithm is not too sensitive to fitness values for partial solutions, implying that the partial fitness does not need to be calculated with any great fidelity. This is useful for problems where the partial fitness has to be estimated. Increasing the rate of a reaction by a factor of 10 to 100 has little effect on the concentrations of other unconnected molecules. Dramatically reducing the rate of an essential component has ramifications for larger molecules that rely on it; and increasing the production rate of most intermediates by a factor of 10 usually yields associated final product concentrations that are about 3 times greater than they would otherwise be. But this is a substantial corrective influence that has been exerted by the network.

The use of equilibrium reactions is crucial to preventing over-sensitivity to vagaries in the fitness function. Acting through a complex web of interactions and interdependencies, even if the reaction rate of some molecules within the network is wildly inaccurate, the network dramatically curtails their production through a combination of reversible reactions and limiting the flow of material to set reaction pathways.

This is perhaps illustrated most convincingly by comparing the effect of increasing the fitness of a component by an order of magnitude. In a normal GA, the fit component would rapidly take over the population to the exclusion of everything else. Here, the component's production is greatly restricted by other molecules in the network; the system is regulating itself. With no outside interference, global knowledge or complex rules, the reaction network has clamped down on a runaway reaction and maintained itself in a stable state.

This is exactly the sort of self-organising behaviour one would hope to see: catalysts used in biological processes speed up reactions by factors as great as 10^{12} without running out of control; Eigen & Schuster [1977]'s and Kauffman [1995]'s models rely on these subtle relationships in their models of early life.

Novelty from back reactions: back reactions are crucial to regulating concentrations. They have an additional effect in the simulation. A back reaction splits an existing molecule in two. If those components do not yet exist in the simulated network, they

might become valuable new building blocks. If they do, the back reaction could still create additional reaction paths, providing new routes through the reaction graph. This could speed up the molecule's production, or reduce its consumption of molecules along its original reaction path, affecting their concentrations and chance of being selected for novel reactions.

Prevention of premature convergence: premature convergence is naturally prevented, primarily through including some history within the model. By storing partial solutions, the system retains information about the search space that has been explored so far. Suppose most of a search has concentrated on a small part of the solution space. If a very different, but excellent, alternative is found, the search will implicitly backtrack through the reaction network to explore this area too. The reaction network provides a dynamic, just-in-time mechanism for avoiding convergence. In GAs, premature convergence is alleviated by *mutation*: the relationship between these processes is still to be understood.

Innovation operator: Creating a novel reaction acts as the innovation operator in the search. It manipulates the equivalent of GA schemata directly, combining them based on their fitness; there is no need to evolve them implicitly by maintaining a large population, as with a GA. It is also not destructive, so should not suffer from problems akin to introns that can plague GP.

7 Improving the search through better chemistry

Equilibrium constant *K*: *K* is currently fixed for a particular run, and is the same for all reactions; typically K = 10. (A number of different values were tried for *K*, with limited effect.) There is an argument for adopting a more complex, or even dynamic, *K*:

- If *K* were decreased as the size of molecule grows, it might prevent final products from being made in such large quantities, leaving more medium length molecules to react. (This is unlikely to fully address the problems noted earlier, though.)
- If *K* were adjusted dynamically, depending on the demand for a molecule, it could preclude the molecule either from being 'sucked dry' or building large, unused stocks of itself. It is unclear what effect this would have on the reaction network as a whole: how quickly would *K* be changed, and by how much? A mechanism employed by biological systems with (ostensibly) similar effects is regulation of catalysts. However, this requires a higher level of control than is assumed to be present in a catalytic network; further it is not clear what form this control should take, given that the answer is not known to the system.

Open system flow: A better model of flow of food into the system and the inclusion of removal of material from the reaction network could allow the inclusion of significant additional selective pressure.

Reactant selection: Molecules to make a novel reaction are selected entirely based on their *concentration*. The rate should be included as well. The difficulty applying this here is that that the reaction rate is determined only once a reaction exists. It may be possible to estimate rate from other similar reactions, although exactly how is not clear.

Catalyst concentration: The current system does not consider the concentration of catalysts when determining reaction rates; they are assumed to be present in large excess. This is not actually true, since all reactions are directly autocatalysed by their products. If the catalyst concentration were considered, reaction rates would tend to follow an autocatalytic S-curve [Logan 1996]. This would prevent a molecule from being made as quickly at low concentrations, stopping the products of novel reactions from using up valuable building blocks. It is possible that this would go some way to dampening the oscillations seen in the current results. It would also reduce the rate at which a molecule's concentration can be reduced through back reactions: as the concentration decreases, so does the catalytic effect and hence the rate.

8 Application to other problems

One of the aims is to create a search algorithm that is not domain specific, one that can be applied to a wide variety of problems. This section outlines how reaction networks could be applied to the *n*-queens problem, to the Travelling Salesman Problem (TSP), to program generation, and to formal proof.

***n*-queens**: The problem is to place *n* chess queens on an $n \times n$ chessboard such that no two queens are attacking each other: they do not share a column, row, or diagonal. Since no two queens can be in the same row, we can represent the solution in terms of only the columns. So [1423] represents (queen 1 in row 1, column 1), (queen 2 in row 2, column 4) and so on. The full solution is a suitable permutation of 1..*n*, such as [1423]; partial solutions are permutations of subsets of 1..*n*, such as [14] and [32], and the food comprises the single numbers in 1..*n*, [1][2][3][4]. Reactions concatenate partial solutions; ones that would put two queens in the same column are forbidden.

Any partial fitness function devised for *n*-queens is unlikely to be as accurate as that used in *n* 1s, as the problem cannot be broken down fully, but the network's ability to control the under- and over-production of its molecules should prevent this from being a major concern.

TSP: The problem is to find a short route through a network of cities. The full solution is a suitable permutation of 1..*n* giving the order of city visits. Partial solutions are partial tours, permutations of subsets of 1..*n*, and the food comprises the single numbers in 1..*n*, [1][2][3][4]. Reactions concatenate partial solutions; ones that would put the same city in the solution more than once are forbidden. The partial fitness function is the length of the partial tour.

Program generation: The problem is to find a computer program to calculate a particular result. The reaction network can be used to build programs as trees, in the style of GP solutions. The food molecules are the functions and terminals specified in the problem. Molecules are subtree fragments. Reactions concatenate by subtrees; terminals are attached to the leaf nodes only, and prevent further concatenation at that point. Typing, spatial or other tree constraints could be imposed on the reactions. Determining the relevant partial fitnesses is an open problem.

Proofs: The problem is to find a proof of a conjecture. The food molecules are the conjecture to be proved, and any relevant lemmas and axioms. Partial solutions are other, inferred, conjectures. Reactions produce new conjectures by applying inference rules to the reactants. The solution is the molecule *true* or *false*. The proof is the reaction pathway from the food conjecture to the solution. Again, determining the relevant partial fitnesses is an open problem.

Back reactions could prove to be particularly valuable here. Suppose that a large, but only moderately good proof had been constructed, yet contains within it is a really useful lemma. If the proof were split, via a back reaction, into the lemma and the rest of the proof, the lemma could then be incorporated into other, fitter proofs.

GA precursor: One avenue that has not been explored, but which could prove fruitful, is to combine this approach with a standard GA. The initial population for the GA would be created by running the algorithm outlined here, and using any final (but non-optimal) solutions, or "padding out" good partial solutions. This would hopefully give a better starting position for the GA, and would overcome many of the points raised above.

9 Conclusions

We have outlined a novel search algorithm inspired by catalysed chemical reaction networks. It performs well on small problems, but some issues with scalability have arisen. However, we believe this to be a fruitful approach, and are proceeding to look at introducing better chemistry, and trying more challenging problems.

One aspect of the original inspiration, Kauffman [1993] [1995]'s autocatalytic networks, that has not been addressed in this algorithm is the *random* choice of reaction to catalyse, and the resulting emergent self-catalysing network. That is also a topic for further study.

10 Acknowledgments

We would like to thank Leo Caves for advice on the chemistry, Peter Mendham for his thoughts and comments, and Fiona Polack for detailed comments on an earlier draft.

References

[1] P. Dittrich, J. Ziegler, W. Banzhaf. Artificial Chemistries: a review. *Artificial Life* **7**:225–275, 2001
[2] M. Eigen, P. Schuster. The Hypercycle: a principle of natural self-organization. *Naturwissenschaften* **64**:541–565, 1977
[3] W. Fontana. Algorithmic Chemistry. *ALife II*, 159–210. Addison-Wesley 1992
[4] D. Gross, B. McMullin. The Creation of Novelty in Artificial Chemistries. *ALife VIII*, pp 400–409. MIT Press, 2002
[5] H. Gutfreud. *Kinetics for the Life Sciences*. Cambridge University Press, 1995
[6] D.T. Haynie. *Biological Thermodynamics*. Cambridge University Press, 2001
[7] Y. Kanada. Combinatorial problem solving using randomized dynamic composition of production rules. *ICEC'95*, pp 3784–9, IEEE, 1995
[8] S.A. Kauffman. *The Origins of Order*. Oxford University Press, 1993
[9] S.A. Kauffman. *At Home in the Universe*. Viking, 1995
[10] S.R. Logan. Fundamentals of Chemical Kinetics. Longman, 1996
[11] C.K. Mathews, K.E. van Holde, K.G. Ahern. *Biochemistry*. Addison-Wesley, 2000
[12] B. McMullin. SCL: An Artificial Chemistry in Swarm. SFI WP 97-01-002. 1997

A Equilibrium constant *K* and rate constant *k*

Here we give a *very* brief overview of the properties of chemical reactions that we use to inspire our artificial reaction network search algorithm. See, for example, Gutfreud [1995], [Logan 1996], Haynie [2001] for more detail.

Whether a chemical reaction is **thermodynamically favourable** or not is determined by the *change* Δ in the **Gibbs free energy** G:

$$\Delta G = \Delta E + P\Delta V - T\Delta S \tag{1}$$

where E = energy; P = pressure; V = volume; T = temperature; S = entropy. If $\Delta G < 0$, the reaction is *favourable*. Consider the reaction $a\mathrm{A} + b\mathrm{B} \rightleftharpoons c\mathrm{C} + d\mathrm{D}$. At equilibrium, there may be very little of the reactants, very little product, or something in between. The Gibbs free energy determines the position of equilibrium. ΔG is the free energy of the products minus that of the reactants:

$$\Delta G = G_{\text{products}} - G_{\text{reactants}} = cG_{\mathrm{C}} + dG_{\mathrm{D}} - aG_{\mathrm{A}} - bG_{\mathrm{B}} \tag{2}$$

The G_X are the **chemical potentials** of the components. The concentrations of reactants and products vary between reactions, so it is useful to normalise the chemical potentials to 1M (mole) concentration. It can be shown that [Mathews *et al* 2000]:

$$G_{\mathrm{A}} = G_{\mathrm{A}}^{0} + RT \ln[\mathrm{A}] \tag{3}$$

where $G_A{}^0$ = chemical potential at 1M; [A] = concentration of A. So the whole reaction's free energy change can be written as:

$$\Delta G = \Delta G^0 + RT \ln\left([\mathrm{C}]^c[\mathrm{D}]^d / [\mathrm{A}]^a[\mathrm{B}]^b\right) \tag{4}$$

where R = gas constant. At equilibrium, the concentrations have fixed equilibrium values, for a given temperature and pressure. The **equilibrium constant** K is:

$$K = \left([\mathrm{C}]^c[\mathrm{D}]^d / [\mathrm{A}]^a[\mathrm{B}]^b\right)_{eq} \tag{5}$$

At equilibrium there is no favourable direction for the reaction: $\Delta G = 0$. (This does not mean that no reactions are occurring; rather, the forward and back reaction rates are identical.) Setting $\Delta G = 0$ in eqn (4), substituting for K from eqn (5), and rearranging:

$$K = \exp\left(-\Delta G^0 / RT\right) \tag{6}$$

Gibbs free energy can be used to determine the *direction* of a reaction, but says nothing about how *quickly* the reaction happens. The **rate** of a chemical reaction obeys the law of mass action, which states (in this context) that "*the rate of reaction is directly proportional to the product of the concentrations of the reactants*" [Gutfreud 1995]. The constant of proportionality (at given temperature and pressure) is called the ***rate constant***.

Consider the simple reaction:

$$\mathrm{A} + 2\mathrm{B} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{C} \tag{7}$$

where k_1 = forward reaction rate constant; k_{-1} = back reaction rate constant. The actual reaction rates are determined by the product of the concentrations and the rate constant, in this case, forward reaction rate = $k_1[\mathrm{A}][\mathrm{B}][\mathrm{B}] = k_1[\mathrm{A}][\mathrm{B}]^2$, reverse reaction rate = $k_{-1}[\mathrm{C}]$. The change in concentrations is thus:

$$\frac{d[\mathrm{A}]}{dt} = 2\frac{d[\mathrm{B}]}{dt} = k_1[\mathrm{A}][\mathrm{B}]^2 - k_{-1}[\mathrm{C}] = -\frac{d[\mathrm{C}]}{dt} \tag{8}$$

In general:

$$\begin{aligned} &a\mathrm{A} + b\mathrm{B} + c\mathrm{C} + \ldots \rightleftharpoons x\mathrm{X} + y\mathrm{Y} + z\mathrm{Z} + \ldots \\ &a\frac{d[\mathrm{A}]}{dt} = k_1[\mathrm{A}]^a[\mathrm{B}]^b[\mathrm{C}]^c \ldots - k_{-1}[\mathrm{X}]^x[\mathrm{Y}]^y[\mathrm{Z}]^z \ldots = -x\frac{d[\mathrm{X}]}{dt} \end{aligned} \tag{9}$$

If the Gibbs free energy of the products is less than the reactants, the overall reaction is favourable. However, but there is a **free energy barrier** to overcome before the reaction can occur, the height of which determines the reaction rate k.