Characterizing Catalytic Mechanisms with Overlay Graphs

Jakob L. Andersen, Rolf Fagerberg, Christoph Flamm, Walter Fontana, Juraj Kolčák, Christophe V.F.P. Laurent, Daniel Merkle* and Nikolai Nøjgaard

Abstract

Understanding the underlying chemistry of a catalytic process is essential for advancing of its medical and industrial applications. A well defined and compact representation of a catalytic process is becoming exceedingly valuable in face of the growth in computer assisted development strategies. A step-wise partition of a catalytic process is traditional in organic chemistry. We argue, however, that such a description remains incomplete, especially in face of automation and formal methods. In this contribution—based on enzymatic reaction mechanisms—we fully formalize the step-wise notion of a catalytic process within the mathematical framework of graph transformation, and propose a concise representation as an overlay graph (OG). We formally define this concept in a dual form as an OG itself and an overlay rule. The former representing a static summary of the process, and the latter being an executable in the graph transformation formalism. We demonstrate that OGs readily expose the underlying chemistry and facilitate the interpretation of catalytic processes. The emergence of electron flow patterns, for instance, provides an excellent stage for classification efforts. We further show how the executable overlay rules can be employed to automatically assign a fully specified mechanism to a reaction with only an overall description, i.e. educt and product molecules.

Introduction

In chemistry, a given overall reaction is often known to proceed as a sequence of smaller reaction steps, collectively termed the reaction mechanism. Each step in a reaction mechanism typically consists of the relocation of a small set of electrons.

In particular, this is the case for catalytic reactions. Catalytic reactions are reactions where an additional reagent—the catalyst—opens up a route between the starting materials and the products with a lower activation barrier than the corresponding uncatalyzed process having the same overall reaction. Alternatively, the catalyst might also be able to accelerate an existing (non-catalyzed) pathway if the transition state exhibits a stronger binding than the substrate. In such a scenario the activation energy would also be lowered.\(^1\) The catalyst is not itself consumed in the process. It participates in intermediate reaction steps, but is in the end regenerated and comes out of the process structurally unchanged.\(^2\)

For catalytic reactions in biochemistry, the number of steps in a reaction mechanism may be quite large: the Mechanism and Catalytic Site Atlas (M-CSA) database\(^3\) lists close to thousand enzymes and for many of these it gives suggestions for reaction mechanisms, some of which contain over ten steps. More importantly, most of the molecular changes induced in intermediate steps are often not reflected in the overall reaction at all.

Thus, for catalytic reactions in general, and for enzymatic reactions in particular, the over-
all reaction is often too simple to capture the essential chemical components involved in the reaction. For instance, the overall reaction is unable to distinguish between different enzymatic mechanisms for a given reaction.

Our aim in this paper is to develop a richer, yet still succinct, notion of overall reaction which expresses major aspects of the underlying reaction mechanism. Our claim is that this may be a useful tool when describing, analyzing, searching for, and designing catalytic reactions, in particular when combined with computational methods. In the second part of the paper, we give some first use cases justifying this claim.

We call our notion Overlay Graphs (OG). Our starting point is the so-called Imaginary Transition Structures (ITS) of Fujita. ITS are structural formulas with three bond types: bonds only seen in the educt molecules, bonds only seen in the product molecules, and bonds appearing in both. We propose to augment the concept of ITS with two additional bond types capturing transient bond changes in the underlying reaction mechanism. The first new type is for bonds that appear in both educts and products, but are transiently absent in the mechanism. The second new type is for bonds that do not appear in neither educts nor products, but are transiently present in the mechanism.

The concept of OG is quite straight-forward on the intuitive level—see Figure 1 for an example. However, we take the effort to also define the concept of OG more rigorously by expressing it in the context of graph transformation. As we outline now, this effort comes with a number of benefits.

It is standard to describe molecules by chemical graphs, i.e., undirected graphs with vertices labeled by atom types and edges labeled by bond types. As chemical reaction constitute a change of educt molecules into product molecules, formal methods for specifying transformations of graphs are well suited for modeling chemical reactions. In this paper, we use the so-called double push-out (DPO) formalism for graph transformation when developing our notion of OG. Besides ensuring a high level of precision in our definitions, it also provides an interface to the computational framework MOD. This framework is able to test for matches between graphs and graph transformation rules, and in case of a match to calculate the effect of applying the transformation rule to the graph. In a chemical context, this amounts to being able to execute (in silico) a set of specified reactions on a set of given substrates.

The DPO formalism is very flexible, since graph transformation rules used in matches can specify the full educt molecule, just the core reaction center, or anything in between, which allows for modeling classes of reactions with a single rule. One further benefit is that individual atoms can be tracked across reactions. An enzyme, although topologically unchanged by the reaction that it catalyzes, may well swap atoms with the substrate, so tracking individual atoms is relevant for instance when investigating biochemical pathways via stable isotope labeling of molecules.

Clearly, being able to computationally model, manipulate, search, and execute chemical reactions on the level of DPO opens up many avenues of investigation. The second part of this paper exemplifies this in the context of OG.

In short, our contributions contain the extension of the notion of ITS graphs to OG by means of a strict generalization. That is, if a mechanisms consists of a single step, the obtained OG is identical to the ITS graph. We additionally capture the notion of OG in full mathematical formalism, allowing us to utilize the concept within MOD, and consequently perform computational analyses. To counteract the lack of knowledge of the exact activate site for many enzymatic reactions, we propose a reduced version of an OG, called substrate OG, which only specifies the substrates. This version allows us to investigate enzymatic reactions without explicit knowledge of the catalytic pocket. Finally, we construct OGs for a number of carboxylic ester hydrolases from the M-CSA database. We use the obtained OGs to demonstrate not only their use in interpreting the underlying chemistry, but also in searching for reactions which could be explained by the same mechanism. The latter application further motivates the possibility of using OGs as...
means of automated mechanism classification, akin to the application of ITS graphs for classification of organic reactions.  

Methods

Reaction Mechanisms

Rather than a singular change, a chemical reaction is often understood as a sequence of reaction steps that describe the conversion of educt into product molecules through a series of intermediates. On the other hand, a reaction mechanism is composed of a sequence of steps and is commonly used to describe catalytic reactions—especially in enzymology. An example of a reaction mechanism consisting of four steps is shown in Figure 1. While the overall reaction description portrays the catalytic components as “observers” and oftentimes does not even explicitly represent them, a reaction mechanism provides the exact instructions on how the reaction should be executed, including transient modifications of the catalytic components.

In order to obtain an unambiguous topological description, we understand a reaction mechanism as a sequence of electron movements on top of the educt molecules. By convention this flow of electrons is depicted by “curved arrows” (magenta arrows in Figure 1). These electron movements are generally aggregated in the form of reaction steps (Figure 1), which can range from individual electron movements to entire reactions. One can therefore utilize reaction mechanisms to describe a single catalytic reaction, as well as a whole metabolic processes, e.g. the TCA cycle.  

Graph Transformation Framework

To be able to reason about ambiguity of mechanisms, we need to express them within a sufficiently formal setting. Since molecules are traditionally represented as undirected, connected, and typed graphs, the formalism of graph transformation seems appropriate. The framework has a well established foundation rooted in category theory, as well as numerous chemical applications.  

A molecule is treated as a typed graph, where a typed node represents an atom with a charge while a typed edge represents a bond of a given order. A collection of molecules is then represented by a disconnected typed graph, referred to as a mixture, where each connected component is a molecule. A rule describes how one mixture can be transformed into another. This is achieved by embedding an input pattern into a mixture and transforming the matched part according to an output pattern.

Formally, a (double-pushout) rule is a span

\[ p = (L \xleftarrow{l} K \xrightarrow{r} R) \]

where \( L \) and \( R \) are the left and right patterns respectively, and \( K \) is the invariant graph, containing elements common to \( L \) and \( R \). The exact correspondence between \( L \) and \( R \) is then given by the injections \( l \) and \( r \).

In our case, we want an atom that is changing charge to preserve its identity. That is, we want to be able to relabel a vertex, rather than delete and recreate it with the new label. To achieve this, we use a previously described formalism, where the invariant graph \( K \) is untyped, allowing us to relate two vertices with different labels via \( l \) and \( r \).

The key aspect of such an interpretation of a reaction mechanism lies in the electron movements being described on top of the educt molecules. This naturally provides the atom maps, which relate atoms in the intermediates and products to their original position in the educt molecules. It is by virtue of the atom maps, that our definition of a reaction mechanism gives a complete topological description of a reaction, including all transient modifications. To further ensure a reaction mechanism is fully unambiguous, including the causal information, it has to constitute reaction steps that correspond to individual electron jumps, that is, the smallest chemically relevant changes.
Figure 1: Example of a four step mechanism (M-CSA entry 218). This EC 3.1.1.3 reaction is performed by a pancreatic Triacylglycerol lipase (UniProtKB: P29183) and catalyzes the conversion of a triacylglycerol (iv) into a fatty acid ion (x) and diacylglycerol (viii). In the first step the Histidine (ii) deprotonates the Serine (i), which is then able to attack the substrate (iv). After this, the Serine is covalently bound to the substrate (vi) and the Histidine side chain is fully protonated (v). In the second step the oxyanion collapses, which leads to the release of diacylglycerol (viii) and the deprotonation of the Histidine (ii). In the third step, the Histidine (ii) deprotonates the water (iii). This allows the water molecule to attack the acylated Serine (vii). In step four, the oxyanion collapses once more, which leads to the elimination of a fatty acid ion (x) and the reconstitution of the amino acids. The lower right panel shows all the steps overlaid on top of each other, corresponding to the OG. Here, the bonds created or broken during the reaction are shown in green and red, respectively. Transiently modified bonds are depicted in black or blue. The atoms and bonds depicted in gray are not directly involved in the reaction.
that appear in either $L$ or $R$, typed by both their type in $L$ and their type in $R$, if defined. The notion of a graph transformation rule and its combined graph is illustrated in Figure 2.

![Figure 2](image)

**Figure 2:** (a) The graph transformation rule corresponding to the first step of the mechanism in Figure 1. The dotted lines represent the implicit non-bond constraints. E.g. the nitrogen and hydrogen atoms cannot be already bonded on the educt side, as the rule creates a new bond between them. (b) The combined graph of the rule in (a).

The application of a rule $p$ on a graph (mixture) $G$ can be broken up into two steps. First, finding an embedding of $L$ in $G$, the match $m$. The match not only has to respect the atom and bond types, but also what we refer to as non-bond constraints, typically between atoms that the rule creates a bond between, thus preventing parallel bonds. Second, what appears on the left side $L$ is removed and the contents of $R$ are added instead, to obtain the mixture $H$. Formally, the rule application is defined in the commutation of the following diagram:

$$
\begin{align*}
L &\leftarrow l - K - r \rightarrow R \\
&\downarrow m \quad \downarrow d \quad \downarrow h \\
G &\leftarrow D \rightarrow H
\end{align*}
$$

We call such transformations of $G$ into $H$ direct derivations, denoted by $G \xrightarrow{p,m} H$. By letting the result graph act as the host for successive applications, we can construct a sequence of direct derivations $G \xrightarrow{p_1,m_1} H_1 \xrightarrow{p_2,m_2} H_2, \ldots, H_{n-1} \xrightarrow{p_n,m_n} H_n$, referred to simply as a derivation, $G \Rightarrow^* H_n$.

Conveniently, a reaction mechanism converting the mixture of educt molecules $E$ into the mixture of product molecules $P$, can be formalized in terms of a derivation $E \Rightarrow^* P$. Each direct derivation in $E \xrightarrow{p_1,m_1} H_1 \xrightarrow{p_2,m_2} H_2, \ldots, H_{n-1} \xrightarrow{p_n,m_n} P$, defines an atom map sending each atom through the derivation. As a consequence, a mechanism defines a unique causal trajectory for each atom in $E$ into $P$.

**Overlay Graphs**

In this section we introduce and formalize the notion of an overlay graph (OG) of a reaction mechanism. An OG is essentially an abstraction of a reaction mechanism, where the causal information is omitted. However, all the transient modifications are preserved, which gives an account of the minimal topological structure immediately necessary for the execution of the reaction.

It turns out that graph transformation rules themselves are a suitable construction for such aggregation of the individual steps. As a mechanism is already defined by means of a derivation and thus, in particular, a sequence of rules, one can employ the rule composition framework and obtain representation of the whole mechanism as a single rule.

Just as the rest of the graph transformation framework, rule composition is defined within the category theory formalism. In principle, composition of two rules $p_1$ and $p_2$ into a single rule $p$ consists of ensuring two properties. The left side of $p$ should include the left side of $p_1$ as well as the part of the left side of $p_2$ which is not produced by $p_1$. Similarly, the right side of $p$ should contain the right side of $p_2$ as well as the part of the right side of $p_1$ which is not consumed by $p_2$.

In the following, we revisit the necessary rule composition notions in an informal way, using the rules corresponding to the first two steps of the mechanism in Figure 1 as an example. Readers are referred to the literature for the comprehensive treatment.

---

19
As aforementioned, the composed rule depends on the overlap between the right side of the first rule, and the left side of the second rule. In general, any such overlap can be chosen for rule composition, our rules, however, come from a derivation. For any two rules $p_i = (L_i \xrightarrow{l_i} K_i \xrightarrow{r_i} R_i)$ for $i \in \{1, 2\}$ we are composing, we thus also have a couple of direct derivations $D_i = H_{i-1} \xrightarrow{p_i,m_i} H_i$, again for $i \in \{1, 2\}$. The direct derivations give us the atom maps by means of the matches $m_1$ and $m_2$, and the corresponding right side embeddings $h_1$ and $h_2$. The overlap can thus be extracted in form of the subgraph of $H_1$ delimited by $h_1$ and $m_2$, denoted as $H$ in Figure 3.

The graph $H$ represents an interface between $R_1$ and $L_2$, and thus between the two rules. This allows us to construct the composed rule $p = p_1 \bullet H, p_2 = (L \xleftarrow{l} K \xrightarrow{r} R)$ where $L, K, R$ as well as $l$ and $r$ follow from the commutative diagram in Figure 4.

![Figure 3: Commutative diagram showing the construction of the graph $H$. Note that $R_1$ and $L_2$ are isomorphic in our case, however, the atom maps provided by the derivation differentiate between the two intermediate oxygen atoms, indicated by different positions. None of the oxygen atoms thus appears in the shared subgraph of $R_1$ and $L_2$, denoted $I$. On the other hand, both oxygen atoms appear in the interface graph $H$.](image-url)

More precisely, the graph $H$ marks the vertices shared by both $p_1$ and $p_2$. The rest of the composition is then an iterative process of extending this interface along the rules $p_1$ and $p_2$, via $C_1$ and $C_2$, respectively for $K_1$ and $K_2$, and finally $L$ and $R$, respectively, for $L_1$ and $R_2$. Finally, the graph $K$ is constructed as the structure shared by $C_1$ and $C_2$. Intuitively, all the constructions consist of completing directed squares within the diagram, and represent the construction of a smallest graph combining two others in the target corner, while the source corner specifies the shared subgraph.

The composition naturally gives rise to the direct derivation $H_0 \xrightarrow{p,m} H_2$ transforming $H_0$ directly into $H_2$ without the use of the intermediate graph $H_1$, called the composed derivation. This derivation is depicted in the bottom part of the commutative diagram in Figure 4.

The rule composition is well suited for the task of aggregating a mechanism as it keeps track of transient modifications. Take the non-bond constraints in $L$ and $R$ as an example, in particular, between the nitrogen and hydrogen atoms. The same non-bond constraint is implicit in the original rules $p_1$ (on the left side) and $p_2$ (on the right side), due to the bond being created or broken, respectively. The composed rule $p$, however, does not create nor break said bond, the inherited constraint being explicit. This property allows us to express that the action carried out by $p$ relies on the transient formation of the nitrogen-hydrogen bond, thus the bond cannot be already present in a matched mixture. Similarly, the carboxylic double bond broken by $p_1$ but reinstated by $p_2$ is present in both $L$ and $R$, however, its absence from $K$ indicates its transient breakage along the execution of $p$.

One might consider transient changes of formal charges to be similar. However, formal charges are encoded as changes in the type (label) of a vertex. As the vertices are not typed in $K$, the change of label is not reflected. This is in line with the transient change in formal charge not imposing a constraint on the applicability of the composed rule.

As each composed rule comes with its com-
Figure 4: Commutative diagram for rule composition of two rules $p_1 = (L_1 \xrightarrow{i_1} K_1 \xrightarrow{r_1} R_1)$ and $p_2 = (L_2 \xleftarrow{l_2} K_2 \xrightarrow{r_2} R_2)$ with direct derivations $D_1 = H_0 \overset{p_1,m_1}{\Longrightarrow} H_1$ and $D_2 = H_1 \overset{p_2,m_2}{\Longrightarrow} H_2$ respectively. We note that $h_1$ and $m_2$ are defined by $D_1$ and $D_2$ as illustrated by the commutative diagram of rule application. The resulting rule is $p = p_1 \bullet_{H_1} p_2 = (L \xleftarrow{l} K \xrightarrow{r} R)$ with $l = s_1 \circ w_1$ and $r = t_2 \circ w_2$. The colors from red to violet encode the order in which the individual graphs are inferred. Black color indicates the input graphs.
posed derivation, we can iterate the process until the entire mechanism is captured within a single rule.

**Definition 1 (Overlay Rule).** Let $M = E \xrightarrow{p_1, m_1} H_1 \xrightarrow{p_2, m_2} H_2, \ldots, H_{n-1} \xrightarrow{p_n, m_n} P$ be a reaction mechanism. Then the overlay rule $p(M)$ of $M$ is the rule $p(M) = p_1 \circ H_1 p_2, \ldots, p_{n-1} \circ H_{n-1} p_n$.

We refer to the corresponding composed derivation $D(M) = E \xrightarrow{p(M), m} P$ as the overlay derivation.

Overlay rules provide the formal support for the notion of an OG. To obtain a visualization akin to the original ITS concept, we construct the OG by collapsing the rule into a single graph. Similarly to the combined graph of a rule, the OG contains all atoms, bonds and even explicit non-bond constraints that appear on either side of the overlay rule. Instead of the usual before-and-after labels of the combined graph of a rule, OGs follow a colour scheme describing the treatment of each element during the mechanism. The action, permanent changes, is given in green and red for atoms and bonds that are introduced, respectively eliminated by the overlay rule. This includes increase, respectively, decrease in valence of a bond. Transient changes appear in black, for bonds that are temporarily broken or modified and blue for the explicit non-bond constraints, that is bonds that are temporarily created. An example OG of the mechanism from Figure 1 appears in Figure 5a.

As any other graph transformation rules, overlay rules are applicable to mixtures beyond the educt graph of the original reaction. As an overlay rule models the aggregated action of a mechanism, it also represents the necessary and sufficient condition for the execution of said mechanism. In other words, any application of an overlay rule bears witness to executability of the represented mechanism in a given mixture.

Of particular interest are direct derivations of the form $E \xrightarrow{p(M_1), m} P$ where $(E, P)$ is reaction different from the source of the mechanism $M_1$. For each such derivation, we can construct a mechanism $M_2$ consisting of the exact same

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5}
\caption{OGs of the triglyceride lipase mechanism from Figure 1. The bonds created or broken during the reaction are shown in green and red, respectively. Non-bond constraints are depicted in blue. The gray parts of the molecules are not part of the OG nor overlay rule and are only included to frame the reaction. (a) OG of the original mechanism. (b) The corresponding substrate OG. Observe that the hydrogen atom donated by the serine is green, meaning that the overlay rule “creates” it, while the hydrogen of the water molecule that is used to restore the serine is in red, effectively “erased” by the rule.}
\end{figure}
electron movements as $M_1$, remapped to the educt graph $E$. It is easy to see from Figure 1, that sources and targets of all electron movements appear within the overlay rule. Their location in $E$ is thus captured by the match $m$.

Overlay rules thus allow for discovery of potential mechanisms by means of a single rule application. Unlike previous approaches,21 overlay rules treat mechanism as an indivisible unit and do not allow for recombination. Applying a rule only once, however, bypasses much of the combinatorial explosion associated with the number of possible mixtures one can achieve by repeated application of rules.21

**Substrate Overlay Rules**

The overlay rules are universal, applicable to any reaction mechanism. Our application focuses being enzymatic reactions, we dedicate this section to treatment of overlay rules within the scope of enzymatic reactions.

The catalytic nature of the reactions does not directly impact the process of mechanism discovery and therefore the overlay rule can be still applied to any educt mixtures of choice. Since the mechanism, and in turn the overlay rule, captures the exact involvement of the catalytic components in the reaction (i.e., active amino acids and cofactors), the new educt graph has to specify the catalysts to match the overlay rule. Explicit specification of the catalytic moiety is very rare, however, especially for enzymatic reactions where the catalytic site of the enzyme might not be known at all.

The knowledge of the exact participation of the catalysts in the mechanism is invaluable for the interpretation and visualization of the reaction mechanism. Due to the cyclical nature of catalysis, however, the chemical structure of the catalyst appears untouched in the overall reaction. Explicit representation of the catalyst thus fails to add value to the overlay rules from the perspective of executability.

Instead, we introduce a reduced version of the rule called substrate overlay rule, where only the part of the mechanism that takes place within the substrate molecules is represented. Analogous to the representation of the catalytic components in the overall reaction schema, the role of the catalyst is made implicit. Substrate overlay rules are thus applicable to mixtures consisting exclusively of substrate molecules.

An obvious prerequisite to the construction of substrate overlay rules is the identification of catalytic components. As the energy landscape is abstracted away in our model, the usual definition of a catalyst is reduced to molecules that appear in the exact same configuration on both sides of the overall reaction. We formalize this notion by a graph $C$ of catalytic molecules contained in both $E$ and $P$ as a subgraph, identified by a pair of injections $C \xhookrightarrow{c} E$ and $C \xhookrightarrow{c} P$.

**Definition 2** (Substrate Overlay Rule). Let $p(M) = L \xhookrightarrow{l} K \xhookrightarrow{r} R$ be an overlay rule, $D = E \xhookrightarrow{p(M), m} P$ the overlay derivation and let $C$ be the graph of the catalytic components with $C \xhookrightarrow{c} E$ and $C \xhookrightarrow{c} P$.

Then the rule $L' \xhookrightarrow{l'} K' \xhookrightarrow{r'} R'$ where $L'$, $K'$, $R'$, $l' = u_1 \circ n_1$ and $r' = u_2 \circ n_2$ come from the commutative diagram in Figure 6 is a substrate rule of $M$.

In simple terms, the substrate overlay rule is obtained by only extracting the part of the overlay rule that corresponds to the substrate parts of $E$ and $P$. Figure 6 illustrates this four step process on the overlay rule in Figure 5. First, $E'$ and $P'$ are constructed as complements of $C$ within $E$ and $P$ (Figure 6 red), that is, as all the educt and product molecules that are not catalysts. In the next step, $L'$ is obtained as the shared part of $E'$ and $L$, and $R'$ is extracted from $P'$ and $R$ (Figure 6 yellow). $L'$ and $R'$ are thus projections of $L$ and $R$ onto the substrate parts of $E$ and $P$, respectively. Finally the same operation is used to obtain $K'$ as the shared part of $L'$, $R'$ and $K$. To be formally correct, this is done in two phases. First, $N_1$ and $N_2$ are built from $L'$ and $K$, respectively $R'$ and $K$ (Figure 6 green), followed by $K'$ itself from $N_1$ and $N_2$ (Figure 6 cyan). The construction relies on the same concepts as the diagram in Figure 4, especially extracting a shared portion of two graphs according to their embedding in a common supergraph.
Figure 6: Commutative diagram for inference of the substrate rule. $E'$ and $P'$ are the substrate part of the educt graph and product graph respectively. The left and right graphs $L'$ and $R'$ of the substrate overlay rule correspond to the shared part of $E'$ and $L$, respectively $P'$ and $R$. Similarly, $K'$ corresponds to the shared part of $L'$, $R'$ and $K$. The colours from red to cyan indicate the other in which the individual graphs are constructed. Black colour indicates the input graphs.
Separation of the substrate overlay rule construction into an educt and product side is crucial, since substrate and catalyst components might trade atoms. Consider for instance the 
OG depicted in Figure 5a. The depicted reaction is catalyzed by an enzyme, represented here as two amino acid side chains. One may be led to believe that the participation of the catalyst has to be via transient modifications (black and blue bonds). This is indeed the case for the histidine which serves as a temporary proton acceptor, illustrated by the non-bond constraints to the hydrogen atoms. However, the serine gives away a hydrogen atom to the triglyceride substrate only to be regenerated by a different hydrogen atom coming from a water molecule. This behavior is represented in the OG as non-transient modifications (red and green bonds).

Within the substrate overlay rule, we see the hydrogen atom of the serine being added, while the water hydrogen atom leaving to the serine is removed by the rule. We illustrate those operations with the same color scheme as the one used for bonds in the OG in Figure 5b. Although atoms are allowed to be added and removed, the catalytic component being regenerated guarantees that each substrate overlay rule is still mass invariant.

The atom exchange between the substrates and the catalytic components is not limited to hydrogen atoms. For instance, a description of a formyl-CoA transferase reaction from entry 155 in the M-CSA database, describes the aspartate 169 as being regenerated using a different oxygen atom than the original one.

One may further observe that the injections \( c_1 \) and \( c_2 \) are not necessarily unique if the educt or product mixtures contain several copies of the same molecule. While the function of an substrate overlay rule is unaffected by the choice of the injections \( c_1 \) and \( c_2 \), some choices lead to larger and unnecessarily complex rules, possibly obfuscating the chemical interpretation. A “most condensed” substrate overlay rule can be obtained using the knowledge of the atom map to coordinate the choice of \( c_1 \) and \( c_2 \).

**Results and Discussion**

**Obtaining Overlay Graphs**

To construct a sample set of OGs to study, we utilized the M-CSA database. The aim of this database is to provide a non-redundant overview of known enzymatic reaction mechanisms. Similarly to our understanding of a reaction mechanism, the M-CSA specifies mechanisms as sequences of elementary reaction steps. The reaction steps, however, do not explicitly encode the atom maps between the intermediates, products and the educt molecules. To separate the two notions, we thus refer to the M-CSA notion of a mechanism as *reaction sequence*. As of the time of writing, the M-CSA database contains 888 reaction sequences.

The reaction sequences of M-CSA are defined by means of electron pushing diagrams. While such a formalism provides atom maps between the educt and the product molecules of individual reaction steps, the links between the products of a step and the educts of the subsequent step are missing. Arguably, the graphical representation of the steps proposed by the M-CSA is sufficient to allow a human user to infer the atom maps. Such intent, however, lacks any formal support and leaves bond rotations between steps a distinct possibility. Additionally, the data behind the visualizations do not necessarily reflect the visual hints in a machine readable format, making any automated analysis laborious.

To ensure we correctly represented the intended mechanism, we considered all possible atom maps. This resulted in 2802 mechanisms, and consequently, overlay rules, corresponding to a total of 600 reaction sequences from the M-CSA. We observed that most of the reaction sequences (435) encode a unique mechanism, followed by the ones that only allowed two possibilities (100). On the other hand, we found five reaction sequences that could be explained by over 100 different mechanisms, and one reaction sequence totaling 964 non-isomorphic overlay rules.

The ambiguity arises from symmetries present in the educt and product graphs. Such
Symmetries are only significant for the overlay rule construction if they involve atoms with varied histories. Consider the example in Figure 7, illustrating how the atom map between two steps reflects in their composition. The first step attaches a hydrogen to an amine group $H_2N$, while the second cleaves a hydrogen atom from the resulting $H_3N^+$ group. All three hydrogen atoms of the $H_3N^+$ group are symmetrical, the letter rule can thus match any of them. The hydrogen atoms that were part of the $H_2N$ from the beginning have a different history than the hydrogen atom used to create the $H_3N^+$ group, resulting in two non-isomorphic mechanisms and in turn OGs.

The symmetries can generally be of two types. Symmetries of the molecule itself, or a part of it, like the hydrogen atoms of the amine group in Figure 7. Or symmetries arising between multiple instances of the same molecule. The latter is the case in the M-CSA entry 38, which includes up to four water molecules with different histories, contributing to the total of 964 overlay rules.

Many of the symmetries deal with disjoint sets of atoms and are thus independent. Hence, the number of possible non-isomorphic overlay rules grows exponentially with the number of symmetries. Even a partial atom map, which fixes at least one of the atoms involved in such symmetries, can therefore reduce the number of OGs exponentially. E.g., the M-CSA entry 186 has 24 possible mechanisms. However, mapping only four atoms, each in only one intermediate, is enough to identify a unique OG.

**Interpretation of OGs**

OGs being directly interpretable is a major strength of the formalism. While the causality information explicit in the reaction sequence is lost, OGs describe the entire reaction in a single picture, and anchor it within the educt molecules by means of the atom maps. To illustrate, we took a closer look at OGs of the subclass of enzyme reactions classified by the Enzyme Commission (EC) number EC 3.1.1.-.

The example from Figure 1 also belongs to this class. The subclass EC 3.1.1.- is well suited for our analysis as it provides a very topological description of the contained reactions, unlike many other EC classes.

In particular, hydrolase reactions (EC 3.-.-.-) identify water as one of the educt molecules, which is split into an alcohol group and a proton in the product molecules. The second level, EC 3.1.-.-, identifies an ester bond as the target for the hydrolase reaction to cleave. Finally, the third level, EC 3.1.1.-, specifies the cleaved ester bond to be part of a carboxylic ester. The carboxylic acyl group does not necessarily have to participate in the ester hydrolase reaction. For all 24 reaction sequences we obtain for the EC 3.1.1.- classified entries in the M-CSA, however, we observe the carboxylic acyl group taking on the role of an electron sink.

The EC 3.1.1.- reaction sequences in the M-CSA database contain on average little ambiguity. Only two reaction sequences allow for more than a single OG (two and five, respectively), for a total of 29 OGs. The ambiguity of the reaction sequence in both cases spans from the symmetries of hydrogen atoms of an $H_2O^+$ group, respectively an $H_3N^+$ group, similar to the example in Figure 7.

The structural characteristics of the EC 3.1.1.- reactions directly translate into patterns in the OGs and lead to a high degree of similarity. In particular, all obtained OGs share the same core, matching the substrate OG in Figure 5b. The core pattern appears in two variations. In Figure 5b, there are two hydrogen...
atoms traded with the catalyst. Some M-CSA reaction sequences, however, only specify one hydrogen atom being transferred from the water molecule to the former ester oxygen without the catalyst acting as an intermediary.

The generated OGs contain a number of differences outside of the common core characteristic of the EC 3.1.1.- class. These differences mostly pertain to the nature of the enzyme itself, in form of the active amino acids, as well as the exact nature of its involvement in the reaction. To this end, we identified two axes of interest along which the OGs, and by extension also the reactions they represent, differ.

First, while the duty of the electron sink falls exclusively to the carboxylic acyl group, the role of the electron source remains open. Most often, the latter role is taken on by the enzyme. In particular, by means of a histidine, or an aspartate, or even a two histidines acting in concert. However, we also observed OGs where the electron source was part of the substrate. Interestingly, in both such cases, the electron source, an N or O atom, was not directly adjacent to the reaction center and was regenerated to its original configuration, mirroring the actions of the enzyme. This points towards a more fine-grained definition of catalysis, in which not a whole molecule, but rather a functional part could be considered a catalyst.

The second difference spans from the necessity to lock the substrate in place by linking it to one of the amino acids by a covalent bond (described hereafter as docking). Most often we observed the substrate carbon atom of the ester bond being bound to a serine, alternatively an aspartate or a threonine. Four reaction sequences, however, completely omit such docking.

To illustrate, we depict an example for each combination of the above two factors in Figure 8. The example in the top left showcasing the most complex involvement of the enzyme, responsible for both providing the electron source and docking the substrate, is the most most prevalent one, present in 19 (66%) of the 29 OGs. Moreover, all the relevant OGs come from reaction sequences that described them uniquely, giving us 19 (79%) out of all 24 reaction sequences which exhibited such behavior.

In opposition, the bottom right OG in Figure 8 showcases the least possible involvement of the enzyme, not providing an electron source nor being used for docking. In fact, the role of the enzyme in the bottom right OG is fully passive, limited to the creation of suitable physico-chemical environment. Only two OGs, derived from a single reaction sequence match this pattern.

Our observations suggest that the enzyme function can be simplified in case part of the function can be outsourced to the substrate, or alternatively, made passive.

Using Overlay Rules to Explain Reactions by a Known Mechanism

One of the benefits of the graph transformation formalism is the executability of the overlay rules and thus, implicitly, the mechanisms. Taking advantage of this property, we took the overlay rules obtained for the M-CSA entries classified as EC 3.1.1.- and used them to pro-
pose reaction mechanisms—including catalytic components—as an explanation for other reactions, specified simply by the overall reaction description \((E, P)\). To this end we extracted over 13,000 overall reaction descriptions from the Rhea database, an expert curated database of biologically interesting reactions.\(^{23}\)

The overall reaction descriptions in the Rhea database do not explicitly list the catalytic components. We therefore used the substrate overlay rules and adopted the catalytic context of the original reactions. As observed in the previous section, there is a high degree of similarity between the overlay rules for the reactions in EC 3.1.1.-. Indeed, 18 overlay rules are isomorphic to each other on the substrate level, Figure 5b depicting one of them. In total, there are only 11 substrate overlay rules unique up to isomorphism among the original 29.

The similarity of the overlay rules does not stop at isomorphisms. Three of the substrate overlay rules are a target of a monomorphism from another rule, effectively only refinements of the latter, having the same action but a larger invariant graph. Such rules are guaranteed to only apply if the smaller rules apply as well, and produce the same result. We therefore discarded such rules, while retaining all possible product mixtures.

A closer inspection further revealed that one reaction sequence cleaves a C–N bond despite being marked as EC 3.1.1.- in the M-CSA. Whether a case of misclassification or an oversight in the M-CSA database, we discarded the resulting substrate overlay rule to avoid introducing noise into our results.

Finally, we introduced a manual change to the 7 remaining substrate overlay rules. The modification spans from a different convention for representation of carboxylic acid, one of the ester hydrolase products, between the M-CSA and Rhea. While the M-CSA, and by extension our rules, use a COOH group, Rhea uses exclusively COO\(^-\) and a free floating proton, H\(^+\), for each reaction marked as EC 3.1.1.-. The modified rules thus additionally cleave the O–H bond of the carboxylic acid. As the carboxylic acid O–H bond comes from the water molecule, the modified rules have a necessarily bigger context, fully specifying the water molecule on the educt side.

Using the 14 substrate overlay rules, both the originals and the manual revision, we reproduced 588 different reactions from Rhea. 97 (16.5\%) of the matches were indeed classified as EC 3.1.1.- within the Rhea database. We thus retrieved 74.6\% of the 130 EC 3.1.1.- reaction in Rhea, all using the modified rules. We further found 30 (5.1\%) reactions classified in different EC classes and 461 (78.4\%) reactions which were assigned no EC class in Rhea. These results are summarized in Table 1.

A manual cursory inspection of a sample of the 461 reactions with no EC class in Rhea database found the reactions visually corresponding to carboxylic ester hydrolases. The investigated reactions equipped with the relevant cross-link were indeed classified as EC 3.1.1.- in the UniProt database,\(^{24}\) e.g. Rhea:17794 or Rhea:32968.

The majority of the reactions classified differently than EC 3.1.1.- we explain are transferases that cleave one ester bond and create a new one using an alcohol group of the other educt molecule. The alcohol group acting as a substitute for the educt water used in the hydrolases. All these matches result from the original substrate overlay rules, as the water molecule is fully specified in the modified version.

Furthermore, there are 33 EC 3.1.1.- reactions our substrate overlay rules did not capture. 14 (42.4\%) were due to encoding issues resulting from periodicity not reflected in the smiles strings or use of different levels of abstraction or tautomerisation between the educts and products in the Rhea database. Further, 5 (15.2\%) reactions involved interactions with iron ions. Direct metal ion interactions are currently not supported by our methods.

Finally, the remaining 14 (42.4\%) contained extra modifications of the products of the carboxylic ester bond cleavage. We detail the types of additional chemistry observed in Table 2.

We explored in greater detail the most prevalent modification, keto-enol tautomerisation. M-CSA database contains two elementary reaction steps corresponding to keto-enol tau-
Table 1: The EC classification of the reactions explained by our substrate overlay rules as declared in the Rhea database.

| EC class   | Incidence | Description                                      |
|------------|-----------|--------------------------------------------------|
| None       | 461       |                                                  |
| EC 3.1.1.- | 97        | Carboxylic ester hydrolases                      |
| EC 2.3.1.- | 19        | Aroytransferases other than amino-acyl           |
| EC 3.6.1.- | 3         | Phosphorus-containing acid anhydrate hydrolases  |
| EC 2.3.2.- | 2         | Aminoacyltransferases                            |
| EC 4.2.99.-| 2         | Other carbon-oxygen lyases                       |
| EC 5.4.1.- | 2         | Intramolecular acyl transferases                 |
| EC 2.1.3.- | 1         | Carboxy-/carbomyltransferases                    |
| EC 3.2.1.- | 1         | Glycosidases                                     |

Table 2: Types of additional modifications to the product molecules of the carboxylic ester bond cleavage observed in the Rhea database, with incidence.

| Description                      | Incidence |
|----------------------------------|-----------|
| Keto-enol tautomerization        | 5         |
| Second carboxylic ester bond cleavage | 3     |
| Decarboxylation                  | 3         |
| Cyclisation                      | 2         |
| Release of sulfite group         | 1         |

romerisation, step 4 in entry 85 and step 4 in entry 463, both described as spontaneous tautomerisation occurring outside of the enzyme active site. To this end, we enriched our rules by composing them with the steps corresponding to the spontaneous keto-enol tautomerisation. The compositions were performed on the ester oxygen, which, instead of an alcohol group, ends up in a ketone group.

Using the composed rules, we were able to explain four of the five Rhea reactions, one eluding due to the keto-enol tautomerisation occurring twice, on the former ester oxygen as well as an alcohol group already present in the educt molecule. The composed rules, however, failed to increase our coverage of the Rhea database outside of the four motivating EC 3.1.1.- reactions.

Outlook

We utilize graph transformation formalism to define reaction mechanisms with no space for ambiguity on the topological level. We demonstrate how said formalism can be used to extend the concept of an ITS from simple reactions to reaction mechanisms, fully capturing the transient behaviour. We demonstrate the utility of said generalization in two areas. For interpretation and visualization in form of an OG and for proposal of reaction mechanisms, including reactions without known catalytic context, in the form of overlay rules. The application potential, however, stretches much further.

The identification of a common patterns within the OGs of EC 3.1.1.- reactions suggests the possibility of using the OGs for an automated enzymatic reaction classification. The identification of patterns within OGs does not stop at classification either, using a suitable recombination framework would allow one to design novel mechanisms within the OG formalism. Last but not least, one should appreciate the notion of OGs being strictly rigorous, attributing a level of precision to the results and signifying abstraction potential.

Author Information

Corresponding Author

Daniel Merkle – Department of Mathematics and Computer Science, University
of Southern Denmark, Odense, Denmark; orcid.org/0000-0001-7792-375X; Email: daniel@imada.sdu.dk

Authors

Jakob L. Andersen – Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark; orcid.org/0000-0002-4165-3732

Rolf Fagerberg – Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark; orcid.org/0000-0003-1004-3314

Christoph Flamm – Department of Theoretical Chemistry, University of Vienna, Vienna, Austria; orcid.org/0000-0001-5500-2415

Walter Fontana – Department of Systems Biology, Harvard Medical School, Boston, MA, USA; orcid.org/0000-0003-4062-9957

Juraj Kolčák – Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark; orcid.org/0000-0002-9407-9682

Christophe V.F.P. Laurent – Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark; orcid.org/0000-0002-9112-6981

Nikolai Nøjgaard – Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark; orcid.org/0000-0002-7053-4716

Acknowledgments

This work is supported by the Novo Nordisk Foundation grant NNF19OC0057834 and by the Independent Research Fund Denmark, Natural Sciences, grants DFF-0135-00420B and DFF-7014-00041.

References

(1) Olivo, G.; Capocasa, G.; Del Giudice, D.; Lanzalunga, O.; Di Stefano, S. New horizons for catalysis disclosed by supramolecular chemistry. Chem. Soc. Rev. 2021, 50, 7681–7724.

(2) Twigg, M. V. Catalyst handbook; Routledge, 2018.

(3) Ribeiro, A. J. M.; Holliday, G. L.; Furnham, N.; Tzyzack, J. D.; Ferris, K.; Thornton, J. M. Mechanism and Catalytic Site Atlas (M-CSA): a database of enzyme reaction mechanisms and active sites. Nucleic Acids Research 2017, 46, D618–D623.

(4) Fujita, S. Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. J. Chem. Inf. Comput. Sci. 1986, 26, 205–212.

(5) Ehrig, H.; Ehrig, K.; Golas, U.; Taentzer, G. In Monographs in Theoretical Computer Science. An EATCS Series; Brauer, W., Rozenberg, G., Salomaa, A., Eds.; Springer: Berlin and Heidelberg, Germany, 2006.

(6) Habel, A.; Müller, J.; Plump, D. Double-pushout graph transformation revisited. Mathematical Structures in Computer Science 2001, 11, 637–688.

(7) Andersen, J. L.; Flamm, C.; Merkle, D.; Stadler, P. F. In Graph Transformation. ICGT 2016. Lecture Notes in Computer science; Echahed, R., Minas, M., Eds.; Springer: Cham, Switzerland, 2016; Vol. 9761; pp 73–88.

(8) Hendrickson, J. B. Comprehensive System for Classification and Nomenclature of Organic Reactions. J. Chem. Inf. Comput. Sci. 1997, 37, 852–860.

(9) Krebs, H.; Kay, J.; Kay, J.; Symposium, B. S. G. B.; Weitzman, P.; Britain), B. S. G. Krebs’ Citric Acid Cycle: Half a Century and Still Turning;
Biochemical Society symposia; Biochemical Society, 1987.

(10) Zubay, G. L.; Parson, W. W.; Vance, D. E. Principles of biochemistry; Wm. C. Brown, 1995.

(11) Ehrig, K.; Heckel, R.; Lajios, G. Molecular Analysis of Metabolic Pathway with Graph Transformation. Graph Transformations. Berlin, Heidelberg, 2006; pp 107–121.

(12) Ehrig, H.; Pfender, M.; Schneider, H. J. Graph-grammars: An algebraic approach. 14th Annual Symposium on Switching and Automata Theory (SWAT 1973). USA, 1973; pp 167–180.

(13) Silva, W. M. C. d.; Andersen, J. L.; Holanda, M. T.; Walter, M. E. M. T.; Brigido, M. M.; Stadler, P. F.; Flamm, C. Exploring Plant Sesquiterpene Diversity by Generating Chemical Networks. Processes 2019, 7.

(14) Andersen, J. L.; Flamm, C.; Merkle, D.; Stadler, P. F. In silico Support for Eschenmoser’s Glyoxylate Scenario. Israel Journal of Chemistry 2015, 55, 919–933.

(15) Andersen, J. L.; Merkle, D.; Rasmussen, P. S. Combining Graph Transformations and Semigroups for Isotopic Labeling Design. Journal of Computational Biology 2020, 27, 269–287.

(16) Andersen, J. L.; Fagerberg, R.; Flamm, C.; Kianian, R.; Merkle, D.; Stadler, P. F. Towards Mechanistic Prediction of Mass Spectra Using Graph Transformation. MATCH, Communications in Mathematical and in Computer Chemistry 2018, 80, 705–731.

(17) Andersen, J. L.; Andersen, T.; Flamm, C.; Hanczyc, M. M.; Merkle, D.; Stadler, P. F. Navigating the Chemical Space of HCN Polymerization and Hydrolysis: Guiding Graph Grammars by Mass Spectrometry Data. Entropy 2013, 15, 4066–4083.

(18) Andersen, J. L.; Flamm, C.; Merkle, D.; Stadler, P. F. Chemical Transformation Motifs — Modelling Pathways as Integer Hyperflows. IEEE/ACM Transactions on Computational Biology and Bioinformatics 2019, 16, 510–523.

(19) Andersen, J. L.; Flamm, C.; Merkle, D.; Stadler, P. F. Rule Composition in Graph Transformation Models of Chemical Reactions. MATCH, Communications in Mathematical and in Computer Chemistry 2018, 80, 661–704.

(20) Behr, N.; Krivine, J.; Andersen, J. L.; Merkle, D. Rewriting theory for the life sciences: A unifying theory of CTMC semantics. Theoretical Computer Science 2021, 884, 68–115, TR: https://arxiv.org/abs/2106.02573.

(21) Andersen, J. L.; Fagerberg, R.; Flamm, C.; Fontana, W.; Kolčák, J.; Laurent, C. V. F. P.; Merkle, D.; Nojgaard, N. Graph transformation for enzymatic mechanisms. Bioinformatics 2021, 37, i392–i400.

(22) EMBL-EBI, M-CSA API. accessed on 24/09/2021; https://www.ebi.ac.uk/thornton-srv/m-csa/api/entries/?format=json.

(23) Lombardot, T.; Morgat, A.; Axelsen, K. B.; Aimo, L.; Hyk-Onospikel, N.; Niknejad, A.; Ignatchenko, A.; Xenarios, I.; Coudert, E.; Redaschi, N.; Bridge, A. Updates in Rhea: SPARQLing biochemical reaction data. Nucleic Acids Research 2019, 47, D596–D600.

(24) Consortium, T. U. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Research 2020, 49, D480–D489.