50 Shades of Rule Composition

From Chemical Reactions to Higher Levels of Abstraction

Jakob L. Andersen[1], Christoph Flamm[2], Daniel Merkle[1], Peter F. Stadler[2−7]

[1] Department of Mathematics and Computer Science
University of Southern Denmark, Denmark.
{daniel,jlandersen}@imada.sdu.dk
[2] Institute for Theoretical Chemistry, University of Vienna, Austria.
xtof@tbi.univie.ac.at
[3] Bioinformatics Group, Department of Computer Science, and Interdisciplinary
Center for Bioinformatics, Leipzig, Germany.
[4] Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany.
[5] Fraunhofer Institute for Cell Therapy and Immunology, Leipzig, Germany.
[6] Center for non-coding RNA in Technology and Health
University of Copenhagen, Denmark.
[7] Santa Fe Institute, USA.
studla@bioinf.uni-leipzig.de

Abstract Graph rewriting has been applied quite successfully to model chemical and biological systems at different levels of abstraction. A particularly powerful feature of rule-based models that are rigorously grounded in category theory, is, that they admit a well-defined notion of rule composition, hence, provide their users with an intrinsic mechanism for compressing trajectories and coarse grained representations of dynamical aspects. The same formal framework, however, also allows the detailed analysis of transitions in which the final and initial states are known, but the detailed stepwise mechanism remains hidden. To demonstrate the general principle we consider here how rule composition is used to determine accurate atom maps for complex enzyme reactions. This problem not only exemplifies the paradigm but is also of considerable practical importance for many down-stream analyses of metabolic networks and it is a necessary prerequisite for predicting atom traces for the analysis of isotope labelling experiments.

1 Introduction

Abstract rule based systems with roots in process algebras and concurrency have been introduced to formalize biological processes more than a decade ago starting with Fontana's models of evolving constructive in λ-calculus [1] and Regev's view on "cells as computation" [2]. Kappa [3], for instance, was designed to model the behaviour of mixtures of "agents" (usually thought of as interacting proteins) using rules that describe changes to the agents' internal states. Conceptually similar approaches have been taken in BioNetGen [4]. Compartmentalization and the relative placement of cellular components is the focus of "membrane computation" [5] or the brane calculus [6]. A term-rewriting-like model has been proposed to account for the computational aspects of epigenetics [7]. For reviews see also [8,9].

Chemistry has motivated the "Chemical Abstract Machine" [10] as model of concurrent computation already in 1990, and graph representations of molecules have been used throughout the entire history of organic chemistry. Concrete models of chemistry in terms of graph rewrite systems, however, have appeared only after the turn of the millenium [11] and a systematic investigation into the practical advantages provided by the rich formal structure of graph rewriting systems is even younger. In this context it is interesting to note that Kappa has an intuitive graphical interpretation as single pushout (SPO) rewriting system on a category of suitably annotated graphs known as Σ-graphs [12]. In the context of chemistry, the more restrictive double pushout (DPO) framework appears to have some advantages. It ensures, e.g., the reversibility of chemical reactions.

A key tool in rule-based calculi is the concept of rule composition. It allows, in particular, different levels of coarse graining in the description of a system's trajectories by contracting transitions between states in a principled manner, explored in some detail for Kappa in [12]. A natural application of the same idea in the realm of chemistry is to relate elementary reactions with overall reactions composed of multiple sequential steps. Here we show rule composition is a useful avenue into disentangling the mechanistic details of multi-step transitions between a known initial and finite state. A practical problem from computational chemistry will serve as an application showcase in the following. Similar analyses could likely be useful, e.g., in generative models of animal development [13]. Neither data nor rules sets are available in the form of a publicly accessible data base, however.

The atom map of a chemical reaction specifies which atom of the product molecules corresponds to the which of the educt atoms. This type of detailed information is in most cases not available in chemical reaction databases, so that it must be reconstructed computationally from the known educts and products only. In general, there is more than one plausible atom map and elaborate experimental techniques marking individual atoms in the educts by rare isotopes are necessary to distinguish between different possibilities. The MACiE database in addition provides information on the reaction mechanisms that are catalysed by the enzymes involved in the overall reaction [14,15]. We show here how these rules together with the knowledge of start and end state can be used to (nearly) uniquely determine the step-wise reaction mechanism. To this end we employ the formal framework of rule composition. Conversely, the alternative atom maps identify variant reaction mechanism that can be disentangled by suitable isotope labelling experiments.

The paper is structured as follows. In Section 2 we will introduce the Double Pushout formalism. Graph grammar rule composition operators will be defined and exemplified by composition of chemical reactions. In Section 3 rule composition will be employed in order to change the level of abstraction for three different chemical systems. Furthermore, full atom traces will be computed. Conclusions are given in Section 4.

2 Methods

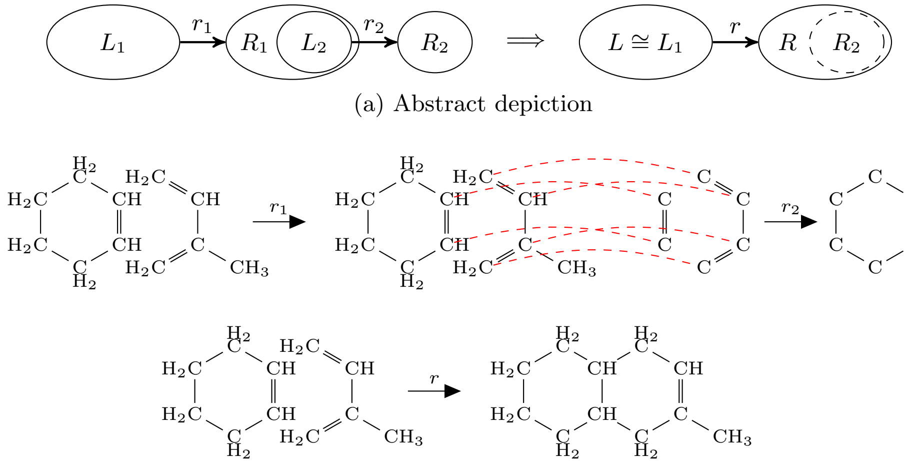
2.1 Rule Composition

We model molecules as labelled simple graphs. Vertex labels denote atom types and edge labels indicate bond types. Chemical reactions are described by graph transformation rules in the Double Pushout (DPO) formalism that encode specific reactions mechanisms [11]. Each rule has the form $(L \xleftarrow{l} K \xrightarrow{r} R)$, where L and R are the left and right graph and K is the context graph glueing the transformation of L into R. A rule is applied to a graph G by finding a subgraph of G isomorphic to L and using the morphisms l and r to remove $L \backslash K$ from G and adding $R \backslash K$ to it, resulting in the transformed graph H. A chemical transformation rule has the special property that no atom (vertex) can be added, removed, or relabelled so that all atoms are represented in K and the restrictions r_V and l_V of the morphisms r and l to the vertex sets are bijective. Thus $\alpha = l_V \circ r_V^{-1}$ is a bijection between the vertex sets of L and R, which encodes the atom-map of the reaction mechanism. The full atom-map is obtained by extending α by the identity on the parts of G that remain unchanged in H. For a more technical description we refer to [16].

In [16] we also described in detail how transformation rules can be composed in a chemically relevant manner. Here we introduce new composition operators and give a conceptual and less formal overview of the different types of compositions and their usage. We use $\circ$ to denote composition. In contrast to the usual notation we read compositions from left to right, i.e., $r_1 \circ r_2$ is the composition which applies r_1 first and then r_2. Two rules $r_1 = (L_1 \leftarrow K_1 \rightarrow R_1)$, $r_2 = (L_2 \leftarrow K_2 \rightarrow R_2)$ are composed as $r_1 \circ_m r_2$ using a partial isomorphism m between R_1 and L_2 (note that not all partial isomorphisms m induce a valid composition). In the following we will describe types of composition for different levels of generality classified by the structure of the match m. Let $r_1 \circ_{\supseteq} r_2$ denote a composition where the match specifies that R_1 is a super-graph of L_2. Fig. 1 illustrates this case, which is analogous to the application of r_2 to the graph R_1.

Molecules correspond to connected components of molecule graphs, and connected components of the graphs in a transformation rule thus correspond to the possibility of merging and splitting molecules. The composition with a super-graph isomorphism can be generalised to a partial component-wise super-graph relation as described in [16]. In such a composition $r = r_1 \circ^{c}_{\supseteq} r_2$, illustrated in Fig. 2, we only require a subset of the connected components of L_2 to be defined by the matching morphism. These components, however, still must be be completely defined. The semantics of this class of composition is analogous to partial function application in programming languages.

At the most general level we consider compositions without restriction on the match. We denote these by $\circ_{\cap}$ as the match specifies a common subgraph of R_1



(a) Abstract depiction

(b) Chemical example

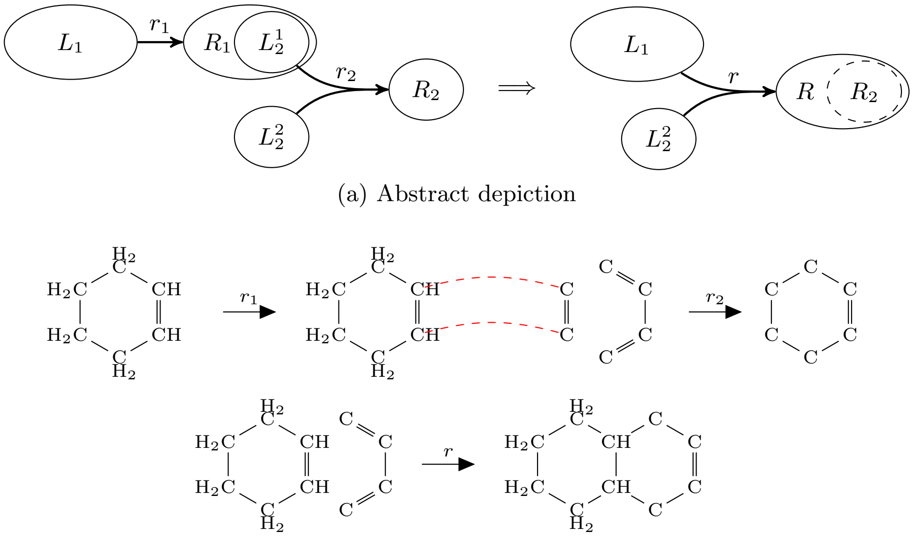
Fig. 1: A composition $r = r_1 \circ_{\supseteq} r_2$ with the matching morphism being a super-graph isomorphism of R_1 to L_2. The context graphs are omitted from the drawings for simplicity. (a) Abstract depiction; L_2 is isomorphic to a subgraph of R_2. (b) Chemical example; $r_1 = (G, G, G)$ is the identity rule for a graph G encoding the educts cyclohexene and isoprene. The second rule, r_2, is the reaction template for the Diels-Alder reaction. The composed rule therefore encodes the overall rule of the Diels-Alder reaction on the input molecules.

and L_2 (see Fig. 3 for an example). The class of chemically valid transformation rules is not closed under this type of composition. For instance, valence restrictions may be violated. As a special case we consider the composition where the common subgraph of R_1 and L_2 is empty (denoted by $\circ_{\emptyset}$). The resulting composed rule encodes the parallel application of the operand rules, as illustrated in Fig. 4.

In the following sections we will primarily use compositions with the supergraph relation, $\circ_{\supseteq}$, and we will therefore simply use $\circ$ to denote these compositions, which we refer to as "full composition". The more relaxed composition, $\circ^{c}_{\supseteq}$, will be referred to as "partial composition". Implicitly, we use parallel composition, $\circ_{\emptyset}$, for constructing identity rules for multiple molecules, and it will be explicitly used for the carbon-tracing in the glycolysis pathway.

2.2 Implementation

The rule composition framework is implemented in C++11 as a part of a library that is primarily aimed at chemical graph transformation and thus includes special features and optimizations for molecules (e.g., use of canonical SMILES strings for graph isomorphism [17,18]). It is, however, not restricted to the domain of chemistry and can be used readily to handle graph grammar models at other levels of abstraction. Python bindings for the library gives a more acces-



(a) Abstract depiction

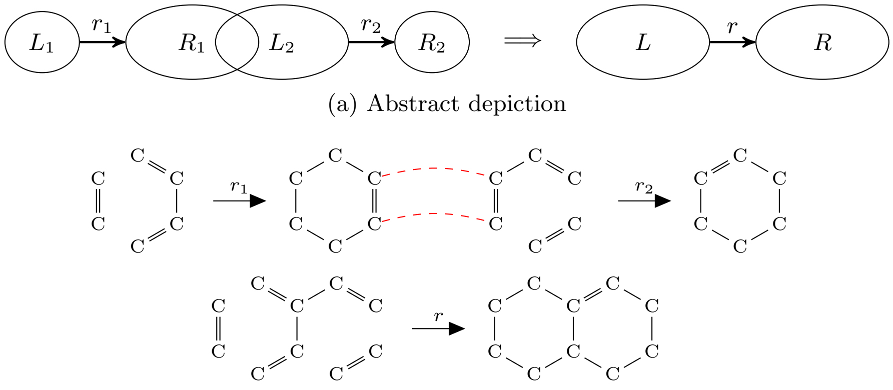
(b) Chemical example

Fig. 2: A composition $r = r_1 \circ^{c}_{\supseteq} r_2$ with the matching morphism being a super-graph isomorphism of R_1 to a non-empty subset of the connected components of L_2. The context graphs are omitted for simplicity. (a) Abstract depiction; connected components of L_2 are either completely unmatched or completely matched. (b) Chemical example; r_1 is the identity rule cyclohexene and r_2 the the Diels-Alder reaction template. The composed rule encodes the partial application of the Diels-Alder reaction to the molecule, leaving the diene to be instantiated at a later stage.

sible interface which allows for intuitive usage. The rule composition operators are syntactically implemented in a manner very similar to the mathematical presentation outlined in this contribution. This is achieved by using operator overloading. The computational runtime for all the experiments presented is below five minutes.

MACiE (Mechanism, Annotation, and Classification in Enzymes) [15] is a publicly available hand-curated database of enzymatic reaction mechanisms, where the individual steps of the overall enzyme reaction have been experimental verified. Detailed stepwise mechanistic information (in pictorial form) for more than 300 overall enzyme reactions can be accessed. However, atom traces for the overall enzyme reactions are not available, and information of the mechanism's flexibility with respect to a reordering of individual steps to achieve a given overall reaction is not included. Reactions in MACiE are a natural candidate for our DPO-based rule composition framework.

In the following we consider three rather complex chemical transformations as showcase examples, one example coming from the MACiE database. Each case takes up residence at a different level of organizational abstraction. First, an overall enzyme reaction mechanism is constructed from elementary reaction



(a) Abstract depiction

(b) Chemical example

Fig. 3: A composition $r = r_1 \circ_\cap r_2$ with the matching morphism being a common subgraph R_1 and L_2. The context graphs are omitted for simplicity. (a) Abstract depiction; any (possibly empty) common subgraph of R_1 and L_2 is a candidate for composition. (b) Chemical example with both the Diels-Alder reaction template composed with it self.

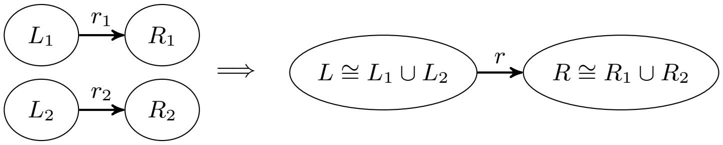


Fig. 4: Composition $r = r_1 \circ_\emptyset r_2$ with an empty match, giving a composed rule which does the operand transformations in parallel.

steps. Second, glycolysis as an example for a complete biochemical pathway with multiple split and merge points that is lumped into a single overall reaction. Finally, the formose process, an auto-catalytic reaction mechanism, illustrates that our methodology works also in networks containing cycles. We emphasize that the full atom traces for each case are computed automatically without additional external information.

3 Results

3.1 *β*-Lactamase

β-lacatamases (MACIE entry 0002, EC number 3.5.2.6) are bacterial enzymes that convey resistance against β-lactame antibiotics such as penicillins by catalysing the overall reaction

$$\underset{\text{(CHEBI:35627)}}{\beta\text{-Lactam}} \quad + \quad \underset{\text{(CHEBI:15377)}}{\text{water}} \quad \rightarrow \quad \underset{\text{(CHEBID:33705)}}{\text{substituted } \beta\text{-amino acid}}$$

by means of a 5-step mechanism, which is detailed in MACiE as follows (see database entry for full details): (1) Lys73 deprotonates Ser70 thereby initiating a nucleophilic addition onto the carbonyl carbon of the β-lactam. (2) The resulting intermediate collapses, cleaving the C-N bond of the β-lactam and the nitrogen deprotonates Ser130. (3) Ser130 deprotonates Lys73. (4) Glu166 deprotonates water, which initiates a nucleophilic addition at the carbonyl carbon. (5) Collapse of this intermediate leads to cleavage of the acyl-enzyme bond and liberates Ser70, which in turn deprotonates the Glu166. The 5 individual steps were modelled as graph grammar rules r_1, …, r_5 depicted in Fig. 5. For step (2) an alternative mechanism has been suggested in [19]: protonation of the β-lactam nitrogen occurs as the first step in the reaction as an initiation step and not as a consequence of the C-N bond cleavage. We modelled this alternative as a replacement of rule r_2 by two graph grammar rules r_{1b} and r_{2b}, see Fig. 6.

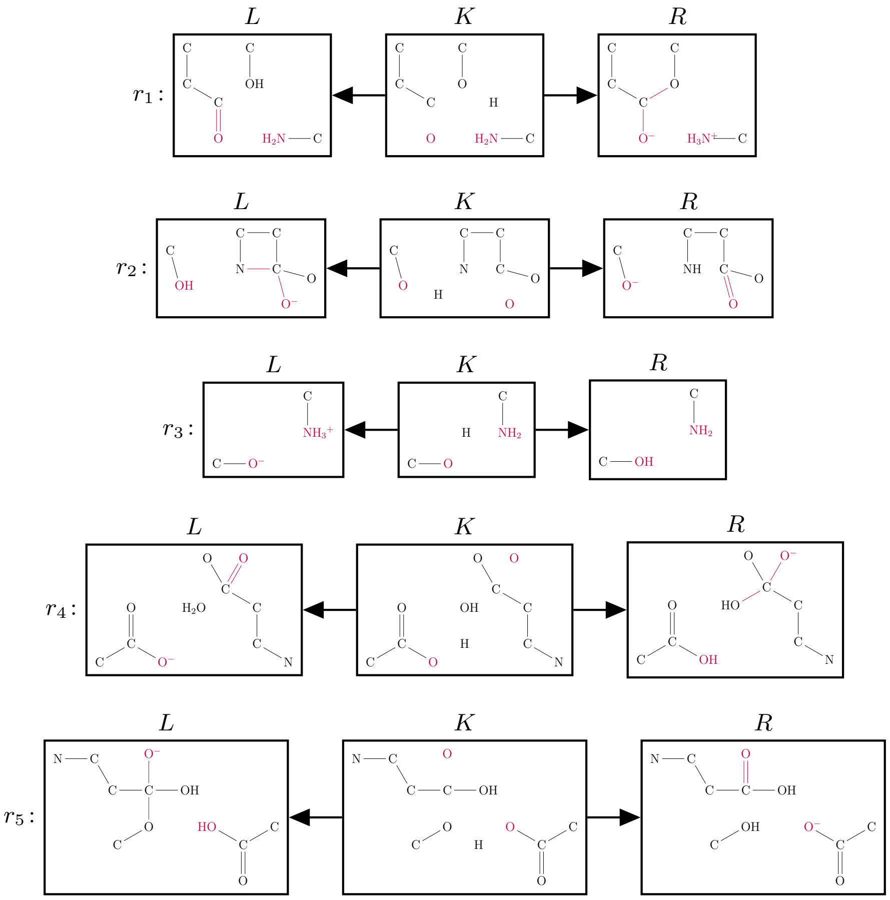


Fig. 5: β-lacatamase: transformation rules for the 5-step enzyme mechanism (MACIE entry 0002, EC number 3.5.2.6).

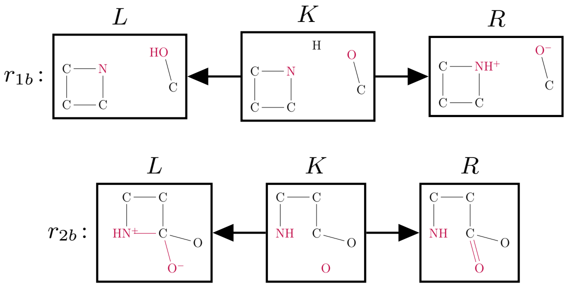
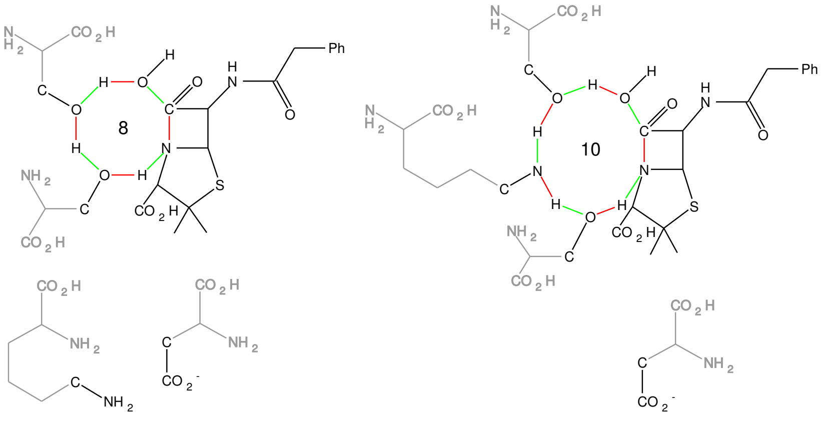


Fig. 6: β-lacatamase: transformation rules to replace step r_2 from Fig. 5, based on the mechanism as suggested in [19].



(a)

(b)

Fig. 7: The two overall reactions resulting from composing the graph rewrite rules for the elementary steps of the β-lactamase EC 3.5.2.6 (MACiE entry 0002), cmp. Eq. (1) (left, 5 steps) and Eq. (2) (right, 6 steps). Red bonds are broken while green bonds are formed during the transformation. While the overall reactions (as typically found in metabolic databases such as KEGG or MetaCyc) are identical, they differ in their hydrogen trace and the size (8 or 10) of the cyclic virtual transition state. Note that the acid/basic catalysts (the two amino acids lysine and glutamine) needed for the reaction to work still show up as precondition in the overall rules. Using partial composition results in two more generic overall reactions. These two rules are depicted as the strict subgraphs resulting from removing the gray parts from the catalysts.

The atom traces for the overall reaction is computed by a composition of the rules $r_1, \ldots, r_5$ with the identity rule for the input compounds, i.e., the β-lactam, water, and the catalysts (Glu166, Lys73, and twice Ser130). Let G and H be the the graph representation of the input and output compounds, respectively. By

$\imath_G = (G, G, G)$ and $\imath_H = (H, H, H)$ we denote the corresponding identity rules. The overall composition

$$\imath_G \circ r_1 \circ r_2 \circ r_3 \circ r_4 \circ r_5 \circ \imath_H \tag{1}$$

results in the two overall rules depicted in Fig. 7. Both are in agreement with the overall mechanism given in MACiE and differ only in their hydrogen traces. The overall cyclic virtual transition states are an 8 cycle and a 10 cycle, which only differ by the exchange of a hydrogen in the amino group of Glu. The alternative model for step 2, which corresponds to

$$\imath_G \circ r_1 \circ r_{1b} \circ \circ r_{2b} \circ r_3 \circ r_4 \circ r_5 \circ \imath_H \tag{2}$$

results in the same two overall rules.

In order to check the flexibility of the reaction with respect to the order of the individual steps of the enzyme mechanism, we investigated all permutations of the rules for the composition order and verified whether the resulting overall rule produces the substituted β-amino acid as final product. Formally, we compute

$$\imath_G \circ r_{\sigma(1)} \circ \ldots r_{\sigma(5)} \circ \imath_H$$

for all 120 permutations σ. Only the following three compositions are well-defined and result in the expected overall rules: $(r_1, r_2, r_3, r_4, r_5)$, $(r_1, r_2, r_4, r_3, r_5)$, and $(r_1, r_2, r_4, r_5, r_3)$. A detailed inspection shows that step r_3 is the recycling step of the mechanism, which can be applied concurrently to steps r_4 and r_5.

The same experiment based on the rule set $\{r_1, r_{1b}, r_{2b}, r_3, r_4, r_5\}$ shows that eight compositions are possible, all resulting in the same atom traces as given above. The first two steps need to be r_1 and r_{1b}, their relative order however is arbitrary. The subsequent rules r_{2b}, r_4, and r_5 must be in this order. The recycling step r_3 requires the rules r_1 and r_{1b} as prerequisite, but can be performed concurrently to the remaining steps, i.e., it may appear in position 3, 4, 5, or 6, thus accounting for the 8 feasible permutations.

This method allows for an automated analysis of the flexibility of the ordering of individual steps. Note, that usually a relatively small number of all possible permutations have to be computed, as most often already the composition of a prefix of an arbitrarily chosen permutation is not possible. For instance, in the previous example, only two of the 30 possible initial two steps are feasible, which prunes most compositions early. The DPO framework provides an inroad to reduce the computational efforts even further. Since each rule is reversible, feasibility can be tested by exploring the space of overall rules from both ends and checking for overlaps at intermediate steps rather then expanding the possible pathways from one end only.

While the focus in this paper is on full rule composition, we illustrate how partial rule composition can be used to automatically detect the required functionality of the catalysts and the additional compounds (in this case a water molecule). Let G' be the graph representation of β-lactam which is the core compound of the reaction. Let $\imath_{G'} = (G', G', G')$ be the identity rule for this compound. The subsequent *partial* composition of the rules, i.e.,

$$\imath_{G'} \circ^c_{\supseteq} r_1 \circ^c_{\supseteq} r_2 \circ^c_{\supseteq} r_3 \circ^c_{\supseteq} r_4 \circ^c_{\supseteq} r_5 \circ^c_{\supseteq} \imath_H$$

result in the overall rule as depicted highlighted in Fig. 7, i.e., any grey molecule or edge disappears. The overall rules show the automatic inference of the necessity of the four functional units of the catalysts and the necessity of the water molecule, as they are subsequently added to the left side of the overall rule during the partial rule composition. When defining graph grammar rules the difficulty often lies in the question of defining the size of the context around a reaction centre: a large context leads to a very specific rule, while a too small context might lead to chemically invalid reactions. Comparing full and partial compositions can be employed as a method to detect the functional units of the catalysts.

The atom mapping of the full composition result shows that in the composed rule with the 8 cycle the acid-base catalysts lysine and glutamic acid are unmodified during the overall process although they are necessary for the mechanism. In the composed rule with the 10 cycle only the acid-base catalyst glutamic acid is unmodified. The other catalysts and the water molecule are modified, however only based on the fact that the hydrogen atom for proton donation is different from the accepting hydrogen.

3.2 Glycolysis

In order to illustrate the potential of rule composition to detect different carbon traces we analyse two variants of the glycolysis pathway. The net reaction of the glycolysis pathway is the conversion of a glucose molecule into two pyruvates while releasing the high-energy compound ATP. For a recent and detailed review see [20]. The most common type of glycolysis is the Embden-Meyerhof-Parnas (EMP) pathway. The alternative Entner-Doudoroff (EP) pathway [21] is known to lead to different carbon atom traces in one of the two pyruvates. Labelling experiments in glycolysis are commonly used to analyse the activity of the different pathways (e.g., [22]). The analysis of such data is quite tedious in practice since the possible atom traces for an overall pathway usually have to be constructed manually.

The individual steps and the enzymes catalysing them are well understood for both the EMP and the ED pathway, and a detailed discussion is far beyond the scope of this paper. The key difference is that EMP yields 2 ATP per glucose while ED produced only 1 ATP for each glucose molecule. In the evolutionary time-line the ED pathway is thought to predate the EMP pathway [23], which is the dominant pathway among eukaryotes. There exists a clear trade-off between the ATP yield and the thermodynamic driving force (rate) of a glycolytic pathway [24]. Flamholz et al. [25] recently found that the high yield EMP pathway needs to maintain much higher enzyme levels (thereby causing greater protein production costs) to support the same flux as through the ED pathway. The two pathways have been modelled using the following transformation rules:

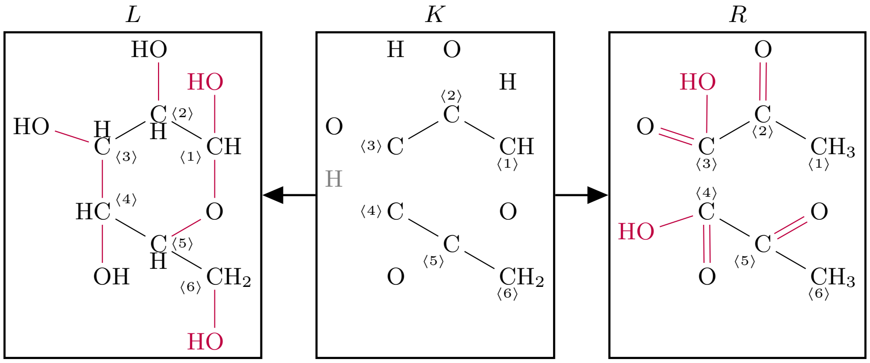


Fig. 8: Simplified transformation rule for the overall EMP pathway with each carbon atom labelled. Some hydroxyl-groups appear to be destroyed and created due to the simplification such that only glucose and pyruvate is depicted.

r_1 Pyranose-furanose
r_2 Furanose-linear
r_3 Ketose-aldose
r_4 ATP-phosphorylation
r_5 ATP-dephosphorylation
r_6 NAD+-phosphorylation
r_7 Phosphomutase
r_8 Enolase
r_9 Keto-enol
r_{10} NAD+-oxoreductase
r_{11} Lactonohydrolase
r_{12} Hydrolyase
r_{13} Reverse aldolase

The details of the rules can be found in the Appendix. As in the previous section, we use $\imath_{G(EMP)} = (G, G, G)$ and $\imath_{G(EP)}$ to model input graphs for the two pathways. For EMP G consists of 1 glucose, 2 ATP, 2 ADP, 2 phosphates, and 2 NAD$^+$. In the case of ED the set of input compounds is 1 ATP, 1 ADP, 1 phosphate, and 2 NAD$^+$. $\imath_{H(EMP)} = (H, H, H)$ and $\imath_{H(EP)}$ correspondingly model the output. In the case of EMP the set of output compounds is 2 pyruvates, 4 ATP, 2 NADH, 2 water, and 2 H$^+$. In the case of ED the set of output compounds is 2 pyruvates, 2 ATP, 2 NADH, 2 H$^+$, and 2 water. Note that the same approach as presented in the MACiE 0002 example for automatically inferring the necessary functional groups of G and H could be applied; an explicit definition of the catalysts in G and H would not be necessary.

For EMP we compute the composition

$$\imath_{G(EMP)} \circ \overbrace{\imath_G \circ r_4 \circ r_1 \circ r_4 \circ r_2 \circ r_{13} \circ r_3}^{\text{Glucose} \rightarrow \text{2 G3P}} \\ \circ \underbrace{(r_6 \circ_\emptyset r_6) \circ (r_5 \circ_\emptyset r_5) \circ (r_7 \circ_\emptyset r_7) \circ (r_8 \circ_\emptyset r_8) \circ (r_5 \circ_\emptyset r_5) \circ (r_9 \circ_\emptyset r_9)}_{\text{2 G3P} \rightarrow \text{2 Pyruvate}} \circ \imath_{H(EMP)}$$

and for ED we compute

$$\imath_{G(ED)} \circ \underbrace{r_4 \circ r_{10} \circ r_{11} \circ r_{12} \circ r_{13}}_{\text{Glucose} \rightarrow \text{G3P + Pyruvate}} \circ \underbrace{r_6 \circ r_5 \circ r_7 \circ r_8 \circ r_5 \circ r_9}_{\text{G3P + Pyruvate} \rightarrow \text{2 Pyruvate}} \circ \imath_{H(ED)}$$

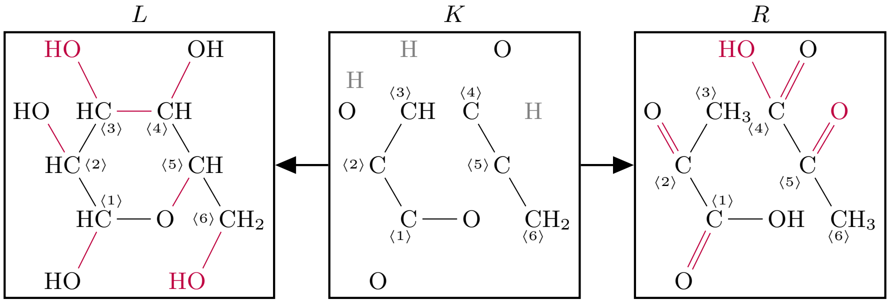


Fig. 9: Simplified transformation rule for the overall ED pathway with each carbon atom labelled. Some hydroxyl-groups appear to be destroyed and created due to the simplification such that only glucose and pyruvate is depicted.

The resulting rules are depicted in Fig. 8 and Fig. 9. To reduce clutter we only draw the glucose and pyruvate components (formally this can be achieved by composing with rules that unbinds the unwanted components). Clearly, the carbon traces of the two rules differ. Such an approach can be used for an automated design of labelling experiments to detect the activity of pathway alternatives.

The prefixes of the rule composition expression allows to infer all the intermediate compounds and their corresponding atom traces relatively to the input compounds. The summary of this analysis is depicted in Fig. 10 for both pathways (traces shown for carbon atoms only). The black reaction arrows show the EMP pathway, the green arrows show the ED pathway. The six carbon atoms from glucose are converted into two pyruvate molecules in two different ways depending on whether EMP or ED was used to catabolise glucose. While the EMP pathway has a Fructose 1,6-bisphosphate as an intermediate, in which a pentose ring is cleaved, in the ED pathway the hexose ring of the Glucose 6-phosphate is cleaved. The carbon atom trace of one of the two pyruvates is identical, while it is inverted in the other pyruvate.

3.3 Formose Reaction

The formose reaction [26] has been extensively discussed as a possible prebiotic route to higher carbohydrates. In contrast to the two previous examples it does not require enzyme catalysis. It converts two formaldeyhydes and a glycolaldehyde into two glycolaldehydes and hence is an example of an overall autocatalytic reaction that is the net result of a rather complex network of individual reactions.

The individual steps of the formose reaction belong to only two distinct reversible reaction patterns, namely keto-enol tautomerism and aldol reaction. Their DPO rule formulations are given in Fig. 11. Different detailed sequences of individual reactions have been hypothesized. Two prominent examples are based on [26] and [27] and are shown Fig. 12. Note, that the most commonly discussed cycle (e.g., [27]) includes the symmetric molecule dihydroxyacetone. However,

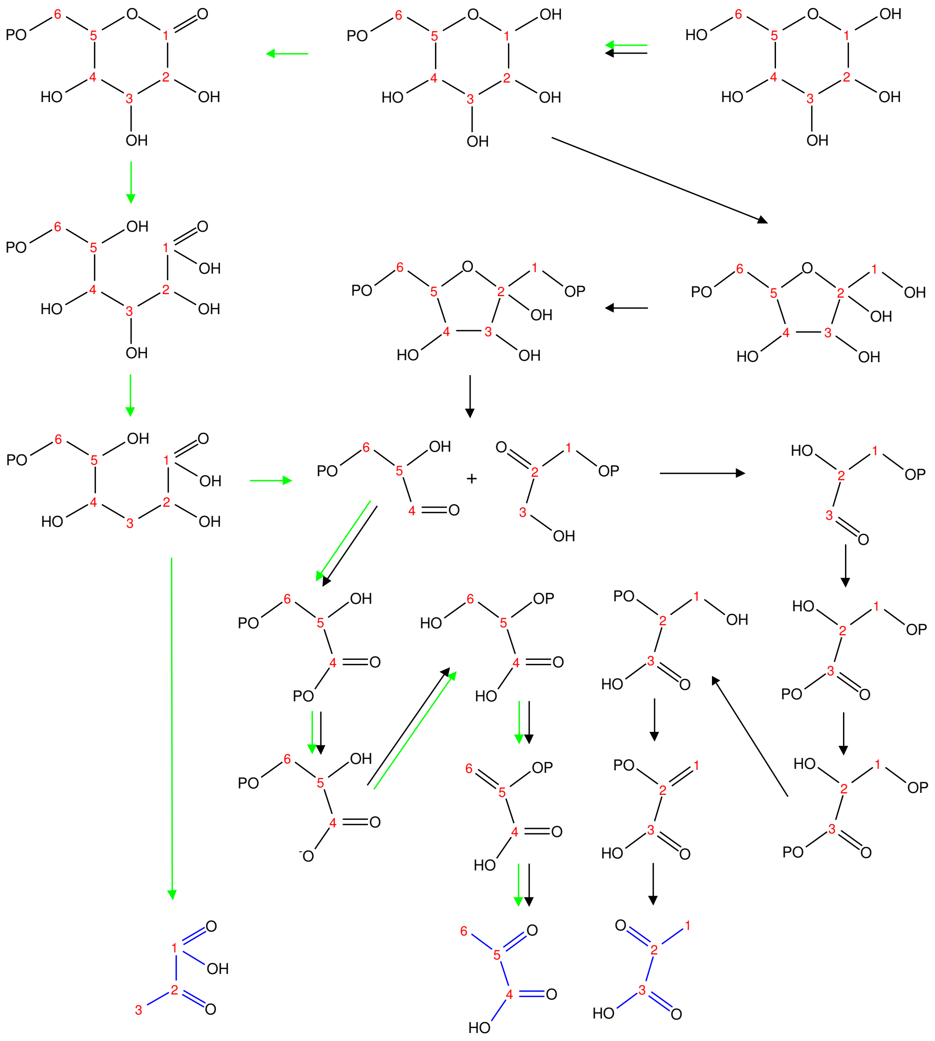
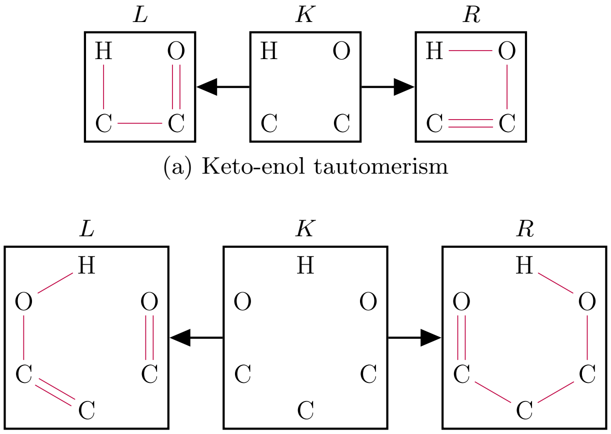


Fig. 10: Carbon atom trace of glycolysis; the Embden-Meyerhof-Parnas pathway (EMP) is depicted with black reaction arrows, the Entner-Doudoroff pathway (ED) is depicted with green reaction arrows. The six carbon atoms from glucose are converted into two pyruvate molecules (highlighted in blue) in two different ways depending on whether EMP or ED was used to catabolise glucose, one pyruvate overlaps in both pathways.

the shortest possible cycle (in term of the number of reactions) is based on [26] and does not include this intermediate.

The input graph G comprises two formaldehyde and one glycolaldehyde molecule, the goal H consists of two copies of glycolaldehyde. Both are represented by their corresponding identity rules $\imath_G = (G, G, G)$ and $\imath_H = (H, H, H)$.



(a) Keto-enol tautomerism

(b) Aldol addition

Fig. 11: The transformation rules for the formose reaction. Only the forward directions of the reversible rules r_1 and r_2 are show. The reverse rules, r_1^{-1} and r_2^{-1}, are obtained by swapping the left and right graphs.

The two proposed pathway are represented as

$$\imath_G \circ r_1 \circ r_2 \circ r_1 \circ r_2 \circ r_1 \circ r_1^{-1} \circ r_2^{-1} \circ r_1^{-1} \circ \imath_H \tag{3}$$

and

$$\imath_G \circ r_1 \circ r_2 \circ r_1 \circ r_1^{-1} \circ r_1 \circ r_2 \circ r_1 \circ r_1^{-1} \circ r_2^{-1} \circ r_1^{-1} \circ \imath_H \tag{4}$$

where r_1, and r_1^{-1} are the keto-enol and enol-keto transitions of keto-enol tautomerism, r_2 is aldol addition and r_2^{-1} is its inverse, i.e., cleavage.

Based on a prefix composition of Eq. (3) and Eq. (4), the traces for the intermediates can be computed as in the glycolysis example. The subsequent modification of the carbon traces is summarized in Fig. 13. Note that in this figure sequences of isomerisation and aldol-addition steps are depicted as one step in order to minimize clutter.

The rule composition based on Eq. (4) results in six non-isomorphic composed overall rules, each having a different carbon trace for the 4 carbon atoms of G. One of those rules is depicted in Fig. 14. While 4! carbon traces are a trivial upper bound, the mechanism allows only for six of them, as the carbon from the second added formaldehyde cannot end up as carbonyl carbon in the resulting glycolaldehyde. If a labelling experiment could be performed with all carbons uniquely labelled and if the glycolaldehydes after exactly one instantiation of the reaction cycle would be analysed, then not twelve but only nine different glycolaldehydes could be observed. If the mechanism follows Eq. (3) (i.e., dihydroxyacetone is not an intermediate of the mechanism), the two input formaldehydes never combine into the same glycolealdehyde, reducing the set of overall reactions to four rules. Using the same labelling experiment as above, only six different glycolaldehydes could be observed.

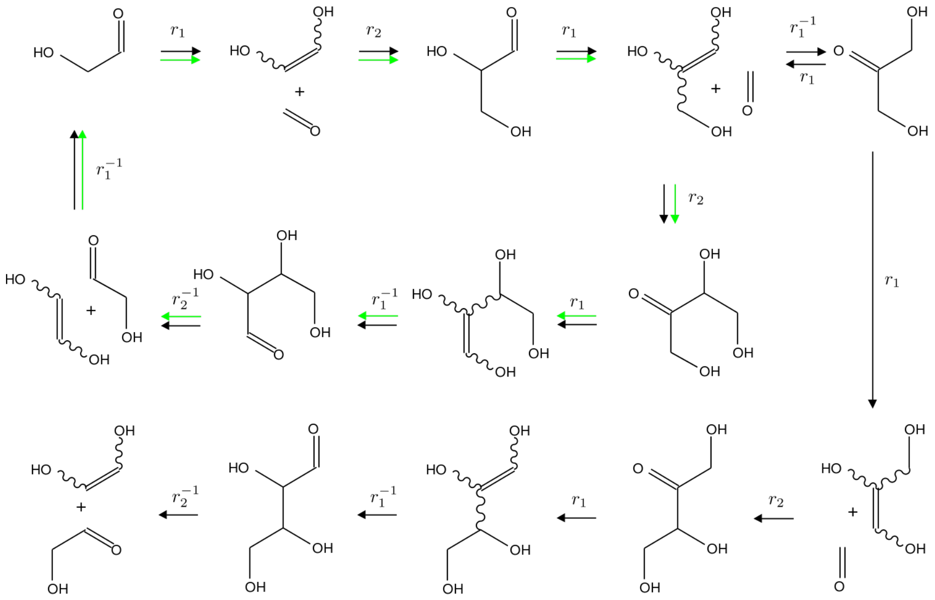


Fig. 12: Detailed mechanism for the formose process. The labels r_1 and r_1^{-1} indicate the keto-enol tautomerisation (forward and backward), r_2 and r_2^{-1} refer to aldol- and retro-aldol reaction. The shortest possible autocatalytic cycle is illustrated with green reaction arrows, the possibilities with dihydroxyacetone as an intermediate are illustrated with black reaction arrows. In order to allow and easy visual tracking of carbon atoms, we duplicated the sequence of compounds after the second aldol addition.

4 Conclusions

Chemical reactions are a particularly fruitful area of applications for rule based transformation systems. After all, graphs are the level of abstraction most commonly used by chemists to represent molecules, and the “named reactions” of organic chemistry explicitly are rewriting rules. The composition of multiple individual elementary steps to a single “overall reaction”, furthermore, is common practice in the chemistry literature. Rule composition is a powerful tool in graph grammars (and potentially also in other calculi modelling concurrent computation) to investigate such concrete (multi-step) derivations in more detail. This approach is much more general, however, and by no means restricted to the domain of chemistry.

The key idea is to reduce a given derivation to a composite transition rule bound to the initial and final objects. This reduced the problem of representing this composite rule as a sequence of elementary rules from which it is composed. In practical applications there is often only a small number of feasible decompositions that are easily identified by backtracking-style enumeration. DPO graph rewriting offers an interesting advantage for the task at hand. DPO rules are

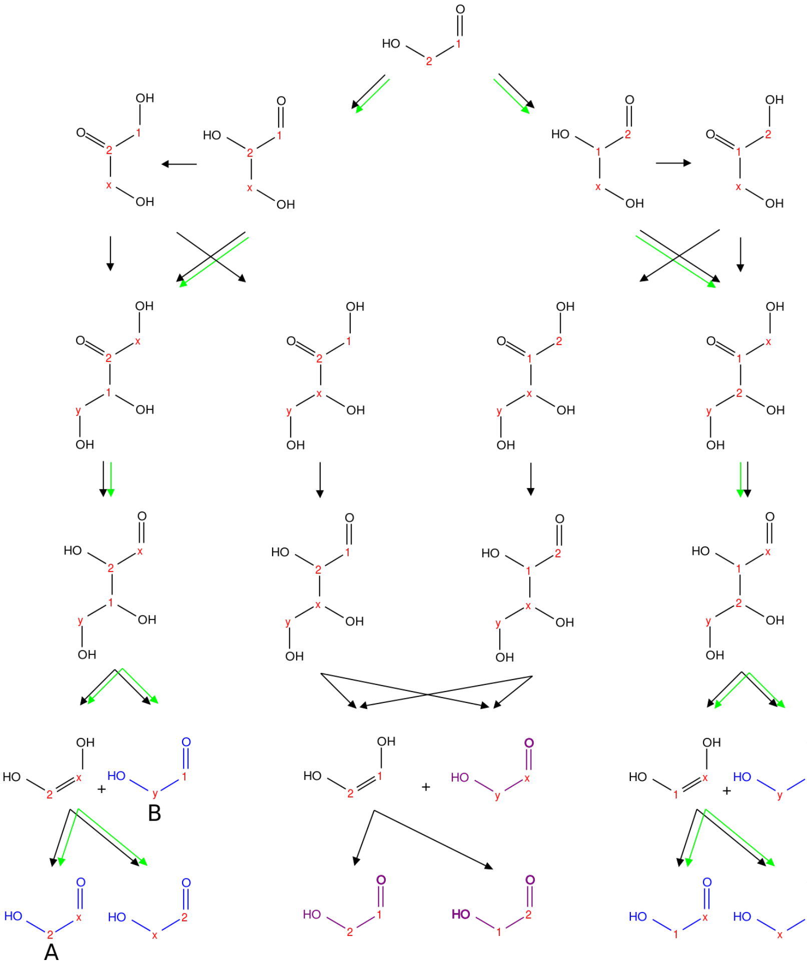


Fig. 13: Carbon atom traces for one round of the formose process: green reaction arrows indicate possible carbon atom traces following the shorter formose cycle, black reaction arrows indicate possible traces following the cycle having dihydroxyacetone as intermediate. The carbonyl (resp. alcohol) carbon of the starting molecule glycolealdehyde is labelled 1 (resp. 2). After condensation of this molecule with two formaldehydes labelled x and y, the intermediate molecule decomposes into two glycolealdehydes. Depending on the mechanism, the labelled carbon atoms end up in nine (resp. six) different positions of the resulting glycolealdehydes (shorter cycle: blue molecules, longer cycle: blue and purple molecules). Note that the carbon from the second formaldehyde (y) can not end up as carbonyl carbon in the resulting glycolealdehydes. The shorter cycle allows for a strict subset of carbon traces only, the two formaldehydes never recombine into a glycolealdehyde. From the six (resp. four) possible composed overall reactions of the longer (resp. shorter) cycle, the one that results in the two blue glycolaldehydes A and B is depicted in Fig. 14.

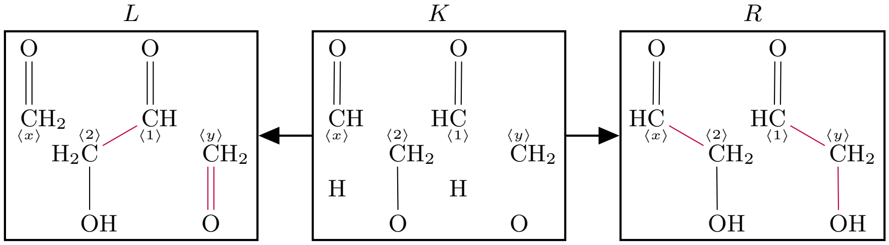


Fig. 14: Formose reaction: one of six possible overall rules based on Eq. (4) including carbon atom trace information. The mapping of the atoms corresponds to creation of the glycolaldehydes denoted A and B in Fig. 13.

guaranteed to be reversible, hence the search for a path from initial to finite state can be broken up into an exploration from both ends.

Overall chemical reactions are good proving ground for the rule composition approach. Chemical reaction databases usually only list the products and educts. In some cases the reaction mechanisms of the elementary steps catalysed by relevant enzyme(s) are also known, but these lines of information are not connected in a way that would make it easy to retrieve the atom maps. These are key to analysing isotope labelling experiments and hence add practical relevance to our examples. The rule composition framework, however, also provides additions information, such as possible alternative in the relative timing of reaction steps and information on concurrency of elementary reactions. While limited in enzyme reactions (the first two of our three showcase examples), one-pot reactions such as the formose reaction form the other extreme, with large numbers of concurrent reactions. In such a scenario rule composition is a concise way to model reaction pathway in a manner that allows coarse graining away from elementary steps while at the same time allowing us to keep track of chemically distinguishable overall transformations that produce different atom traces. This information is important since it allows to disentangle the relative importance, and possibly even the reaction kinetics, of alternative pathways in complex reaction mixtures. In that sense rule composition can be viewed as an efficient and automatic model reduction technique.

5 Acknowledgments

This work was supported in part by the *Volkswagen Stiftung*, the COST Action CM1304 “Emergence and Evolution of Complex Chemical Systems”, and the Danish Council for Independent Research, Natural Sciences.

References

1. Fontana, W., Buss, L.W.: What would be conserved if “the tape were played twice”? Proc. Natl. Acad. Sci. USA **91** (1994) 757–761

2. Regev, A., Shapiro, E.: Cells as computation. Nature **419** (2002) 343
3. Danos, V.: Formal molecular biology. Theor. Comp. Sci. **325** (2004) 69–110
4. Blinov, M.L., Yang, J., Faeder, J.R., Hlavacek, W.S.: Graph theory for rule-based modeling of biochemical networks. In Priami, C., A. Ingólfsdóttir, A., Mishra, B., Nielson, H.R., eds.: Transactions on Computational Systems Biology VII. Volume 4230 of Lect. Notes Comp. Sci., Springer 89–106
5. Păun, G.: Computing with membranes. J. Comp. Syst. Sci. **61** (2000) 108–143
6. Cardelli, L.: Brane calculi. In Danos, V., Schachter, V., eds.: Computational Methods in Systems Biology, CMSB'04. Volume 3082 of Lect. Notes Comp. Sci., Springer (2005) 257–278
7. Arnold, C., Stadler, P.F., Prohaska, S.J.: Chromatin computation: Epigenetic inheritance as a pattern reconstruction problem. J. Theor. Biol. **336** (2013) 61–74
8. Hlavacek, W., Faeder, J.R., Blinov, M.L., Posner, R.G., Hucka, M., Fontana, W.: Rules for modeling signal-transduction systems. Sci. STKE **2006** (2006) 334–re6
9. Sekar, J.A., Faeder, J.R.: Rule-based modeling of signal transduction: a primer. Methods Mol Biol. **880** (2012) 139–218
10. Berry, G., Boudol, G.: The chemical abstract machine. In: POPL '90 – Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, Assoc. Computing Machinery (1990) 81–94
11. Benkö, G., Flamm, C., Stadler, P.F.: A graph-based toy model of chemistry. J. Chem. Inf. Comput. Sci. **43** (2003) 1085–1093
12. Danos, V., Feret, J., Fontana, W., Harmer, R., Hayman, J., Krivine, J., Thompson-Walsh, C., Winskel, G.: Graphs, rewriting and pathway reconstruction for rule-based models. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012). Volume 18 of Leibniz International Proceedings in Informatics (LIPIcs). (2012) 276–288
13. Beck, M., Benkö, G., Eble, G., Flamm, C., Müller, S., Stadler, P.F.: Graph grammars as models for the evolution of developmental pathways. In Schaub, H., Detje, F., Brüggemann, U., eds.: The Logic of Artificial Life: Abstracting and Synthesizing the Principles of Living Systems, Berlin, IOS Press, Akademische Verlagsgesellschaft (2004) 8–15
14. Holliday, G.L., Bartlett, G.J., Almonacid, D.E., O'Boyle, N.M., Murray-Rust, P., Thornton, J.M., Mitchell, J.B.O.: MACiE: a database of enzyme reaction mechanisms. Bioinformatics **21** (2005) 4315–4316
15. Holliday, G.L., Andreini, C., Fischer, J.D., Rahman, S.A., Almonacid, D.E., Williams, S.T., Pearson, W.R.: MACiE: exploring the diversity of biochemical reactions. Nucleic Acids Research **40** (2012) D783–D789
16. Andersen, J.L., Flamm, C., Merkle, D., Stadler, P.F.: Inferring chemical reaction patterns using graph grammar rule composition. J. Syst. Chem. **4**(4) (2013)
17. Weininger, D.: SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. **28**(1) (1988) 31 – 36
18. Weininger, D., Weininger, A., Weininger, J.L.: SMILES 2. Algorithm for Generation of Unique SMILES Notation. J. Chem. Inf. Comput. Sci. **29**(2) (1989) 97 – 101
19. Atanasov, B.P., Mustafi, D., W., M.M.: Protonation of the beta-lactam nitrogen is the trigger event in the catalytic action of class A beta-lactamases. Proc. Natl Acad. Sci. **97**(7) (2000) 3160–3165
20. Bar-Even, A., Flamholz, A., Noor, E., Milo, R.: Rethinking glycolysis: on the biochemical logic of metabolic pathways. Nat Chem Biol **8**(6) (2012) 509–517

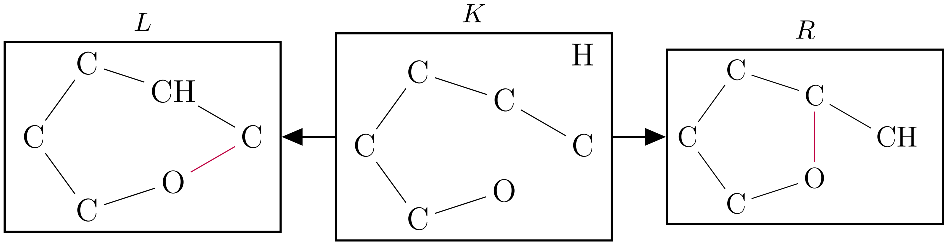
21. Entner, N., Doudoroff, M.: Glucose and gluconic acid oxidation of pseudomonas saccharophila. J. Biol. Chem. **196** (1952) 853–862
22. Borodina, I., Schöller, C., Eliasson, A., Nielsen, J.: Metabolic network analysis of streptomyces tenebrarius, a streptomyces species with an active entner-doudoroff pathway. Appl Environ Microbiol **71**(5) (2005) 2294–2302
23. Romano, A.H., Conway, T.: Evolution of carbohydrate metabolic pathways. Res Microbiol **147**(6/7) (1996) 448–455
24. Stettner, A.I., Segré, D.: The cost of efficiency in energy metabolism. PNAS **110**(24) (2013) 9629–9630
25. Flamholz, A., Noor, E., Bar-Even, A., Liebmeister, W., Milo, R.: Glycolytic stratewgy as a tradeoff between energy yield and protein cost. PNAS **110**(24) (2013) 10039–10044
26. Benner, S., Kim, H., Ricardo, A.: Planetary organic chemistry and the origins of biomolecules. Cold Spring Harb Perspect Biol **2**(7) (2010) a003467
27. Breslow, R.: On the mechanism of the formose reaction. Tetrahedron Letters **1**(21) (1959)

Appendix

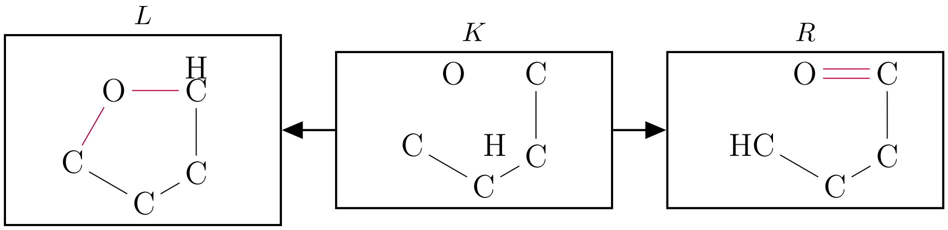
This Appendix is provided for the review process. In case of acceptance it will be published as a web supplement.

A Transformation Rules for Glycolysis

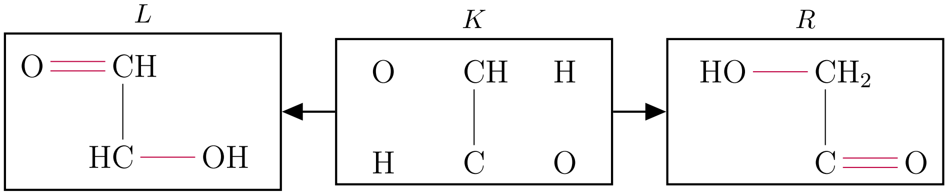
A.1 r_1, Pyranose-furanose



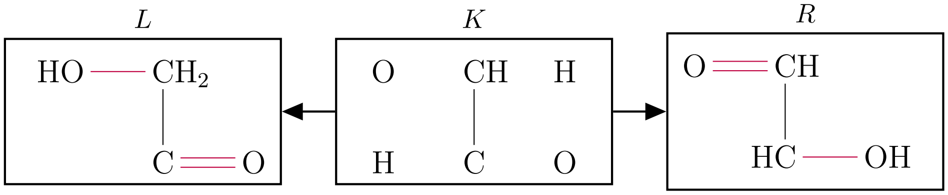
A.2 r_2, Furanose-linear



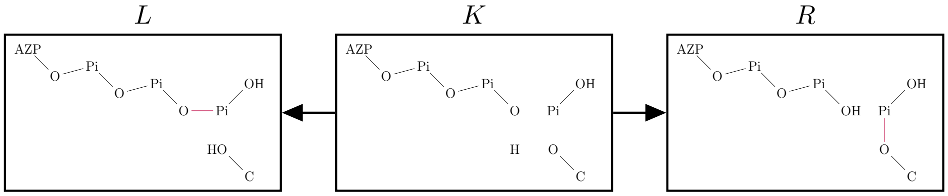
A.3 r_3, Ketose-aldose



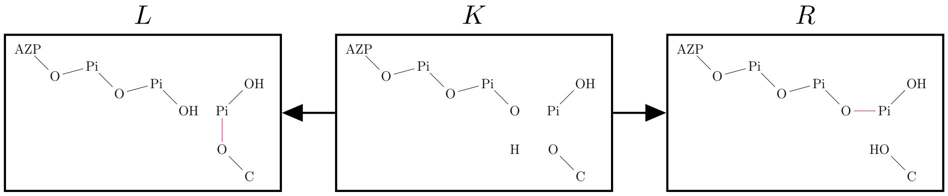
A.4 r_4, ATP-phosphorylation



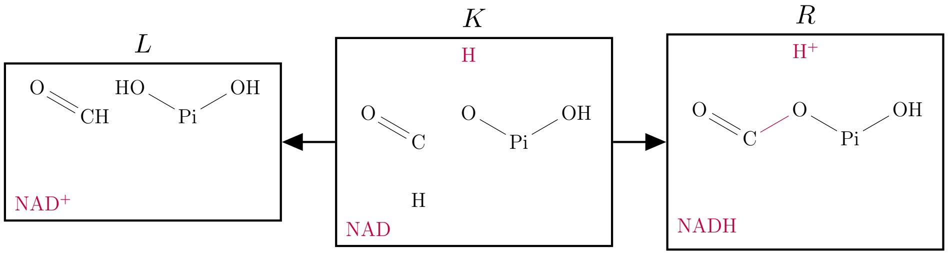
A.5 r_5, ATP-dephosphorylation



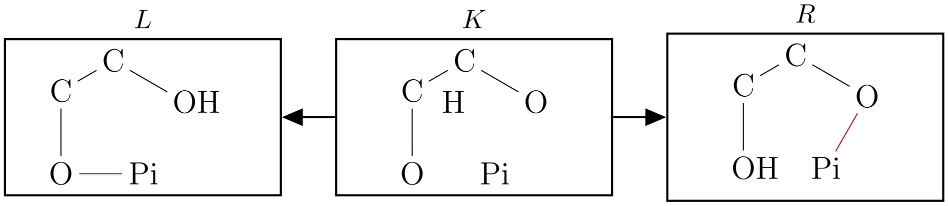
A.6 r_6, NAD+-phosphorylation



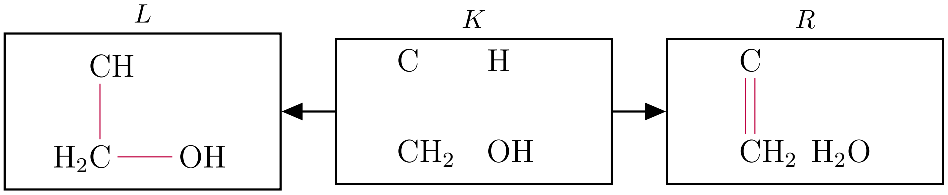
A.7 r_7, Phosphomutase



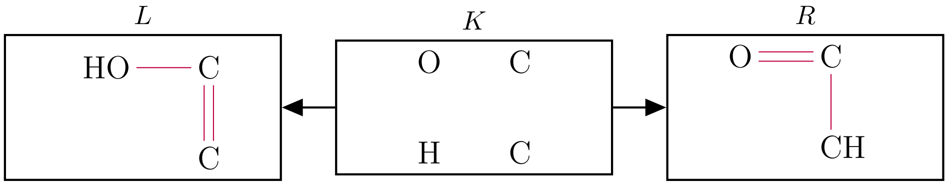
A.8 r_8, Enolase



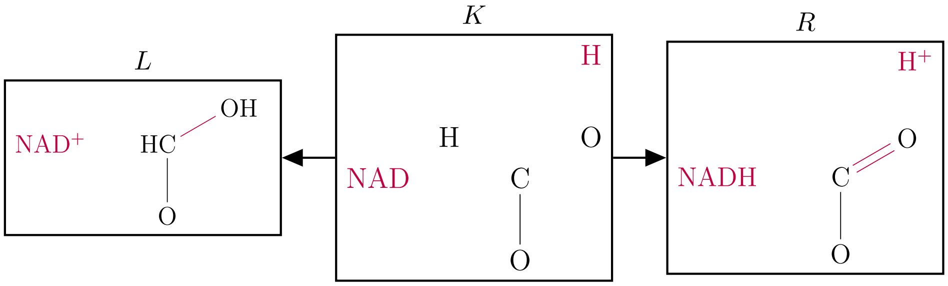
A.9 r_9, Keto-enol



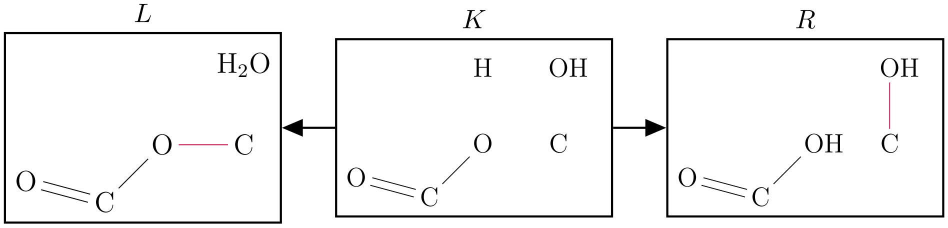
A.10 r_{10}, NAD+-oxoreductase



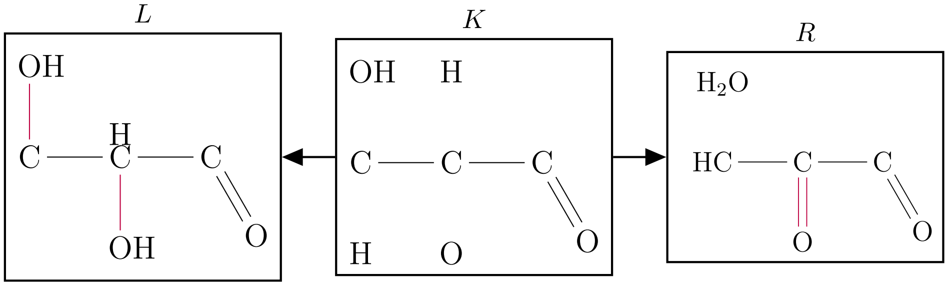
A.11 r_{11}, Lactonohydrolase



A.12 r_{12}, Hydrolyase



A.13 r_{13}, Reverse aldolase


L
OH
C
H
C
C
OH
O
K
OH
H
C
C
C
H
O
O
R
H$_2$O
HC
C
C
O
O