



International Colloquium on Graph and Model
Transformation On the occasion of the 65th birthday of
Hartmut Ehrig
(GraMoT 2010)

From Graph Transformations to Differential Equations

Mayur Bapodra and Reiko Heckel

21 pages

From Graph Transformations to Differential Equations

Mayur Bapodra and Reiko Heckel

Department of Computer Science, University of Leicester, UK
{mb294, reiko}@le.ac.uk

Abstract: In a variety of disciplines models are used to predict, measure or explain quantitative properties. Examples include the concentration of a chemical substance produced within a given period, the growth of the size of a population of individuals, the time taken to recover from a communication breakdown in a network, etc.

The models such properties arise from are often discrete and structural in nature. Adding information on the time and/or probability of any actions performed, quantitative models can be derived. In the first example above, commonly referred to as kinetic analysis of chemical reactions, a system of differential equations describing the evolution of concentrations is extracted from specifications of individual chemical reactions augmented with reaction rates.

Recently, this construction has inspired approaches based on stochastic process specification techniques aiming to extract a continuous, quantitative model of a system from a discrete, structural one. This paper describes a methodology for such an extraction based on stochastic graph transformations.

The approach is based on a variant of the construction of critical pairs and has been implemented using the AGG tool and validated for a simple reaction of unimolecular nucleophilic substitution (SN_1).

Keywords: stochastic graph transformations, chemical reactions, law of mass action, ordinary differential equations

1 Introduction

In chemistry, understanding the kinetics of reactions, i.e, the evolution of concentrations of chemical agents over time, is essential for a variety of purposes ranging from the interpretation of experiments in the lab to the planning of large scale industrial systems. Analogies between chemical reactions and rewriting have inspired approaches like CHAM [2] where chemical reactions serve as a paradigm for concurrency, or Kappa [8] focussing on modelling biological systems. In graph rewriting the modelling of chemical reactions has been studied in [7]. The idea is to present molecules as graphs (or graph expressions) and to formalise reaction equations as graph or term rewrite rules. Such an encoding offers opportunities for cross-fertilisation, transferring concepts, problems, and ideas between chemistry and computer science.

In this paper, we will exploit this in order to rephrase chemical reaction kinetics in terms of stochastic graph transformations. More precisely, we want to derive the ordinary differential equations (ODEs) that describe the evolution of concentrations of chemical species over time. While in simple cases these equations can be derived by hand using the law of mass action

[15], for large, complex reaction networks automation is required. We provide the formalisation necessary for automating the approach and conduct a feasibility study based on a prototype implementation. More generally, we hope to contribute to bridging the gap between discrete modelling techniques focusing on change of structure and continuous ones modelling quantitative changes. While focussing on chemical reactions, the technique is a general one and could be applied to other systems that can be described in a similar way.

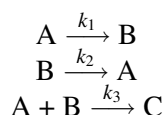
Specifically, our approach is based on stochastic graph transformations [11] which combine rules to capture the reactive behaviour of the system with a specification of rate constants governing the speed at which the reactions occur. We have studied (and continue to do so) the problems of stochastic simulation and model checking of such specifications [13, 16]. These analysis approaches suffer from scalability issues in terms of the size of the populations considered (such as the number of starting molecules allowed in the system). Ordinary differential equations are an alternative level of abstraction less prone to this problem.

This paper is organised as follows. First, a treatment of background and related work is given. This is followed by a formal description of the derivation of ODEs from graph transformation systems. Then our case study of the SN_1 reaction is introduced, and the results obtained by the analysis are given, including a short discussion of tool support. Finally, the conclusion contains a discussion of required further work, and ways to evolve the current methodology.

2 Background

The derivation of ODEs from behavioural specifications is based on (variations on) the law of mass action. In chemistry, this law uses reaction rates of rules at the level of individual molecules reactions to predict the evolution of concentrations of substances in large quantities.

In order to state the dynamics of a chemical system in terms of ODEs, the complete set of possible reactions in the system must be known. For each reaction, we must also know how many molecules of each species are consumed and produced by that reaction. We build up what is known as a stoichiometric matrix which relates each elementary reaction to each molecular species in the system by the aggregate effect the reaction has on that species' population. Consider the following 3 elementary reactions which comprise an example reaction mechanism.



The k_x values are rate constants for each of these reactions. The rate law of any elementary reaction is defined as the product of a rate constant for that reaction (e.g. k_1 for the first reaction) with the concentrations of any reactants needed in the initiation of that reaction (e.g. the concentration $[A]$ of A for the first reaction). The rate law gives us an indication of the “speed” at which that elementary reaction goes. The rate laws for the reactions above are given respectively as

$$\begin{array}{l} k_1[A] \\ k_2[A] \\ k_3[A][B] \end{array}$$

To obtain an ODE for this reaction system with respect to a particular reactant we consider how each elementary reaction affects the population of that reactant. For example, A is consumed once by k_1 , produced once by k_2 , and consumed once by k_3 . To get an ODE for A, we multiply how many net molecules of A are produced by a reaction (negative values indicate net consumption) by the rate law for that reaction, and sum over all of the rate laws. For A, this gives:

$$d[A]/dt = k_1[A] + k_2[A] - k_3[A][B]$$

From the procedure above, it is clear that all that is needed to derive these ODE's is an indication of what effect each reaction has on each chemical species in the system. This information is entirely contained in the stoichiometric matrix described above. The rate laws are extracted by multiplying the rate constant for each reaction by the concentrations of every molecule needed to initiate the reaction. If we do this for every elementary reaction, we can produce a rate law vector of length n , where n is the total number of elementary reactions. A multiplication of this vector and the stoichiometric matrix produces a system of ordinary differential equations:

$$d[X]/dt = S \cdot R \quad (1)$$

where $d[X]/dt$ is the differential with respect to time, t , of a chemical species, X , in the system, S is the stoichiometric matrix, and R is the rate law vector. Thus, the derivation of ODEs is straightforward once the stoichiometric matrix is established.

	A	B	C
k_1	-1	1	0
k_2	1	-1	0
k_3	-1	-1	1

Table 1: Example stoichiometric matrix

The stoichiometric matrix for our simple example reaction mechanism is given in Table 1. This can be seen as an incidence matrix for a simple net with places A, B, C and transitions k_1, k_2, k_3 , shown in Figure 1. As discussed in Section 2, a Petri net whose transitions are labelled by rate constants can be translated into a system of ODEs in much the same way as a system of reaction equations [10]. In fact, starting out from the incidence matrix of the net, the ODEs can be written out immediately following the matrix equation (1) above. It therefore suffices to encode the graph transformation system into a place transition net, which is the aim throughout the rest of this section.

3 Related Work

Existing approaches to the derivation of ODEs are based on process calculi, biological modelling languages, nets or rewrite systems. In the first category, Cardelli [4] converts expressions in the Chemical Ground Form (CGF) process algebra (a close formalisation of chemical reaction rules)

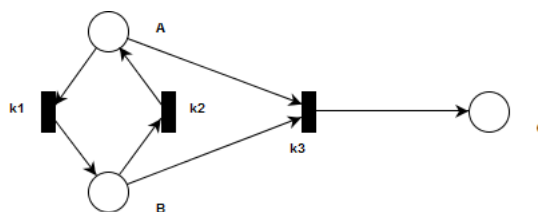


Figure 1: Place Transition Net representing reaction mechanism of Table 1

into ODEs. PEPA [14], a stochastic process algebra that incorporates rate information into communication activities, can express a reaction system as a set of interacting sequential components. These interactions can be expanded to a state representation of the reaction system, from which an activity matrix bearing a resemblance to the stoichiometric matrix can be extracted. The stoichiometric matrix describes the consumption or production of chemical species by individual reactions and is a necessary ingredient to the derivation of ODEs (see Section 4).

As described in Section 2, this matrix bears a close similarity to the incidence matrix of a Petri net, so it comes as no surprise that nets have been utilised to the same end. In [10] it is described how a discrete Petri net can be converted into a continuous one by allowing places to have a positive real number of tokens representing the concentration of that particular chemical species in the system. The ODEs can be deduced from the incidence matrix for such a Petri net.

Both the process algebraic methods above and the Petri net method are limited in that the specification of reactions requires the molecules involved to be defined in their entirety. For an extremely large reaction network comprising hundreds of reactions between molecules, such a specification becomes unfeasible. In contrast, we follow the work of Faeder et. al [8] and Feret et. al [9] in using a rule-based approach. Here, we model reactions at the level of functional groups rather than entire molecules. Reactions are described by local context, depicting only those elements that are directly involved in the reaction [5]. In this way, a fully defined set of concrete reactions between all reacting species is reduced to a smaller set of reaction rules that can be contextualised to fit possibly numerous specific pairs of reactants.

The BioNetGen Language (BNGL) [8] is a versatile rule-based modelling language, developed with biochemical networks in mind. The language can be used to specify rules and starting reactants as terms, which can be visualised graphically. To give a continuous numerical solution for the ODEs representing the system, the BioNetGen software constructs the entire reaction network via the systematic and exhaustive application of rules to a set of seed species, and their subsequent derivatives, until no further change occurs.

In contrast to measuring the concentration of species, [9] introduces the concept of “coarse-graining”, where so-called fragments are an alternative unit of dynamics. Fragments are sub-graphs of species and can be envisioned as the smallest units at which the system can make distinctions on the dynamics. This reduces the ODEs necessary to describe a system to a much smaller set, as one fragment encapsulates a number of species, including their reactivity. Fragments are constructed directly from the rules according to a number of constraints. The modelling language used is Kappa, which again has an algebraic and a graphical representation.

We aim to follow these two approaches, rephrasing their constructions in terms of graph trans-

formation theory to understand the relationship between the two fields and explore possible mutual benefits. Specifically, we attempt the derivation of ODEs via critical pair analysis of graph transformation rules. This is essentially a combination of the static analysis approach in [9] with the BNGL method in [8] which requires an enumeration of the chemical species in the system. However, where BNGL requires execution of the rules on a set of seed species (which may require numerous instances of each starting reactant), our method only requires the starting materials to be defined more generally, without considering repeated instances. The adherence to a graphical representation is particularly useful in reducing the semantic gap between the model and real chemistry [3, 18], since graphical representations potentially bear a conceptual similarity to structural formulae used in Chemistry. Atoms (or molecular species, for a higher level of abstraction) can naturally be seen as nodes in a graph, with the bonds between them represented as bi-directional edges.

We differ slightly from both the approaches in that we attempt to define a more general consideration of rate constants. In Chemistry, the reactivity of a site in a molecule (and therefore the rate constants for reactions involving that site) can be affected by additional context, possibly several positions away in the carbon chain. This additional context may not be a defining part of the functionality of that site, and therefore would not necessarily be included in the rule that captures that functionality. In both the coarse-grained and BNGL approaches, to account for this additional rate-altering context, a general rule must be expanded to include that context, so that a different rate constant can be assigned. This can be seen as an instantiation of the general rule so that it is applicable in a more specific instance. In both approaches, the instantiation must be done manually. In our approach, we automatically instantiate all general rules to apply to specific, singular molecular context so that a distinct rate constant can be applied to each of these instantiated reactions. We make the assumption that additional context always has an effect on the rate constant. Also, this instantiation may be important in systems outside the chemical and biological realms. As our aim is to avoid restricting the application domain, we incorporate this generalisation into the methodology presented here.

4 From Graph Transformation to ODEs

As stated in Section 2, the problem of deriving ODEs from a stochastic graph transformation system can be reduced to encoding the system into a suitable place-transition net. Before describing this process, in this section we will first discuss the modelling of chemical molecules by graph transformation systems.

In [7], hypergraphs [6] are used to represent molecules, with hyperedges modelling atoms or molecular subgroups and nodes representing bonds. We follow a similar approach, but use bipartite graphs instead, where both atoms and bonds are represented as nodes. Bond nodes are connected to atom nodes by edges. Edges connect bond nodes with each other. Figure 2 illustrates this in the case of a water molecule. This representation allows us to use tools such as AGG to analyse a system of reaction rules. Formally, we follow the double-pushout approach [6] to the transformation of typed graphs.

A typed graph transformation system $\mathcal{G} = (TG, C, P, \pi)$ consists of a type graph TG , a set of constraints C , a set of rule names P and a function π associating with each name $p \in P$ a

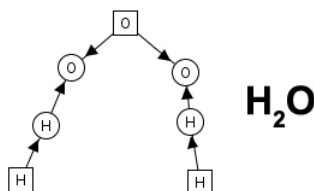


Figure 2: Graph representing H_2O . Atoms are square nodes (with their chemical symbol given by the node label), while round nodes represent bonding capacity for that element. The square O node is connected to two two round O nodes, since oxygen has a valency of 2.

span of injective TG -typed graph morphisms $\pi(p) = (L \xleftarrow{l} K \xrightarrow{r} R)$. The subset of TG -typed graphs satisfying the constraints C will be denoted by $\mathcal{G}ra(TG, C)$. The application $G \xrightarrow{p, m} H$ of p at a match $m : L \rightarrow G$, subject to the usual gluing conditions, is defined by a double-pushout construction. In this paper, we only consider transformations at injective matches. We assume that rule applications preserve the validity of constraints, i.e., if $G \xrightarrow{p, m} H$, then $G \in \mathcal{G}ra(TG, C)$ implies $H \in \mathcal{G}ra(TG, C)$.

Graph transformation systems can model structural transformations of molecules. Stochastic graph transformation adds the potential of representing quantitative properties of reactions, such as the inherent speed at which they occur. In a chemical system, the reaction speed is captured by the rate constant as a measure of the reactivity of the given components. Assigning rate constants (or rates for short) to rules of a graph transformation system we obtain a stochastic graph transformation system. Formally a *stochastic graph transformation system* $\mathcal{S}\mathcal{G} = (TG, C, P, \pi, \rho)$ consists of a graph transformation system (TG, C, P, π) and a function $\rho : P \rightarrow \mathbb{R}^+$ assigning to each rule name the rate of an exponential probability distribution governing the delay of application once the rule is enabled [11].

As anticipated, our approach consists in deriving from a graph transformation system the incidence matrix of a PT net. This matrix can be seen as stoichiometric matrix and, combined with the rates, gives rise to a system of ODEs. However, not all graph transformation systems can be translated into PT nets without loss of information. We first identify a class of *accountable* systems where this translation is straightforward, then show how to map a wider class of GTS into accountable ones.

Definition 1 (accountable GTS) Given a graph M , called molecular pattern, the *number of occurrences of M in a graph G* , denoted $\#G(M)$, is the number of injective graph morphisms from M to G . For each graph $G \in \mathcal{G}ra(TG, C)$ this defines a multiset $\#G : \mathcal{M} \rightarrow \mathbb{N}$.

A graph transformation system $\mathcal{G} = (TG, C, P, \pi)$ is *accountable to a set of graphs \mathcal{M}* if there exists an *accounting function* $acc : P \times \mathcal{M} \rightarrow \mathbb{Z}$ such that for all transformations $G \xrightarrow{p} H$ and $M \in \mathcal{M}$, $acc(p, M) = \#H(M) - \#G(M)$ is the (possibly negative) growth of the number of occurrences of M from G to H .

In an accountable system, the number of occurrences of each pattern in \mathcal{M} created/destroyed by each rule in R must be fixed, i.e., independent of the graph and match the rule is applied to.

Given a GTS \mathcal{G} accountable to \mathcal{M} , we can define an equivalent place-transition net with set of transitions P , set of places M and incidence matrix I given by $I(p, M) = acc(p, M)$. The challenge is therefore to derive from an existing stochastic GTS an equivalent one that satisfies the accountability property. We will do so by replacing the rules of the given system by sets of instances, obtained from the original rules by embedding into larger context.

Formally, the relation between a rule and its instantiation is described by a DPO diagram which, for this purpose, is regarded as a vertical morphism between two rule spans.

Definition 2 (DPO morphisms) Given spans $s = (L \xleftarrow{l} K \xrightarrow{r} R)$ and $s' = (L' \xleftarrow{l'} K' \xrightarrow{r'} R')$, a DPO morphism $s \xrightarrow{e} s'$ consists of a triple of graph morphisms $e_L : L \rightarrow L'$, $e_K : K \rightarrow K'$, $e_R : R \rightarrow R'$ such that the two resulting squares form a double-pushout diagram. Unless stated otherwise, we will assume that all DPO morphisms are injective. Two spans are isomorphic if there exists an isomorphism between them in the category of rule spans and DPO morphisms.

The problem is to derive an accountable system while preserving the rewrite relation generated by the GTS. We will do so by step-wise refinement of the rules of the system. In a refinement, rules are replaced by sets of instances preserving the rewrite relation, i.e., every step of the original rule $G \xrightarrow{p} H$ can be simulated by one of its instances. In terms of item 2 below, $s \xrightarrow{o} t$ is the original step and its factorisation into $s \xrightarrow{e} s' \xrightarrow{d} t$ shows s' as instance of s as well as its application to the bottom span t of the transformation.

Definition 3 (refinement) For a rule s and a set of rules S' over the same type graph, $s \triangleleft S'$ is a refinement if

1. for each $s' \in S'$ there exists a DPO morphism $s \xrightarrow{e} s'$;
2. for all DPO morphisms $s \xrightarrow{o} t$, there exists $s' \in S'$ such that o factorises over s' , that is, there are DPO morphisms $s \xrightarrow{e} s' \xrightarrow{d} t$ such that $d \circ e = o$;
3. there are no isomorphic copies, i.e., $s' \cong s'' \in S'$ implies that $s' = s''$.

Given a graph transformation system $\mathcal{G} = (TG, C, P, \pi)$ and a refinement $\pi(p) \triangleleft S'_p$ for each $p \in P$, a refined GTS $\mathcal{G}' = (TG, C, P', \pi')$ is given by

- $P' = \bigcup_{p \in P} \{p\} \times S'_p$
- $\pi'(\langle p, s' \rangle) = s'$

An accountable refinement is one where \mathcal{G}' is accountable.

Hence, rule names in P' are pairs $\langle p, s' \rangle$ of a rule name in P and an instance s' of the corresponding rule span $\pi(p)$, with the mapping π' defined as projection returning s' . Note that, in particular, $s \triangleleft \{s\}$ is a refinement of a span by itself.

In order to determine the effect of a rule on a pattern, we have to analyse all the possible ways the rule can overlap with the pattern and so either create or destroy it. Each such overlap will lead to a different instance of the rule, with a known effect on the number of occurrences of that pattern.

Definition 4 (pattern-based refinement) Given a graph transformation system $\mathcal{G} = (TG, C, P, \pi)$ and a set of patterns \mathcal{M} , we define a family of rule refinements $(\pi(p) \triangleleft S'_p)_{p \in P}$ to determine a refined system \mathcal{G}' as described in Def. 3. For all rules p such that $\pi(p)$ is not an identity, consider all diagrams of the shape shown in the left (or right) of Fig. 3 such that

- $\pi(p) = s = (L \leftarrow K \rightarrow R)$ and $M \in \mathcal{M}$;
- the vertical arrows $L \rightarrow G, K \rightarrow D, R \rightarrow H$ form an (injective) DPO morphism;
- m is injective, m and l (or m and r , resp.) are jointly surjective;
- $m(M) \not\subseteq l(L)$ (or $m(M) \not\subseteq r(R)$, resp.);
- there is no morphism n such that $g \circ n = m$ (or $h \circ n = m$, resp.)

Then, for each $p \in P$, rule refinement S'_p is obtained as a system of representatives for the set of spans $s' = (G \leftarrow D \rightarrow H)$ derived at the bottom of either the left- or the right-hand side diagram in Fig. 3. That means, S'_p contains exactly one representative s' of each isomorphism class $[s']$ of spans in S'_p .

Otherwise, if $\pi(p)$ is an identity, set $S'_p = \{\pi(p)\}$.

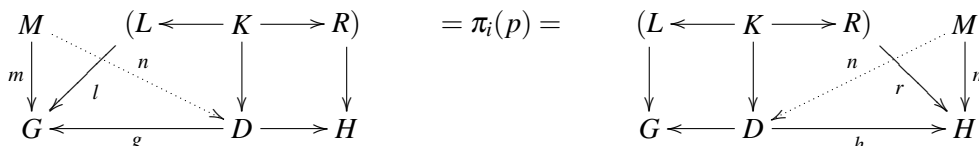


Figure 3: Rule instance consuming M (left) and rule instance creating M (right)

Such a construction can be realised using critical pair analysis as implemented in the AGG system. For this purpose, the pattern graph M can be seen as an identity rule $M \leftarrow M \rightarrow M$. The diagram in Fig. 3 on the left represents a critical pair between rule p and identity rule M , i.e., the occurrence of M is destroyed by the application of rule p . Similarly, a potential sequential dependency between rule p and identity rule M means that the occurrence of the pattern is created by the rule. In the following section we will demonstrate how, in a sample chemical reaction system, the systematic critical pair and dependency analysis between reaction rules and identity rules (patterns) yields the stoichiometric matrix needed to derive ODEs.

The following proposition guarantees that in the case of graphs representing sets of molecules, the refinement is well-defined and stepwise refinement leads to an accountable system.

Proposition 1 Assume a GTS $\mathcal{G} = (TG, C, P, \pi)$ and a set of patterns \mathcal{M} , such that graphs G satisfy the constraints C if and only if they can be presented as coproducts over copies of patterns $G = M_1 + \dots + M_n$ for $M_i \in \mathcal{M}$ (not necessarily distinct). In this case, S'_p as defined above is indeed a refinement of p . Furthermore, an accountable system is obtained after finitely many refinement steps.

Proof. First, we show that patten-based refinement is indeed a refinement in the sense of Def. 3, i.e., that the instances generated by the diagrams in Fig. 3 satisfy Def. 3 (1-3). Items 1 and 3 follow directly from Def. 4. For item 2, assume a DPO morphism $s \xrightarrow{o} t$ with $t = (X \xleftarrow{x} Y \xrightarrow{z} Z)$ and $o_L : L \rightarrow X$ the left-hand component. By assumption, $X = M_1 + \dots + M_n$ for $M_i \in \mathcal{M}$. Pick one M_i such that $M_i \cap o_L(L) \neq \emptyset$ and there is no morphism $y : M_i \rightarrow Y$ such that $x \circ y = i : M_i \rightarrow X$ is M_i 's coproduct injection. That means, M_i is destroyed by the transformation. Such a choice for M_i exists if rule p is deleting.

Let $M_i + L \xrightarrow{f_L} G \xrightarrow{d_L} X$ be the epi-mono factorisation of the unique morphism induced by the universal property of the coproduct $M_i + L$ with respect to injection $M_i \rightarrow X = M_1 + \dots + M_n$ and match $o_L : L \rightarrow X$. Composed with L 's injection, this leads to a factorisation $L \rightarrow M_i + L \xrightarrow{f_L} G \xrightarrow{d_L} X$ of o_L over G , which extends to a factorisation of the DPO morphism $o : s \rightarrow t$ over the span $s' = G \leftarrow D \rightarrow H$.

Dually, if the rule creates at least one element, we will find an M_i that is created by the transformation. By the symmetry of the DPO diagram, the same arguments yield a similar decomposition. That means, for non-identity rules, every DPO morphism can be factorised as required. For identity rules, the factorisation is given trivially over the original rule span.

To see that a finite number of refinement steps will produce an accountable system, observe that $o_L(L)$ only overlaps with finitely many M_i s. In every instance of the construction described on the left of Fig. 3, we extend L by an M_i that is not already contained in L . As an end result, all non-identity rules will be refined into the form $M_1^L + \dots + M_{n_L}^L \leftarrow M_1^K + \dots + M_{n_K}^K \rightarrow M_1^R + \dots + M_{n_R}^R$ and their applications will be disconnected from the context. That means, each application will destroy exactly the occurrences of patterns that are part of L and not in K and create those that are in R but not in K . \square

In the next section we are going to apply these notions to an example from Chemistry.

5 Case Study - the SN_1 reaction

In applying the methodology described in the preceding section, we start with a GTS consisting of general, generic reaction rules. We then refine the system such that it satisfies the accountability property. From the resulting system we derive the incidence matrix of a PT net which, at the same time, provides a representation of the stoichiometric matrix for the reaction system from which we can derive the ODEs directly.

SN_1 stands for unimolecular nucleophilic substitution. It involves the replacement of a good leaving group on a carbon atom (e.g. a chlorine atom) with a group that is less able to support negative charge (e.g. a hydroxyl group, OH). Unimolecular refers to the fact that the rate determining step involves only one molecule, the halocarbon. The reaction occurs readily once the Cl- group leaves the molecule. The reaction was chosen for its simplicity as a fundamental step in testing and demonstrating the basics of our methodology.

5.1 The Graph Transformation System

Figure 4 shows the type graph used for modelling this simple reaction. The type graph captures the necessary conditions to restrict the bonding of atoms, their valencies (the total number of bonds a particular atom is allowed to have) and any other idiosyncrasies of molecular chemistry. In Figure 4, atoms are represented as square nodes, each distinct species of reactive interest having its own node type. The round nodes are atom-specific bonding nodes. A bond between atoms is represented by an edge (arrows with filled arrowheads) between two of these bonding nodes. Each bonding node is connected to only one atom node. Each atom node has an exact number of bond nodes it can connect to, which is established using cardinalities between the two types of nodes (the numbers alongside the edges). For example, an O atom can only have two bonds. In this way, the valency of each species is established.

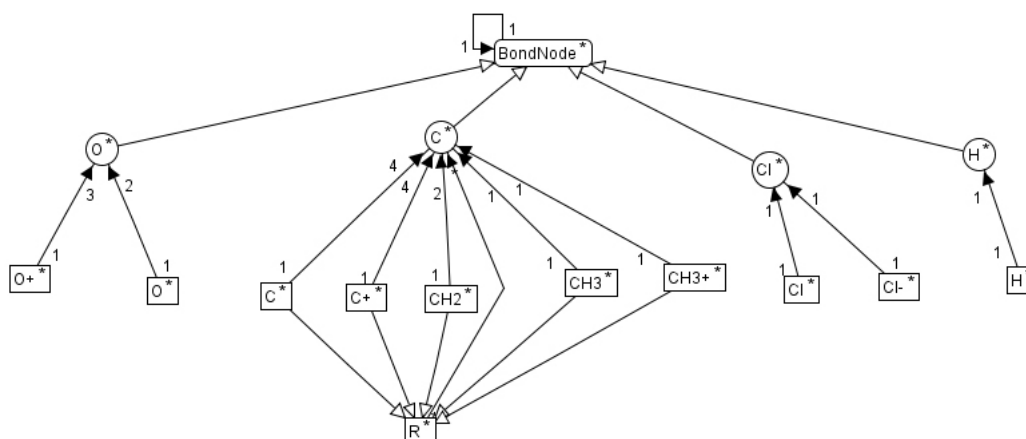


Figure 4: Type graph for SN_1 reaction, produced in AGG

The type graph also exhibits inheritance (arrows with unfilled arrow heads). All bonding nodes are subtypes of the generic *BondNode* type. This enables us to more easily specify the allowed edges between each pair of bonding node types. Inheritance is only used in the type graph and constraints, while rules are only given over types that are leaves in the hierarchy. Type graph and constraints are mechanisms to check the validity of a start graph, or a derived graph before committing the applied transformation, in much the same way that database constraints are evaluated in a transaction.

Note that C and C^+ , for example, have the same bonding node type (C) associated with them. If a reaction rule changes the charge of an atom, the atom node must first be deleted, and then a new charged one introduced in its place, so no mapping can exist between the neutral and charged atom on the left-hand side and RHS of the rule respectively. By maintaining that both node types can be connected to the same bonding node, the mapping for the bonding node of the changing atom can remain constant. This allows us to indirectly track the identity of atoms even if their charge changes throughout a reaction.

While the type graph contains C, C^+ , and H node types, our model abstracts the methyl group to a single CH_3 node type, with its charged version to CH_3^+ . Since the critical pair analysis

constructs an overlap between the graphs on the left-hand side and RHS of rules, a lower number of nodes in these graphs results in a lower number of overall overlappings, and a smaller number of nodes and edges in each overlapping. This in turn results in a quicker and more efficient analysis. A single CH_3 node, if expressed in terms of C atom nodes, H atom nodes, C bonding nodes, H bonding nodes and the edges between them, would constitute a total of 10 nodes. This abstraction of atomic structure into groups is carried out in Chemistry also, in the drawing of structural formulae, and is valid if and only if no reaction rules can be applied to the full atomic representation of the abstracted group. This can be statically checked using the analysis technique described in the previous section, i.e., creating a molecular identity rule from the full atomic representation and analysing whether there are any critical overlappings with any of the reaction rules. If there are, any abstraction of this group which affects the presence of the critical pairs is invalid.

In addition to the type graph, constraints may be used to specify conditions on graphs (and rules) within the system. As edges are directional, a convention is needed to ensure that edges between a specified pair of atoms are always facing the same direction. To this end, it was decided that edges would go from the less electronegative atom to the more electronegative one e.g. for a bond between C and O, the edge would always go from the C bonding node to the O bonding node. A further constraint was necessary to prevent multiple incoming edges to the carbon/hydrocarbon and oxygen round bond nodes. So that all of the carbon/hydrocarbon bonding possibilities need not be enumerated for the constraint, the type graph utilises inheritance to create an R supertype for all carbon/hydrocarbon node types. The constraint then only requires a single combination, between 2 R nodes. Finally, a constraint was needed to prevent CH_3^+ from being able to bond to anything. Figure 5 shows all five of these constraints, with “not” representing the fact that graphs containing these structures should be considered invalid.

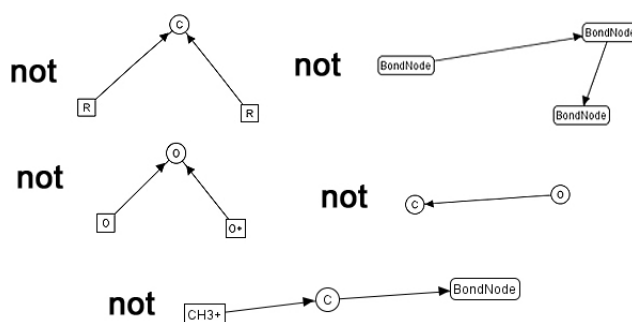


Figure 5: Additional graph constraints

Note that we have used negative constraints in our model to restrict the behaviour of bond nodes. If the overlap of M and L in the consumption evaluation described in the previous section (or M and R in the production evaluation) creates a graph G or H respectively that violates one of these negative constraints, we can deduce that the overlap is invalid and can be discarded, since the overlap is a minimal union of the two graphs. Any violation here is still present if this union were embedded into a larger graph. If the negative constraint is not violated, the overlap is maintained as valid. A positive constraint, however, specifies the required presence of a certain

configuration of nodes and edges somewhere in the graph. If violated in the minimal union, it may still be fulfilled once the union is embedded into a larger graph. Positive constraints are therefore avoided in our graph representation (or if they were present, would not impact on the analysis).

An example of a graph typed over the type graph in Figure 4 is given in Figure 6. These are the starting materials for the S_N1 reaction, namely water and chloromethane.

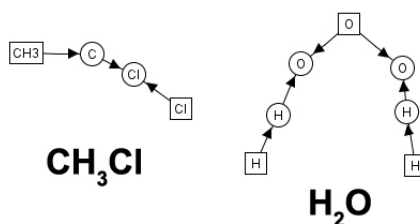


Figure 6: Graph depicting starting materials for S_N1 reaction, produced in AGG. Left: CH_3^+ , Right: H_2O

The first step in our methodology is to capture all possible general rules for a reaction system. This involves the definition of reactions at the level of functional groups only, i.e., local molecular context. Figure 7 depicts the general rules governing the reactivity for this small reaction mechanism. The top rule dictates that if a chloromethane molecule is found in a graph, it can be broken to form charged carbocation (CH_3^+) and Cl^- ions. The middle rule captures the attack of the positively charged carbocation by a lone pair of electrons on the O of an OH group. In the lower rule, the positively charged O^+ accepts electrons from the O^+-H bond in the presence of the Cl^- created in the very first step of the reaction. This yields hydrochloric acid (HCl), and an alcohol product, containing an OH group. The reverse of each of these three rules were also added as potential reactions (by simply reversing the left-hand side and RHS of the rule). These reverse reactions may be unlikely but this will eventually be designated by a negligible rate constant for that particular reaction. For generality and a complete specification of possible activity, it is important to include them.

We formulate the two known starting materials in Figure 6 as two molecular identity rules, where the left-hand side and RHS are the same and contain only the graph of a particular molecule. The first step towards making our GTS into an accountable one is to generate the full set of places (i.e. molecular species) that would appear in the PT net translated from our accountable GTS. To discover unknown intermediates and products in the reaction, a critical pair analysis is conducted between each general reaction rule and each molecular identity rule. This is an iterative process as described in Section 4, which eventually gives the set of graphs \mathcal{M} , to which our GTS must be accountable. The results of the first iteration are given in Table 2. Each entry signifies how many of the overlappings were critical for each pair, remembering that critical pair analysis checks all possible unions of L and M for parallel conflict analysis, and R and M for sequential dependence analysis.

There are 2 critical overlappings wherever there is a conflict with H_2O . Due to the nature of our molecular representation, some of the critical overlappings returned may be spurious over-

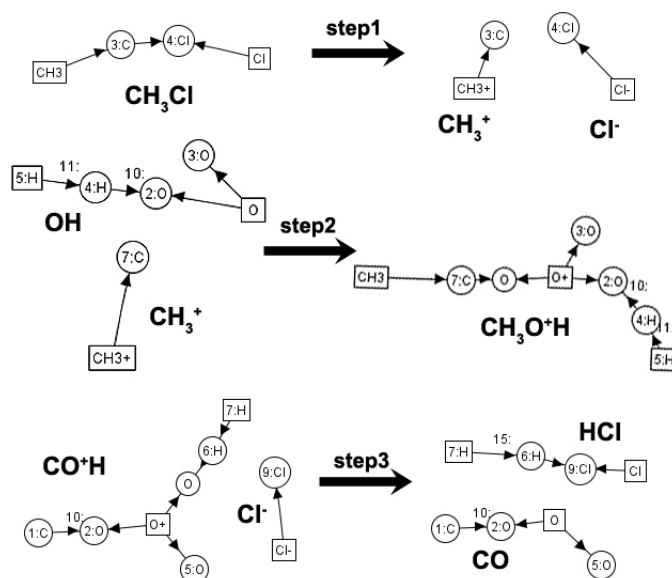


Figure 7: Steps 1 (top), 2 (middle) and 3 (bottom) of SN₁ reaction, general reaction rules

	CH ₃ Cl		H ₂ O	
	PC	SD	PC	SD
Step1	1	0	0	0
Step-1	0	1	0	0
Step2	0	0	2	0
Step-2	0	0	0	2
Step3	0	0	0	0
Step-3	0	0	0	0

Table 2: Results of first pass critical analysis - parallel conflicts (PC) and sequential dependencies (SD)

lappings. One critical overlapping of the Step2-H₂O rule pair is given in Figure 8. The shaded grey area covers all of the critical nodes and edges in this overlapping. The other critical overlapping is structurally identical in that the types of the edges and nodes covered are the same. The two overlappings arise, however, due to the symmetry around the critical O atom node (with mapping id 1). Chemically, however, these configurations have no significance to the selection of a reaction or to the outcome of the reaction. We can treat them as equivalent, formally captured by the obvious notion of isomorphism between the corresponding transformations. Therefore this entry in the table can be reduced to 1. To evaluate two overlappings for chemical equivalence, the results of applying the reaction rule to the overlappings at their respective critical matches can be compared. If the results are isomorphic, the two overlappings can be considered to be chemically equivalent, since they react in the same way to give the same products up to isomorphism. For the SN₁ reaction studied, the molecules involved were small, possessing only

one reactive site per instantiated reaction, therefore in all cases these overlappings were reduced to 1. If there were two asymmetric reactive sites in a molecule for an elementary reaction step, there would be 2 chemically distinct overlappings, since application of the reaction rule would lead to alternative products. In this case, we would have discovered two possible instantiations of the same general reaction rule.

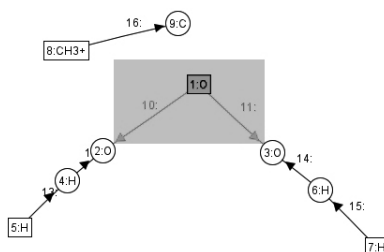


Figure 8: One critical overlapping between general reaction rule Step2, and H₂O (critical graph elements are contained within the shaded area)

New intermediates in the system can be discovered by the systematic application of the general rule to the molecules it has an impact on. Where a parallel conflict is apparent, the application of the general rule to that molecule at the match given by the critical overlapping gives us a chemical species that is produced by this reaction. Similarly, for sequential dependencies, the application of the reverse of the general rule to the molecule, gives us the species that undergoes this general reaction to give the molecule in the pattern graph in question. For example, new molecules arising from the critical overlappings in Table 2 are Cl⁻ and CH₃⁺. We add these new molecules as molecular identity rules and repeat the critical pair analysis as these new intermediates may produce further intermediates under the rule-defined behaviour of the system. After five iterations of the critical pair analysis, the total set of chemical species for this case study can be derived as molecular identity rules. The graphs representing these intermediates are given in Figure 9, along with their chemical formulae. The complete set of these is then the set \mathcal{M} to which our graph transformation system must be accountable, as described in Section 4. The result of the last iteration, of the critical pair analysis is given in Table 3.

	CH ₃ Cl		H ₂ O		CH ₃ ⁺		Cl ⁻		CH ₃ O ⁺ H ₂		HCl		CH ₃ OH		CH ₃ O ⁺ HCH ₃		CH ₃ OCH ₃		
	PC	SD	PC	SD	PC	SD	PC	SD	PC	SD	PC	SD	PC	SD	PC	SD	PC	SD	
Step1	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
Step-1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Step2	0	0	2	0	1	0	0	0	0	2	0	0	1	0	0	2	0	0	0
Step-2	0	0	0	2	0	1	0	0	2	0	0	0	0	1	2	0	0	0	0
Step3	0	0	0	0	0	0	1	0	2	0	0	1	0	1	2	0	0	0	2
Step-3	0	0	0	0	0	0	0	1	0	2	1	0	1	0	0	2	2	0	0

Table 3: Results of iterative critical pair analysis - parallel conflicts (PC) and sequential dependencies (SD)

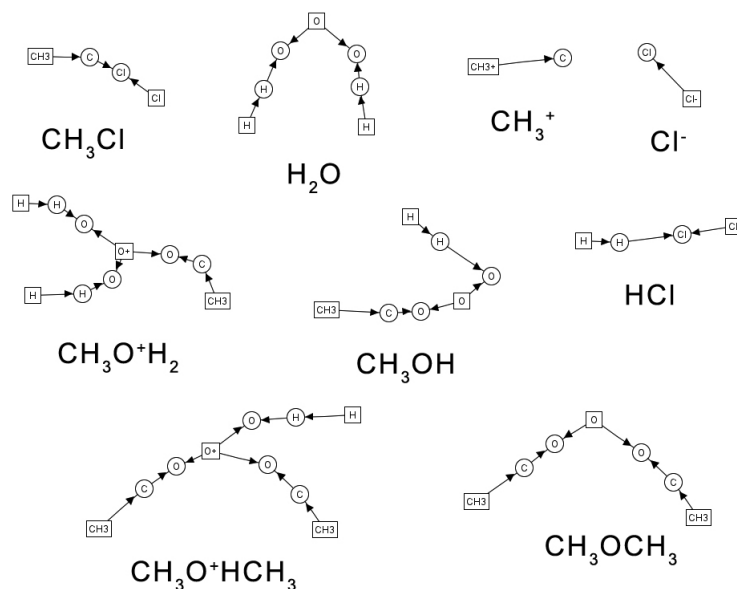


Figure 9: Complete set of chemical species for S_N1 reaction, derived through iterations of critical pair analysis

5.2 From Graph Transformation System to Place Transition net

To achieve the accountability condition necessary to translate our GTS to a PT net, we need to instantiate the general rules using the results of Table 3. The reduction of chemically equivalent overlappings explained above becomes important at this stage of the refinement. Two instantiations of a general rule based on two chemically equivalent overlappings would be span isomorphic. As such, we only need to add one of the two instantiations from this equivalence class to the refined graph transformation system as per the formalism in Section 4. Reducing the values in Table 3 to chemically differentiable critical overlappings using the analysis described above aids in simplifying the identification of these equivalence classes.

An interesting general reaction in our case study is that of *Step2*. There are entries in the parallel conflicts part of the table for H_2O , CH_3^+ and CH^3OH . Closer manual examination reveals that this encapsulates two reactions; the reaction of H_2O with CH_3^+ , and a secondary reaction of the methanol product, CH^3OH , with CH_3^+ . This general rule can therefore be instantiated twice, as per Figure 10. This analysis can be carried out methodically for an even larger number of instantiation candidates than just the three we have here. Essentially, the overlap of M_1 with L in Figure 3 forms a graph into which a second overlap with another molecular pattern graph, M_2 , can take place directly. The order in which the overlaps are constructed is inconsequential. This is possible since the type graph cardinalities ensure that molecules are always defined in their entirety. As such, there is no possible intersection of graph elements between the two separate molecules in the combined overlap, unless the two molecules are identical (in which case the overlap is meaningless and discarded).

To identify whether a pair of molecules are indeed valid instantiation candidates, we can ex-

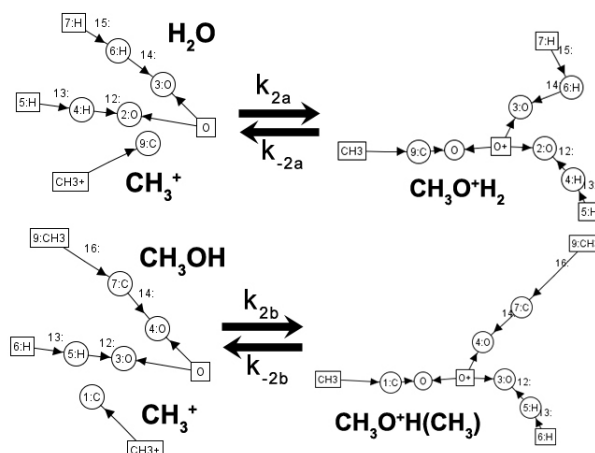


Figure 10: Instantiation of step2 for reaction with water (top) and methanol (bottom)

amine the construction in Figure 11. M_1 forms a union with L to give G_1 , while M_2 forms a union with L to give G_2 . G_1 and G_2 can be combined to give a final overlap, which is used to instantiate the rule fully. Since l_1 and l_2 identify the two different reactivity sites needed for the reaction, for a valid combination of G_1 and G_2 , their intersection should be empty. Formally, this condition is represented as:

$$l_1^{-1}(m_1(M_1)) \cap l_2^{-1}(m_2(M_2)) = \phi$$

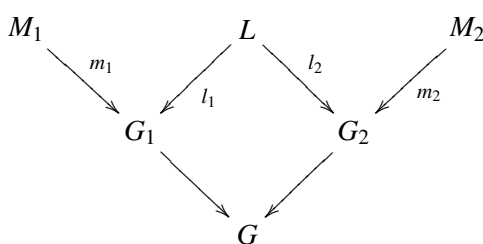
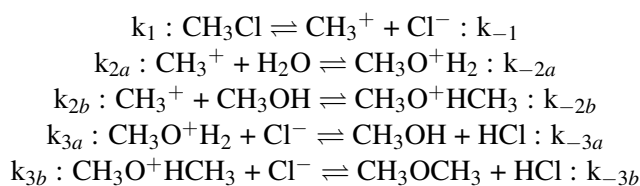


Figure 11: Combination of critical overlappings for instantiation of general rules

For each unique, valid set of instantiation species, we instantiate the general rule to include the full molecules in its context. All instantiated reaction rules are then named with rate constants, specific to the reaction they refer to. For this case study, step3 can also be instantiated twice to mark the deprotonation of $\text{CH}_3\text{O}^+\text{H}_2$, and of $\text{CH}_3\text{O}^+\text{HCH}_3$. Finally then, k_1 refers to the first step of the reaction (the Cl^- leaving the chloromethane molecule), k_{2a} is the attack of water on the resulting carbocation, k_{2b} the attack of methanol on the same carbocation, k_{3a} the deprotonation of the result of the water attack, and k_{3b} the deprotonation of the result of the methanol attack. The reverse reactions are represented by k_{-x} for each of these forward reactions.

Finally, we have a refined graph transformation system, satisfying the accountability property with respect to the set of molecules discovered through several iterations of the critical pair

analysis. With the complete set of instantiated rules and their impact on each molecular pattern graph, we can build the incidence/stoichiometric matrix for the PT net for this reaction (see Table 4) by subtracting the total parallel conflict critical overlappings (how many molecules are consumed) from the total sequential dependence critical overlappings (how many molecules are produced) for a particular instantiated reaction molecular pattern graph pair. This then gives us the impact each elementary reaction has on each chemical species. To reiterate, the reactions that the entries in the stoichiometric matrix refer to are given below:



	CH ₃ Cl	H ₂ O	CH ₃ ⁺	HCl	Cl ⁻	CH ₃ O ⁺ H ₂	CH ₃ OH	CH ₃ O ⁺ HCH ₃	CH ₃ OCH ₃
k₁	-1	0	1	0	1	0	0	0	0
k₋₁	1	0	-1	0	-1	0	0	0	0
k_{2a}	0	-1	-1	0	0	1	0	0	0
k_{-2a}	0	1	1	0	0	-1	0	0	0
k_{2b}	0	0	-1	0	0	0	-1	1	0
k_{-2b}	0	0	1	0	0	0	1	-1	0
k_{3a}	0	0	0	1	-1	-1	1	0	0
k_{-3a}	0	0	0	-1	1	1	-1	0	0
k_{3b}	0	0	0	1	-1	0	0	-1	1
k_{-3b}	0	0	0	-1	1	0	0	1	-1

Table 4: Stoichiometric matrix resulting from final pass critical pair analysis and chemical equivalence analysis

5.3 ODEs from the PT net

ODE extraction from the results in Figure 4, following the procedure outlined at the beginning of Section 4 led to the ODEs shown in Figure 12. These results agree with ODEs derived manually for this simple system using the law of mass action.

5.4 Tool support

AGG [19] is a widely used graph transformation tool that was crucial to our implementation. It allows the construction of a graph grammar, namely type graph, rewriting rules, start graph and constraints. AGG also has its own critical pair analysis (CPA) engine, accessible through GUI and command line. The GUI gives a summary table similar to Table 3 and a graph for each critical overlap, specifying which graph elements are critical and their identities in each of the two rules.

From Graph Transformations to Differential Equations

$$\begin{aligned} d[\text{CH}_3^+]/dt = & -k-1[\text{CH}_3^+][\text{Cl}^-] +k-2a[\text{CH}_3\text{O}+\text{H}_2] +k-2b[\text{CH}_3\text{O}+\text{HCH}_3] \\ & +k1[\text{CH}_3\text{Cl}] -k2a[\text{CH}_3^+][\text{H}_2\text{O}] -k2b[\text{CH}_3^+][\text{CH}_3\text{OH}] \end{aligned}$$

$$d[\text{CH}_3\text{Cl}]/dt = +k-1[\text{CH}_3^+][\text{Cl}^-] -k1[\text{CH}_3\text{Cl}]$$

$$\begin{aligned} d[\text{CH}_3\text{O}+\text{H}_2]/dt = & -k-2a[\text{CH}_3\text{O}+\text{H}_2] +k-3a[\text{CH}_3\text{OH}][\text{HCl}] +k2a[\text{CH}_3^+][\text{H}_2\text{O}] \\ & -k3a[\text{CH}_3\text{O}+\text{H}_2][\text{Cl}^-] \end{aligned}$$

$$\begin{aligned} d[\text{CH}_3\text{O}+\text{HCH}_3]/dt = & -k-2b[\text{CH}_3\text{O}+\text{HCH}_3] +k-3b[\text{CH}_3\text{OCH}_3][\text{HCl}] \\ & +k2b[\text{CH}_3^+][\text{CH}_3\text{OH}] -k3b[\text{CH}_3\text{O}+\text{HCH}_3][\text{Cl}^-] \end{aligned}$$

$$d[\text{CH}_3\text{OCH}_3]/dt = -k-3b[\text{CH}_3\text{OCH}_3][\text{HCl}] +k3b[\text{CH}_3\text{O}+\text{HCH}_3][\text{Cl}^-]$$

$$\begin{aligned} d[\text{CH}_3\text{OH}]/dt = & +k-2b[\text{CH}_3\text{O}+\text{HCH}_3] -k-3a[\text{CH}_3\text{OH}][\text{HCl}] -k2b[\text{CH}_3^+][\text{CH}_3\text{OH}] \\ & +k3a[\text{CH}_3\text{O}+\text{H}_2][\text{Cl}^-] \end{aligned}$$

$$\begin{aligned} d[\text{Cl}^-]/dt = & -k-1[\text{CH}_3^+][\text{Cl}^-] +k-3a[\text{CH}_3\text{OH}][\text{HCl}] +k-3b[\text{CH}_3\text{OCH}_3][\text{HCl}] \\ & +k1[\text{CH}_3\text{Cl}] -k3a[\text{CH}_3\text{O}+\text{H}_2][\text{Cl}^-] -k3b[\text{CH}_3\text{O}+\text{HCH}_3][\text{Cl}^-] \end{aligned}$$

$$d[\text{H}_2\text{O}]/dt = +k-2a[\text{CH}_3\text{O}+\text{H}_2] -k2a[\text{CH}_3^+][\text{H}_2\text{O}]$$

$$\begin{aligned} d[\text{HCl}]/dt = & -k-3a[\text{CH}_3\text{OH}][\text{HCl}] -k-3b[\text{CH}_3\text{OCH}_3][\text{HCl}] +k3a[\text{CH}_3\text{O}+\text{H}_2][\text{Cl}^-] \\ & +k3b[\text{CH}_3\text{O}+\text{HCH}_3][\text{Cl}^-] \end{aligned}$$

Figure 12: Result of ODE extraction for SN_1 reaction

AGG is open source and is implemented in Java with its own API. Adaptation of the standard implementation allowed us to schedule only relevant pairs for critical pair computation. Standard analysis would analyse every rule with every other rule. Since we only need to examine critical overlappings between reaction rules and identity rules this was a first step towards a more efficient implementation. Methods and classes from the API were also used for the Java component that performed the chemical equivalence reduction described in Section 5. CPA saves results in an XML file that can be reloaded using API methods. The ODE extraction module was also implemented with the aid of these methods. Finally, a script was created which piped the results of the CPA through the chemical equivalence program and ODE extraction program in turn. Currently, the overall program produces the ODEs as a text file, but the output could easily be changed to LaTeX syntax, or a format recognized by a math solver for example. Figure 13 shows the basic architecture of the tool chain used.

It is worth noting that the abstraction of the CH_3 group to a single node, described in Section 5, was prompted by implementation issues for the fully detailed atomic representation. It is obvious that the critical pair analysis requires more resources the larger the rules become, since the overlap that must be constructed is larger (meaning more checks to be made for critical graph elements). Also, it is likely there will be more possible overlappings overall. Every CH_3 group represented as C, H atom nodes and C, H bonding nodes, with edges between them introduces 9 extra nodes and 9 extra edges to the graph. This early implementation, after several hours, eventually ran out of memory during CPA (assigning 1.5GB to the Java Virtual Machine), with the largest number of overlappings to check for any one pair of rules exceeding 35,000. There were also far more spurious, chemically equivalent overlappings for every critical pair, due to the extra symmetry and permutability introduced by the additional nodes and edges. In the abstracted

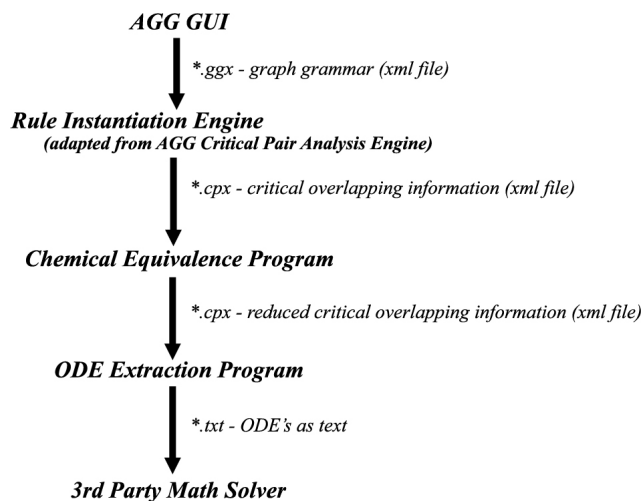


Figure 13: Basic tool chain architecture (arrow labels show input and output file extensions)

case, this number was reduced to just over 11,000 for any pair and the instantiation round of the critical pair analysis along with critical overlapping structural equivalence testing and ODE extraction took less than 8 minutes (assigning 1.0GB to the Java Virtual Machine). While a more general model with a lower level of detail would have been an interesting representation to consider, it was equally important to use valid abstraction techniques to ensure a workable implementation.

6 Conclusion

This paper has presented a graph transformation approach to defining molecular reactions that retains much of the visual notation adopted by Chemists themselves. Using a variant of critical pair analysis, a system of reaction rules presented in this form can be refined into one that can be expressed by a place-transition net. A set of ODEs describing the overall reaction dynamics can be extracted from the incidence matrix of that PT net. The approach was demonstrated by means of a simple SN_1 reaction. The methodology and implementation are based on existing theory and tools for graph transformation systems, predominantly that of constraints, critical pairs, and isomorphism of graphs and rules.

Future directions for research include the study of unbounded systems such as polymerisation and free radical reactions. Currently, our methods require manual observation to derive intermediate reaction results. For reactions where the size and number of molecular patterns is unbounded, it is imperative to be able to automate the generation of molecular patterns as well as to allow for incompleteness leading to approximated solutions. To evaluate the method, ODE extraction should be carried out alongside stochastic model checking and simulation, to determine how the results compare and complement each other.

Bibliography

- [1] Bapodra, M., (2009): *Chemical Reaction Rate Analysis Using Graph Transformations*. BSc Computer Science with Management Project, Final Report Available at <http://www.cs.le.ac.uk/people/mb294/docs/BScFinalReport.pdf>.
- [2] Berry, G., Boudol, G., (1989): *The Chemical Abstract Machine*. Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages pp. 81–94 Available at <http://portal.acm.org/citation.cfm?id=96717>.
- [3] Benko, G., Flamm, C., Stadler, P.F., (2003): *A Graph-Based Toy Model of Chemistry*. *Journal of Chemical Information and Computer Sciences* 43, pp. 1085–1093. Available at <http://pubs.acs.org/doi/abs/10.1021/ci0200570>.
- [4] Cardelli, L., (2008): *From Processes to ODEs by Chemistry*. *IFIP International Federation for Information Processing* 273, pp. 261–281. Available at <http://www.springerlink.com/content/8g12ph2857372xm2/>.
- [5] Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J., (2007): *Rule-based modelling of cellular signalling*. 18th International Conference on Concurrency Theory, Lisbon, Portugal, September, 2007, *Lecture Notes in Computer Science* 4703, pp. 17–41. Available at <http://www.pps.jussieu.fr/~danos/>.
- [6] Ehrig, H., Ehrig, K., Prange, U., Taentzer, G., (2006): *Fundamentals of Algebraic Graph Transformation*. *EATCS Monographs*, Springer
- [7] Ehrig, K., Heckel, R., Lajios, G., (2006): *Molecular Analysis of Metabolic Pathways with Graph Transformation*. *International Conference on Graph Transformations 2006*, *Lecture Notes in Computer Science* 4178, pp. 107–121. Available at <http://www.springerlink.com/content/6r637jx517320v02/>.
- [8] Faeder, J.R., Blinov, M.L., Hlavacek, W.S., (2009): *Rule-Based modelling of Biochemical Systems with BioNetGen*. *Methods in Molecular Biology, Systems Biology* 500, pp. 113–167. Available at <http://www.springerlink.com/content/nn1v343vw2kmj00w/>.
- [9] Feret, J., Danos, V., Krivine, J., Harmer, R., Fontana, W., (2009): *Internal coarse-graining of molecular systems*. *Proceedings of the National Academy of Sciences of the United States of America* 106(16), pp. 6453–6458. Available at <http://www.pnas.org/content/106/16/6453>.
- [10] Gilbert, D., Heiner, M., (2006): *From Petri Nets to Differential Equations - An Integrative Approach for Biochemical Network Analysis*. *ICATPN 2006*, *Lecture Notes in Computer Science* 4024, pp. 181–200. Available at <http://www.springerlink.com/content/18516147038t012j/>.
- [11] Heckel, R., Lajios, G., Menge, S., (2004): *Stochastic Graph Transformation Systems*. *International Colloquium on Theoretical Aspects of Computing 2005*, *Lecture Notes in Computer Science* 3256, pp. 210–225. Available at <http://iospress.metapress.com/content/c7ha18g96nbm7g2e/>.

- [12] Heckel, R., (2006): *Graph Transformation in a Nutshell*. *Electronic Notes in Theoretical Computer Science* 148, pp. 187–198.
- [13] Heckel, R., (2005): *Stochastic Analysis of Graph Transformation Systems: A Case Study in P2P Networks*. *International Colloquium on Theoretical Aspects of Computing 2005, Lecture Notes in Computer Science* 3722, pp. 53–69. Available at <http://www.springerlink.com/content/u54h1w273875u67r/>.
- [14] Hillston, J., (2005): *Fluid Flow Approximation of PEPA models*. *Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, IEEE Computer Society Press* pp. 33–43. Available at <http://www.dcs.ed.ac.uk/pepa/features/>.
- [15] Keeler, J., Wothers, P., (2008): *Chemical Structure and Reactivity, An Integrated Approach*, Oxford: Oxford University Press.
- [16] Khan, A., Torrini, P., Heckel, R., (2008): *Model-based Simulation of VoIP Network Reconfigurations using Graph Transformation Systems*, *ECEASST 16*, Available at <http://eceaast.cs.tu-berlin.de/index.php/eceaast/article/view/251>.
- [17] Palacz, W., (2008): *Practical Aspects of Graph Transformation Meta-Rules*. *Intelligent Computing in Engineering* pp. 70–77. Available at <http://www.eg-ice.org/Workshops/ICE08/papers%20pdf/P002V03.pdf>.
- [18] Rossello, F., Valiente, G., (2005): *Graph Transformation in Molecular Biology*. *Formal Methods (Ehrig Festschrift), Lecture Notes in Computer Science* 3393, pp. 116–133. Available at <http://iospress.metapress.com/content/c7ha18g96nbm7g2e/>.
- [19] <http://tfs.cs.tu-berlin.de/agg/>: *The AGG Homepage*. Last Visited October 2009.