A categorical approach to synthetic chemistry

Ella Gale
University of Bristol, UK
ella.gale@bristol.ac.uk

Leo Lobski
University College London, UK
leo.lobski.21@ucl.ac.uk

Fabio Zanasi
University College London, UK
University of Bologna, Italy
f.zanasi@ucl.ac.uk

We introduce a mathematical framework for retrosynthetic analysis, an important research method in synthetic chemistry. Our approach represents molecules and their interaction using string diagrams in layered props – a recently introduced categorical model for partial explanations in scientific reasoning. Such principled approach allows one to model features currently not available in automated retrosynthesis tools, such as chirality, reaction environment and protection-deprotection steps.

1 Introduction

A chemical reaction can be understood as a rule which tells us what the outcome molecules (or molecule-like objects, such as ions) are when several molecules are put together. If, moreover, the reaction records the precise proportions of the molecules as well as the conditions for the reaction to take place (temperature, pressure, concentration, presence of a solvent etc.), it can be seen as a precise scientific prediction, whose truth or falsity can be tested in a lab, making the reaction reproducible. Producing complicated molecules, as required e.g. by the pharmaceutical industry, requires, in general, a chain of several consecutive reactions in precisely specified conditions. The general task of synthetic chemistry is to come up with reproducible reaction chains to generate previously unknown molecules (with some desired properties) [38]. Successfully achieving a given synthetic task requires both understanding of the chemical mechanisms and the empirical knowledge of existing reactions. Both of these are increasingly supported by computational methods [33]: rule-based and dynamical models are used to suggest potential reaction mechanisms, while database search is used to look for existing reactions that would apply in the context of interest [34]. The key desiderata for such tools are tunability and specificity. Tunability endows a synthetic chemist with tools to specify a set of goals (e.g. adding or removing a functional group[1]), while by specificity we mean maximising yield and minimising side products.

In this paper, we focus on the area of synthetic chemistry known as *retrosynthesis* [15, 34, 37]. While reaction prediction asks what reactions will occur and what outcomes will be obtained when some molecules are allowed to interact, retrosynthesis goes backwards: it starts with a target molecule that we wish to produce, and it proceeds in the "reverse" direction by asking what potential reactants would produce the target molecule. While many automated tools for retrosynthesis exist (see e.g. [24, 13, 12, 25, 9, 35, 18]), there is no uniform mathematical framework in which the suggested algorithms could be analysed, compared or combined. The primary contribution of this paper is to provide such a framework. By formalising the methodology at this level of mathematical generality, we are able to provide insights into how to incorporate features that the current automated retrosynthesis tools lack: these include modelling chirality, the reaction environment, and the protection-deprotection steps (see for example [19]), which are all highly relevant to practical applications. Our formalism, therefore, paves the way for new automated retrosynthesis tools, accounting for the aforementioned features.

[1] Part of a molecule that is known to be responsible for certain chemical function.

Submitted to:
ACT 2023

Mathematically, our approach is phrased in the algebraic formalism of *string diagrams*, and most specifically uses *layered props*. Layered props were originally introduced, in [26], as models for systems that have several interdependent levels of description. In the context of chemistry, the description levels play a twofold role: first, each level represents a reaction environment, and second, the rules that are available in a given level reflect the structure that is deemed relevant for the next retrosynthetic step. The latter can be seen as a kind of coarse-graining, where by deliberately restricting to a subset of all available information, we reveal some essential features about the system. Additionally, organising retrosynthetic rules into levels allows us to include conditions that certain parts of a molecule are to be kept intact. While the presentation here is self-contained and, in particular, does not assume background on layered props, we emphasise that our approach is principled in the sense that many choices we make are suggested by this more general framework. We point such choices out when we feel the intuition that comes from layered props is helpful for understanding the formalism presented in the present work.

The rest of the paper is structured as follows. In Section 2 we give a brief overview of the methodology of retrosynthetic analysis as well as the existing tools for automating it. Section 3 recalls the conceptual and mathematical ideas behind layered props. The entirety of Section 4 is devoted to constructing the labelled graphs that we use to represent molecules and parts of molecules: these will be the generating objects of the monoidal categories we introduce in Section 5, where we also exhibit these categories as components of a layered prop. Section 6 describes how to reason about retrosynthesis within the layered prop introduced in the preceding section. In Section 7 we outline how our formalism can be used for comparing existing automated retrosynthesis tools, as well as for designing new ones.

2 Retrosynthetic analysis

Retrosynthetic analysis starts with a target molecule we wish to produce but do not know how. The aim is to "reduce" the target molecule to known (commercially available) outcome molecules in such a way that when the outcome molecules react, the target molecule is obtained as a product. This is done by (formally) partitioning the target molecule into functional parts referred to as *synthons*, and finding actually existing molecules that are chemically equivalent to the synthons; these are referred to as *synthetic equivalents* [16, 38, 10]. If no synthetic equivalents can be found, the partitioning step can be repeated, this time using the synthons as the target molecules, and the process can continue until either synthetic equivalents are found or a maximum number of steps is reached and the search is stopped. Note that the synthons themselves do not refer to any molecule as such, but are rather a convenient formal notation for parts of a molecule. For this reason, passing from synthons to synthetic equivalents is a non-trivial step involving intelligent guesswork and chemical know-how of how the synthons *would* react if they were independent chemical entities.

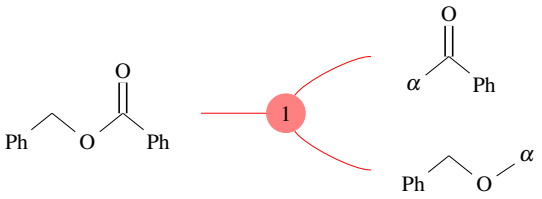
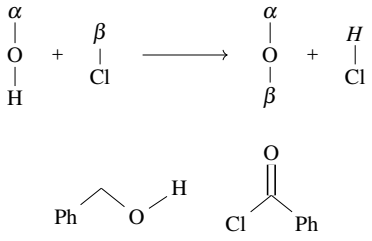


Figure 1: A retrosynthetic disconnection.

Clayden, Warren and Greeves [10] give the example in Figure 1 when introducing retrosynthesis. Here the molecule on the left-hand side is the target, and the resulting two parts on the right-hand side are the synthons. We use variable α to indicate where the cut has been made. Since there is a reaction shown on the right (top picture), the synthons are chemically equivalent to the



molecules (bottom picture). This is the simplest possible instance of a retrosynthetic sequence. In general, the interesting sequences are much longer, and, importantly, contain information under what conditions a reaction will take place.

Existing tools Many tools for automatic retrosynthesis have been successfully developed starting from the 1960s [24, 13, 25, 9, 35]. They can be divided into two classes [34]: *template-based* [20, 39] and *template-free* [25, 32]. Template-based tools contain a rule database (the *template*), which is either manually encoded or automatically extracted. Given a molecule represented as a graph, the model checks whether any rules are applicable to it by going through the database and comparing the conditions of applying the rule to the subgraphs of the molecule [34]. Choosing the order in which the rules from the template and the subgraphs are tried are part of the model design. Template-free tools, on the other hand, are data-driven and treat the retrosynthetic rule application as a translation between graphs or their representations as strings: the suggested transforms are based on learning from known transforms, avoiding the need for a database of rules [34, 35].

While successful retrosynthesic sequences have been predicted by the computational retrosynthesis tools, they lack a rigorous mathematical foundation, which makes them difficult to compare, combine or modify. Other common drawbacks of the existing approaches include not including the reaction conditions or all cases of chirality as part of the reaction template [34, 25], as well as the fact that the existing models are unlikely to suggest protection-deprotection steps. Additionally, the template-free tools based on machine learning techniques sometimes produce output that does not correspond to molecules in any obvious way, and tend to reproduce the biases present in the literature or a data set [34].

For successful prediction, the reaction conditions are, of course, crucial. These include such factors as temperature and pressure, the presence of a solvent (a compound which takes part in the reaction and whose supply is essentially unbounded), the presence of a reagent (a compound which is unaltered by the reaction, but without which the reaction would not occur), as well as the presence of a catalyst (a compound which increases the rate at which the reaction occurs, but is itself unaltered by the reaction). The above factors can change the outcome of a reaction dramatically [29, 14]. There have indeed been several attempts to include reaction conditions into the forward reaction prediction models [27, 21, 36, 28]. However, the search space in retrosynthesis is already so large that adding another criterion to be searched over should be done with caution. A major challenge for predicting reaction conditions is that they tend to be reported incompletely or inconsistently in the reaction databases [11].

Chirality (mirror-image asymmetery) of a molecule can alter its chemical and physiological properties, and hence constitutes a major part of chemical information pertaining to a molecule. While template-based methods have been able to successfully suggest reactions involving chirality (e.g. [13]), the template-free models have difficulties handling it [25]. This further emphasises usefulness of a framework which is able to handle both template-based and template-free models.

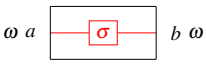
The protection-deprotection steps are needed when more than one functional group of a molecule A would react with a molecule B. To ensure the desired reaction, the undesired functional group of A is first "protected" by adding a molecule X, which guarantees that the reaction product will react with B in the required way. Finally, the protected group is "deprotected", producing the desired outcome of B reacting with the correct functional group of A. So, instead of having a direct reaction $A + B \to C$ (which would not happen or would happen imperfectly due to a "competing" functional group), the overall reaction chain looks like:

$$(1)\ A + X \to A' \quad \text{(protection)}, \quad (2)\ A' + B \to C', \quad (3)\ C' + Y \to C \quad \text{(deprotection)}.$$

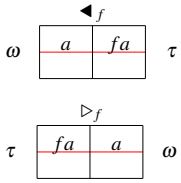
The trouble with the protection-deprotection steps is that they temporarily make the molecule larger, which means that an algorithm whose aim is to make a molecule smaller will not suggest them.

3 Layered props

Layered props were introduced in [26] as categorical models for diagrammatic reasoning about systems with several levels of description. They have been employed to account for partial explanations and semantic analysis in the context of electrical circuit theory, chemistry, and concurrency. Formally, a layered prop is essentially a functor $\Omega : P \to \mathbf{StrMon}$ from a poset P to the category of strict monoidal categories, together with a right adjoint for each monoidal functor in the image of Ω. Given $\omega \in P$, we denote a morphism $\sigma : a \to b$ in $\Omega(\omega)$ by the box on the right. We think of σ as a *process* with an input a and an output b happening in the *context* ω. Note, however, that these diagrams are not merely a convenient piece of notation that capture our intuition: they are a completely formal syntax of string diagrams, describing morphisms in a certain subcategory of pointed profunctors [26].



The monoidal categories in the image of Ω are thought of as languages describing the same system at different levels of granularity, and the functors are seen as translations between the languages. Given $\omega \leq \tau$ in P, let us write $f \coloneqq \Omega(\omega \leq \tau)$. Then, for each $a \in \Omega(\omega)$ we have the morphisms drawn on the right. The reason for having morphisms in both directions is that we want to be able to "undo" the action of a translation while preserving a linear reasoning flow. The two morphisms will not, in general, be inverse to each other: rather, they form an adjoint pair. This corresponds to the intuition that some information is gained by performing the translation, and that the translation in the reverse direction is our best guess, or an approximation, not a one-to-one correspondence. For this reason, we refer to $\blacktriangleleft$ as *refinement* and to $\triangleright$ as *coarsening*.



There are two ways to compose morphisms in parallel in a layered prop: internally within a monoidal category $\Omega(\omega)$ using its own monoidal product (composition inside a context), and externally using the Cartesian monoidal structure of **StrMon** (doing several processes in different contexts in parallel). We represent the latter by stacking the boxes on top of each other. Additional morphisms of a layered prop ensure that the internal and the external monoidal structures interact in a coherent way. Finally, a layered prop comes with "deduction rules" (2-cells) which allow transforming one process into another one. We refer the reader to [26] for the details.

In this work, the processes in context will be the retrosynthetic disconnection rules and the chemical reactions. The context describes the reaction environment as well as the level of granularity at which the synthesis is happening (i.e. what kinds of disconnection rules are available). The objects in the monoidal categories are given by molecules and parts of molecules: this is the subject of Section 4. Section 5 formally defines the disconnection rules and the chemical reactions, which are, in turn, used to generate the layered prop of interest.

4 Molecule partitions

By a *molecule partition* we mean a molecule that has been cut along some bonds. Mathematically, we define a molecule partition as a (connected) labelled graph whose edge labels are the number of electron pairs shared by the covalent bond, and whose vertex labels are either atoms, charges or free variables (Definition 1). A molecule is then defined as a molecule partition with no charges or free variables (Definition 3). Ionic compounds can be defined as collections of molecule partitions with no free variables whose charges cancel out (Definition 6). In order to account for chirality, we add spatial information to a molecule partition, making it an *oriented molecule partition* (Definition 12).

Oriented molecule partitions together with the reaction context form the objects of the layered prop we suggest as a framework for synthetic chemistry. The morphisms of this layered prop correspond to retrosynthetic disconnection rules and chemical reactions; this is the topic of the next section.

Let us define the set of *atoms* as containing the symbol for each main-group element of the periodic table: $\mathrm{At} \coloneqq \{H, C, O, P, \dots\}$. Define the function $\mathbf{v} : \mathrm{At} \sqcup \{+,-\} \sqcup \mathrm{FW} \to \mathbb{N}$ as taking each element symbol to its valence[2], and define $\mathbf{v}(-) = \mathbf{v}(+) = \mathbf{v}(\alpha) = 1$.

Definition 1 (Molecule partition)**.** A *molecule partition* is a triple (V, τ, m), where V is a finite set of *vertices*, $\tau : V \to \mathrm{At} \sqcup \{+,-\} \sqcup \mathrm{FW}$ is a function taking each vertex to its *label* and $m : V \times V \to \mathbb{N}$ is a function satisfying the following conditions:

- there is at least one $v \in V$ such that $\tau(v) \in \mathrm{At} \sqcup \{+,-\}$,
- for all $v \in V$, we have $m(v,v) = 0$,
- for all $v, w \in V$, we have $m(v,w) = m(w,v)$,
- for all $v, u \in V$ with $v \neq u$, there are $w_0, \dots, w_n \in V$ such that $w_0 = v$ and $w_n = u$ and $m(w_{i-1}, w_i) \neq 0$ for each $i = 1, \dots, n$,
- for all $v \in V$, we have $\sum_{u \in V} m(u,v) = \mathbf{v}\tau(v)$.

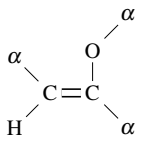
In other words, the integers $m(i,j)$ form an adjacency matrix of an irreflexive, symmetric and connected graph (where a non-zero number indicates an edge with that integer as the label), and the sum of each row or column gives the valence of the (label of) corresponding vertex.

When drawing a molecule partition (V, τ, m), we simply replace the vertices in V by their labels (see Example 5). Most of the time, we draw molecule partitions up to the isomorphism of labelled graphs (Definition 2), unless the precise labelling of the elements of V plays a role. We adopt the usual chemical notation for n-ary bonds by drawing them as n parallel lines. Chemically, the vertices labelled by the free variable α should be read as free bonds, allowing the partition to bind to other partitions.

Definition 2 (Labelled graph isomorphism)**.** A *labelled graph isomorphism* (or simply *isomorphism* for short) from a molecule partition (V, τ, m) to (W, κ, n) is an isomorphism of sets $f : V \to W$ such that $\kappa f = \tau$ and $n(f \times f) = m$.

Definition 3 (Molecule)**.** We say that a molecule partition (V, τ, m) is a *molecule* if the image of the function τ is contained in At.

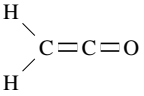
A molecule is just a molecule partition with no free variables or charges. This corresponds to molecules in the chemical sense, except for allowing formal, non-existing molecules.



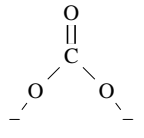
Definition 4 (Ion)**.** We say that a molecule partition (V, τ, m) is an *ion* if

- the image of the function τ is contained either in $\mathrm{At} \sqcup \{+\}$ or $\mathrm{At} \sqcup \{-\}$ (but not in both),
- there is at least one $v \in V$ with $\tau(v) \in \{+,-\}$.

In words, an ion is a molecule partition with no free variables and a non-zero number of charges all of the same sign.



Example 5. We give examples of a molecule partition, a molecule (ethenone) and an ion (carbonate anion) on the right. In order to define the topmost molecule partition precisely, we have to fix a labelling for the vertex set, e.g. $V = \{1,2,3,4,5,6,7\}$, and then put $\tau(1) = \tau(5) = \tau(7) = \alpha$, $\tau(2) = \mathrm{H}$, $\tau(3) = \tau(4) = \mathrm{C}$, $\tau(6) = \mathrm{O}$, $m(1,3) = m(2,3) = m(4,5) = m(4,6) = m(6,7) = 1$, $m(3,4) = 2$, $m(x,y) = m(y,x)$ and $m(x,y) = 0$ for all other pairs. However, as mentioned above, we usually omit these details.



[2]This is a bit of a naive model, as valence is, in general, context-sensitive and not determined by a single atom. We leave accounting for this to future work.

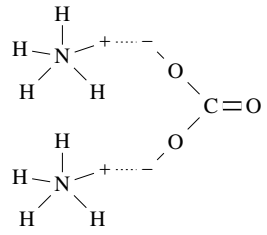
Given a multiset of molecule partitions I, let us write V_I for the disjoint union of the vertices of the molecule partitions in I, and $\tau_I : V_I \to \mathrm{At} \sqcup \{+,-\} \sqcup \mathrm{FW}$ for the labelling function which is determined by the labelling functions of the individual molecule partitions.

Definition 6 (Ionic compound)**.** An *ionic compound* is a pair (I, i), where I is a finite multiset of ions and $i \subseteq V_I \times V_I$ is a symmetric relation such that

- for every $v \in V_I$ with $\tau_I(v) \in \{+,-\}$, there is a unique $w \in V_I$ with $i(v,w)$,
- if $i(v,w)$, then $\tau_I(v), \tau_I(w) \in \{+,-\}$ and $\tau_I(v) \neq \tau_I(w)$,
- for any pair M, M' of ions in I, either $M = M'$, or there is a sequence $M = M_0, M_1, \ldots, M_n = M'$ in I such that for all $i = 0, \ldots, n-1$ there are vertices $v \in M_i$ and $w \in M_{i+1}$ with $i(v,w)$.

Note that, since ions are connected graphs, the last condition in the definition of an ionic compound is equivalent to saying that the disjoint union of the graphs together with the new relation i is connected. Graphically, we represent the relation i as a dashed line. This represents an ionic bond.

Example 7. An ionic compound is a collection of ions such that the sum of the negative charges is equal to the charge of the positive charges, and the opposite charges can be matched in a way that creates only one compound (i.e. the resulting graph is connected). For example, ammonium carbonate is represented on the right.



Chirality Next, we introduce (rudimentary) spatial information into our model of molecule partitions. The idea is to record for each triple of atoms whether they are on the same line or not, and similarly, for each quadruple of atoms whether they are in the same plane or not.

Definition 8 (Plane relation)**.** Let S be a set. We call a ternary relation $\mathcal{P} \subseteq S \times S \times S$ a *plane relation* if the following hold for all elements A, B and C of S:

- $ABB \notin \mathcal{P}$,
- if $\mathcal{P}(ABC)$ and $\mathfrak{p}(ABC)$ is any permutation of the three elements, then $\mathcal{P}(\mathfrak{p}(ABC))$.

Definition 9 (Tetrahedron relation)**.** Let S be a set, and let $\mathcal{P}$ be a fixed plane relation on S. We call a quaternary relation $\mathcal{T} \subseteq S \times S \times S \times S$ a *tetrahedron relation* if the following hold for all elements A, B, C and D of S:

- if $\mathcal{T}(ABCD)$, then $\mathcal{P}(ABC)$,
- if $\mathcal{T}(ABCD)$ and $\mathfrak{p}(ABCD)$ is any even permutation of the four elements, then $\mathcal{T}(\mathfrak{p}(ABCD))$.

Unpacking the above definitions, a plane relation is closed under the action of the symmetric group S_3 such that any three elements it relates are pairwise distinct, and a tetrahedron relation is closed under the action of the alternating group A_4 such that if it relates some four elements, then the first three are related by some (fixed) plane relation (this, inter alia, implies that any related elements are pairwise distinct, and their any 3-element subset is related by the fixed plane relation).

The intuition is that the plane and tetrahedron relations capture the spatial relations of (not) being on the same line or plane: $\mathcal{P}(ABC)$ stands for A, B and C not being on the same line, that is, determining a plane; similarly, $\mathcal{T}(ABCD)$ stands for A, B, C and D not being in the same plane, that is, determining a tetrahedron. The tetrahedron is moreover oriented: $\mathcal{T}(ABCD)$ does not, in general, imply $\mathcal{T}(DABC)$. We visualise $\mathcal{T}(ABCD)$ in Figure 2 by placing an "observer" at B who is looking at the edge AC such that A is above C for them. Then D is on the right for this observer.

Placing an observer in the same way in a situation where $\mathcal{T}(DABC)$ (which is equivalent to $\mathcal{T}(CBAD)$), they now see D on their left.

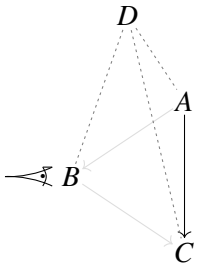


Figure 2: Observer looking at the edge AC from B sees D on their right.

Remark 10. We chose not to include the orientation of the triangle (plane), which amounts to the choice of S_3 over A_3 in the definition of a plane relation (Definition 8). This is because we assume that our molecules float freely in space (e.g. in a solution), so that there is no two-dimensional orientation.

The following example demonstrates that the plane and tetrahedron relations indeed capture planes and tetrahedrons in the Euclidean setting.

Example 11. Let us define the plane relation $\mathcal{P}$ on the 3-dimensional Euclidean space $\mathbb{R}^3$ by letting $\mathcal{P}(abc)$ if and only if $(b-a)\times(c-a)\neq 0$, where $\times$ denotes the vector product. We then have $\mathcal{P}(abc)$ precisely when c does not lie on the line determined by a and b, that is, when the three points uniquely determine a plane in $\mathbb{R}^3$.

With respect to the above plane relation, let us define the tetrahedron relation $\mathcal{T}$ by letting $\mathcal{T}(abcd)$ if and only if $\overline{(b-a)(c-a)(d-a)} > 0$, where the bar denotes the scalar triple product. We then have $\mathcal{T}(abcd)$ precisely when the points a, b, c and d are vertices of a non-degenerate (a non-zero volume) tetrahedron in such a way that d lies on that side of the plane determined by a, b and c to which the vector $(b-a)\times(c-a)$ points (see Figure 2).

Definition 12 (Oriented molecule partition, ion, and ionic compound)**.**

- An *oriented molecule partition* is a tuple $(V,\tau,m,\mathcal{P},\mathcal{T})$ where (V,τ,m) is a molecule partition, $\mathcal{P}$ is a plane relation on V and $\mathcal{T}$ is a tetrahedron relation on V with respect to $\mathcal{P}$.
- An *oriented ion* is an ion which is oriented as a molecule partition.
- An *oriented ionic compound* is a tuple $(I,i,\mathcal{P},\mathcal{T})$ where (I,i) is an ionic compound, $\mathcal{P}$ is a plane relation on V_I and $\mathcal{T}$ is a tetrahedron relation on V_I with respect to $\mathcal{P}$.

From now on, we adopt the convention that every molecule partition, ion and ionic compound is oriented: if the plane and tetrahedron relations are not specified, we take them to be empty (which means there are no constraints on the configuration).

Definition 13 (Preservation and reflection of orientation)**.** Let $(M,\mathcal{P}_M,\mathcal{T}_M)$ and $(N,\mathcal{P}_N,\mathcal{T}_N)$ be oriented molecule partitions, and let $f : M \to N$ be an isomorphism in the sense of Definition 2. We say that f *preserves orientation* if for all vertices A, B, C and D of M we have:

- $\mathcal{P}_M(ABC)$ if and only if $\mathcal{P}_N(fA,fB,fC)$,
- $\mathcal{T}_M(ABCD)$ if and only if $\mathcal{T}_N(fA,fB,fC,fD)$.

Similarly, we say that f *reflects orientation* if for all vertices A, B, C and D of M we have:

- $\mathcal{P}_M(ABC)$ if and only if $\mathcal{P}_N(fA,fB,fC)$,
- $\mathcal{T}_M(ABCD)$ if and only if $\mathcal{T}_N(fD,fA,fB,fC)$.

Definition 14 (Chirality)**.** We say that two molecule partitions are *chiral* if there is an orientation reflecting isomorphism, but no orientation preserving isomorphism between them.

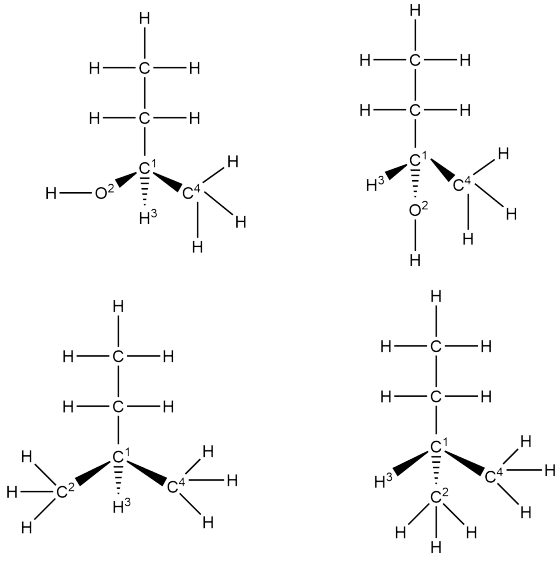
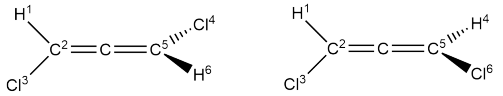


Figure 3: Top: 2-butanol. Bottom: isopentane.

Example 15. Consider 2-butanol, whose molecular structure we draw in two different ways in the top row of Figure 3. Here we adopt the usual chemical convention for drawing spatial structure: a dashed wedge indicates that the bond points "into the page", and a solid wedge indicates that the bond points "out of the page". In this case, we choose to include the names of the vertices for some labels as superscripts. The spatial structure is formalised by defining the tetrahedron relation for the graph on the left-hand side as the closure under the action of A_4 of $\mathcal{T}(1234)$, and for the one on the right-hand side as (the closure of) $\mathcal{T}(4123)$. In both cases, the plane relation is dictated by the tetrahedron relation, so that any three-element subset of $\{1,2,3,4\}$ is in the plane relation. Now the identity map (on labelled graphs) reflects orientation. It is furthermore not hard to see that every isomorphism restricts to the identity on the vertices labelled with superscripts, so that there is no orientation preserving isomorphism. Thus the two molecules are chiral according to Definition 14.

Example 16. Let us modify the above example by replacing the OH-group of 2-butanol with a CH_3-group, so that we obtain two different ways to draw isopentane, as shown in the bottom row of Figure 3. We keep the tetrahedron relations as before, so that $\mathcal{T}(1234)$ on the left-hand side and $\mathcal{T}(4123)$ on the right-hand side. Now the identity map still reflects orientation, however, in this case there is also an isomorphism which swaps vertices 2 and 4, which preserves orientation. Thus the two molecules are not chiral according to Definition 14.

Example 17. Example 15 with 2-butanol demonstrated how to capture central chirality using Definition 14. In this example, we consider 1,3-dichloroallene as an example of axial chirality. We draw two versions, as before. The tetrahedron relation is generated by $\mathcal{T}(1234)$ and $\mathcal{T}(6123)$ for both molecules (note, however, that the vertices 4 and 6 have different labels). Now the isomorphism which swaps vertices 4 and 6 and is identity on all other vertices is orientation reflecting, but not orientation preserving. The only other isomorphism is $1 \mapsto 4$, $2 \mapsto 5$, $3 \mapsto 6$, $4 \mapsto 3$, $5 \mapsto 2$, $6 \mapsto 1$, which does not preserve orientation. Thus the two molecules are indeed chiral.



5 Layers of morphisms

The main construction of this paper is the layered prop whose layers all share the same set of objects: namely, they are generated by oriented molecule partitions and ionic compounds. The morphisms of a layer are determined by three factors: (1) which retrosynthetic disconnection rules are allowed, (2) which environmental molecules are present (these can act as solvents, reagents or catalysts), and (3) which reactions are allowed. The aim of this section is to give the details of this construction.

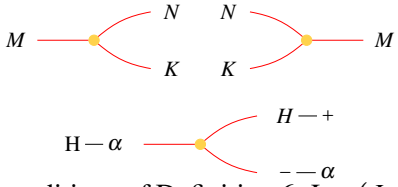
We denote by $\mathcal{P}art\mathcal{M}ol^+$ the free commutative monoid on the set of oriented molecule partitions and ionic compounds. The monoid operation is denoted by $+$, so that a typical element in $\mathcal{P}art\mathcal{M}ol^+$ is an unordered list $M_0 + \cdots + M_n$ with possibly repeated elements.

We begin by defining the disconnection rules, which consist of *electron detachment* (Def. 18), *ionic bonding* (Def. 20), *partitioning* (Def. 22), *cutting* (Def. 24) and *saturation* (Def. 26). These rules are chosen since they are used in the current retrosynthesis practice (e.g. [38, 10]). However, we also conjecture that the rules are complete in the sense that every reaction (Def. 28) can be decomposed into a sequence of disconnection rules.

Definition 18 (Electron detachment)**.** Let $M = (V, \tau, m, \mathcal{P}, \mathcal{T})$ be an oriented molecule partition such that at least one of its vertices is labelled by the free variable α. Let us fix some vertex v_α such that $\tau(v_\alpha) = \alpha$, and let X be the unique vertex such that $m(X, v_\alpha) = 1$. Further, suppose that $\tau(X) \notin \{+, -\}$. We then define

two molecule partitions $M^+ = (V, \tau^+, m, \mathcal{P}, \mathcal{T})$ and $e^\alpha = (\{-, \alpha\}, \tau_e, m_e)$. For M^+, the set of vertices, the edge function and the plane and tetrahedron relations are the same as for M, while we let $\tau^+(v_\alpha) = +$ and let τ^+ agree with τ otherwise. For e^α, define $\tau_e(-) = -$, $\tau_e(\alpha) = \alpha$ and $m_e(-, \alpha) = 1$. Define the *electron detachment relation* $E \subseteq \mathcal{P}art\mathcal{M}ol \times (\mathcal{P}art\mathcal{M}ol \times \mathcal{P}art\mathcal{M}ol)$ by letting $ME(M^+, e^\alpha)$ for all M and v_α satisfying the above conditions.

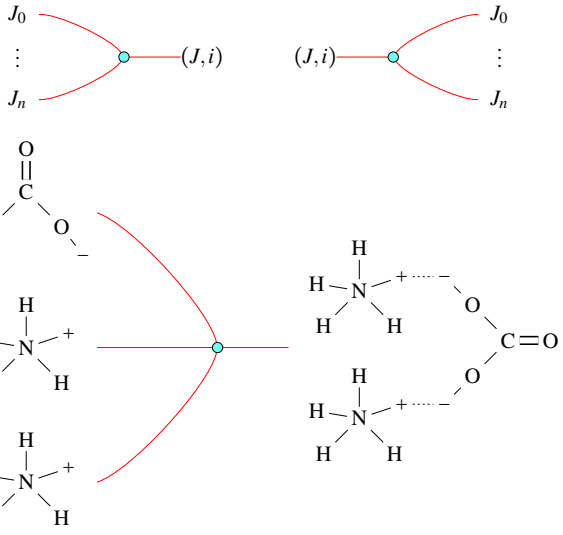
Using the electron detachment relation, for all molecule partitions M, N and K such that $ME(N, K)$, we define the morphisms as drawn on the right.



Example 19. The effect of the electron detachment relation is to detach (or attach) an electron from a molecule (partition), thus leaving it with a positive charge. We show this below right.

Definition 20 (Ionic bonding)**.** Let $J_0, \ldots, J_n$ be a collection of oriented ions such that there is an ionic bond relation i satisfying the conditions of Definition 6. Let (J, i) denote the resulting ionic compound, whose plane and tetrahedron relations are the union of those on individual ions. We define the relation $I \subseteq \mathcal{P}art\mathcal{M}ol^+ \times \mathcal{P}art\mathcal{M}ol^+$ by stipulating that $J_0 + \cdots + J_n I(J, i)$ for all $J_0, \ldots, J_n$ and (J, i) as above.

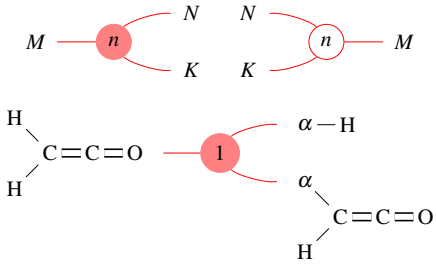
For all ions $J_0, \ldots, J_n$ and ion compounds (J, i) such that $J_0 + \cdots + J_n I(J, i)$, we define the morphisms on the right.



Example 21. An example of ionic bonding is given by forming ammonium carbonate from two ammonium cations and one carbonate anion, as shown below right.

Definition 22 (n-fold bond partitioning relation)**.** Let $M = (V, \tau, m, \mathcal{P}, \mathcal{T})$ be an oriented molecule partition and let $u, v \in V$. Denote by $m' : V \times V \to \mathbb{N}$ the function such that $m'(u, v) = 0$ and $m' = m$ otherwise. Suppose that (1) $m(u, v) = n \geq 1$, and (2) the graph (V, m') is not connected. In such case, we denote by $V(u)$ and $V(v)$ the connected components of u and v, respectively, in (V, m'). Let $M_u^\alpha = (V(u) \sqcup \coprod_{i=1}^n \{\alpha_i\}, \tau_\alpha, m_u, \mathcal{P}_v, \mathcal{T}_v)$ be the molecule partition where $\tau_\alpha(\alpha_i) = \alpha$ for each $i = 1, \ldots, n$ and $\tau_\alpha = \tau$ otherwise, and $m_u(u, \alpha_i) = 1$ for each $i = 1, \ldots, n$ and $m_u = m$ otherwise, and for each $i = 1, \ldots, n$ and all $B, C, D \in V(u)$ we have $\mathcal{P}_v(\alpha_i BC)$ if and only if $\mathcal{P}(vBC)$ and $\mathcal{T}_v(\alpha_i BCD)$ if and only if $\mathcal{T}(vBCD)$. The molecule partition $M_v^\alpha = (V(v) \sqcup \coprod_{i=1}^n \{\alpha_i\}, \tau_\alpha, m_v, \mathcal{P}_u, \mathcal{T}_u)$ is defined similarly. Define the *n-fold bond partitioning relation* $P_n \subseteq \mathcal{P}art\mathcal{M}ol \times (\mathcal{P}art\mathcal{M}ol \times \mathcal{P}art\mathcal{M}ol)$ by stipulating that $MP_n(M_u^\alpha, M_v^\alpha)$ for all M, v and u that satisfy the above conditions.

For all molecule partitions M, N and K such that $MP_n(N, K)$, we introduce the n-fold partitioning morphisms, as drawn on the right.

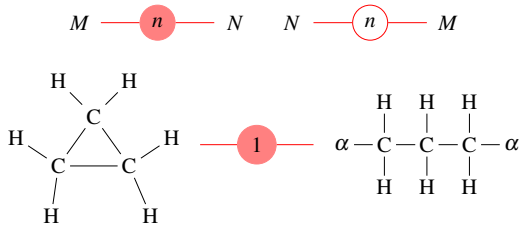


Example 23. The effect of an n-fold partition is to split a molecule partition into two parts along an n-fold bond, as below right.

Definition 24 (n-fold bond cutting relation)**.** Let $M = (V, \tau, m, \mathcal{P}, \mathcal{T})$ be a molecule partition and let $u, v \in V$. Denote by $m' : V \times V \to \mathbb{N}$ the function such that $m'(u, v) = 0$

and $m' = m$ otherwise. Suppose that (1) $m(u,v) = n \geq 1$, and (2) the graph (V,m') is connected. Define $M^{\alpha}_{u,v} = (V \sqcup \coprod_{i=1}^{n} \{\alpha_i, \beta_i\}, \tau_\alpha, m_{u,v}, \mathcal{P}, \mathcal{T})$ to be the molecule partition where $\tau_\alpha(\alpha_i) = \tau_\alpha(\beta_i) = \alpha$ for each $i = 1, \ldots, n$ and $\tau_\alpha = \tau$ otherwise; further, $m_{u,v}(u, \alpha_i) = m_{u,v}(v, \beta_i) = 1$ for each $i, j = 1, \ldots, n$, and $m_u = m'$ otherwise. The plane and tetrahedron relations are unchanged. Define the *n-fold bond cutting relation* $C_n \subseteq \mathcal{P}art\mathcal{M}ol \times \mathcal{P}art\mathcal{M}ol$ by stipulating that $MC_nM^{\alpha}_{u,v}$ for all M, v and u as above.

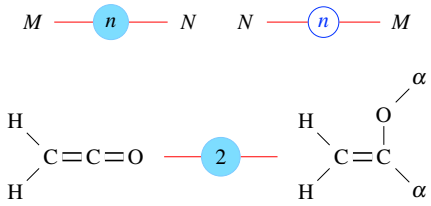
For all molecule partitions M and N such that MC_nN, we introduce the n-fold cutting morphisms, as drawn on the right.



Example 25. The effect of an n-fold bond cutting is to make a cut in a cyclic molecule along an n-fold bond.

Definition 26 (n-fold saturation relation)**.** Given a natural number $n \geq 2$, define the *n-fold saturation relation* $S_n \subseteq \mathcal{P}art\mathcal{M}ol \times \mathcal{P}art\mathcal{M}ol$ as follows. Let $M = (V, \tau, m, \mathcal{P}, \mathcal{T})$ be an oriented molecule partition, and suppose there are $u, v \in V$ such that $m(u,v) = n$. In such case let us define the molecule partition $M^{\alpha}_{u,v} = (V \sqcup \{\alpha, \beta\}, \tau_\alpha, m_{u,v}, \mathcal{P}, \mathcal{T})$ by letting $\tau_\alpha(\alpha) = \tau_\alpha(\beta) = \alpha$ and $\tau_\alpha = \tau$ otherwise; and by further letting $m_{u,v}(u,v) = n-1$, $m_{u,v}(u,\alpha) = m_{u,v}(v,\beta) = 1$ and $m_{u,v}(w,z) = m(w,z)$ for all other cases. The plane and tetrahedron relations are unchanged. We define S_n by stipulating that $MS_nM^{\alpha}_{u,v}$ for all M, u and v that satisfy the above conditions.

For all molecule partitions M and N such that MS_nN, introduce the n-fold saturation morphisms, as drawn on the right.



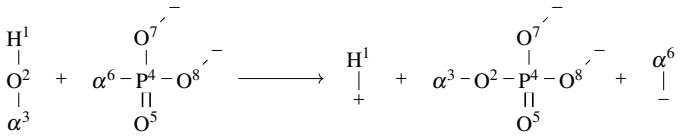
Example 27. The effect of an n-fold saturation is to "open" one edge in an n-fold bond, thus making it an $n-1$-fold bond (hence the requirement that n is at least two). We show this below right.

Given $A \in \mathcal{P}art\mathcal{M}ol^{+}$, let us denote by (V_A, τ_A, m_A) the labelled graph which is the disjoint union of the individual graphs in A, and by $N_A \coloneqq \tau_A^{-1}(\mathrm{At} \sqcup \{\alpha\})$ the subset of those vertices in V_A which are labelled by either an atom or the free variable α. We refer to N_A as the *neutral* vertices of A, as they are not labelled with a charge.

Definition 28 (Reaction)**.** A *reaction* consists of an ordered pair of elements (A,B) of $\mathcal{P}art\mathcal{M}ol^{+}$ such that A and B have the same net charge in the sense that $\left|\tau_A^{-1}(+)\right| - \left|\tau_A^{-1}(-)\right| = \left|\tau_B^{-1}(+)\right| - \left|\tau_B^{-1}(-)\right|$, together with a labelled isomorphism $f : N_A \to N_B$, that is, a bijection such that $\tau_B f = \tau_A$.

We write $A \to B$ for a reaction, and label the neutral vertices in A and B with superscripts to indicate the bijection.

Example 29. The rule shown on the right appears in the equation describing glucose phosphorylation. It is a reaction in the sense of Definition 28.



Let us denote by $\mathcal{D} \coloneqq \{E, I, P_n, C_n, S_{n+1} : n \in \mathbb{Z}_+\}$ the set of *disconnection rules* containing a symbol for each relation defined in this section, by $\mathcal{R}$ the set of all reactions and by $\mathcal{M}$ the set of all molecules.

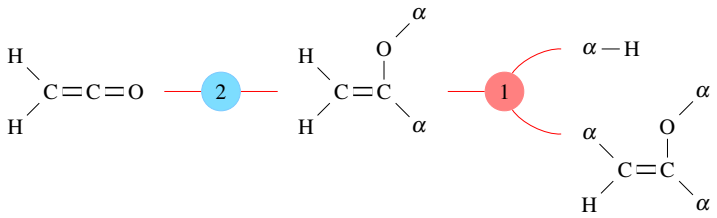
Definition 30 (Solvent-free category)**.** Given finite subsets $D \subseteq \mathcal{D}$ and $R \subseteq \mathcal{R}$, we define the *solvent-free category* generated by (D,R) as the monoidal category whose objects are $\mathcal{P}art\mathcal{M}ol^{+}$ and whose morphisms are generated by the relations in D and R in the sense that for every $d \in D$ and $A, B \in \mathcal{P}art\mathcal{M}ol^{+}$ with AdB, we add $d : A \to B$ as a generating morphism, and likewise add every reaction $r : A \to B$ in R as a generating morphism. Moreover, we require all the morphisms generated from D to be invertible.

Definition 31 (Solvent category)**.** Given finite subsets $D \subseteq \mathcal{D}$, $R \subseteq \mathcal{R}$ and $M \subseteq \mathcal{M}$, let us enumerate the molecules in M as $M_1, \ldots, M_k$. We define the *solvent category* generated by (D, R, M) as the monoidal category whose objects are $\mathcal{P}art\mathcal{M}ol^+$ and

- a morphism $A \to B$ is given by a morphism $M_1^{n_1} + \cdots + M_k^{n_k} + A \to B$ in the solvent-free category generated by (D, R),
- if two morphisms $A \to B$ and $B \to C$ are given by $f : M_1^{n_1} + \cdots + M_k^{n_k} + A \to B$ and $g : M_1^{m_1} + \cdots + M_k^{m_k} + B \to C$, then the composite $A \to C$ is given by $g \circ (\mathrm{id}_{M_1^{m_1} + \cdots + M_k^{m_k}} \otimes f) : M_1^{n_1+m_1} + \cdots + M_k^{n_k+m_k} + A \to C$.

Construction of a solvent category extends to a functor $\mathcal{P}_f(\mathcal{D}) \times \mathcal{P}_f(\mathcal{R}) \times \mathcal{P}_f(\mathcal{M}) \to \mathbf{StrMon}$, which determines a layered prop. In this layered prop, the context is given by a finite collection of disconnection rules, a finite collection of reactions and a finite collection of molecules. We think of the latter as containing the solvent, the reagents and the catalysts. The morphisms inside a context are the allowed disconnection rules and reactions, with an unbounded supply of the specified molecules (solvent, reagent and catalyst). In the next section, we will formulate retrosynthetic analysis using the layered prop of solvent categories.

Example 32. We can use a morphism in the layer generated by the disconnection rules $\{P_1, S_2\}$ to show how to obtain the molecule partition of Example 5 from the molecule therein.



6 Retrosynthesis in a layered prop

We are now ready to formulate step-by-step retrosynthetic analysis, using the layered prop defined in the previous section. Below, T refers to a fixed target molecule, and a *layer* refers to the choice of the context (disconnection rules, reactions and environment molecules).

1. Choose a layer (based on the structure of T),
2. Apply an available disconnection rule to T,
3. Replace the free variables created in Step 2 with molecules,
4. Do one of the following:
   (a) Search through reactions in the layer to find a reaction whose domain are the compounds in Step 3 and which would restore the bond disconnected in Step 2,
   (b) Apply a previously trained algorithm to the compounds in Step 3, and check whether T is amongst the outputs,
5. If the previous step terminates successfully, accept the molecules of Step 3 as new targets, otherwise return to Step 1,
6. Repeat Steps 1-5 until existing compounds are reached.

Note how our framework is able to incorporate both template-based and template-free retrosynthesis, corresponding to the choices between (a) and (b) in Step 4. Because Step 1 chooses a layer for each disconnection, the output retrosynthetic sequence will come with a specified reaction context for each reaction. Currently existing tools do not provide this information (mostly for complexity reasons), and hence in our framework correspond to always choosing a solvent-free layer in Step 1.

Steps 1-3 all require making some choices. Two approaches to reduce the number of choices have been proposed in the automated retrosynthesis literature: to use molecular similarity [13], or machine learning [25]. Chemical similarity can be used to determine which disconnection rules, reactions and environment molecules are actually tried: in Steps 2 and 3, disconnection rules that appear in syntheses

of molecules similar to T can be prioritised, while in Step 1 constraints on the reaction context can be imposed based on the structure of T in order to limit the choice of layers. Alternatively, separate algorithms can be used to learn what is most fruitful to try in these steps.

7 Discussion and future work

To build a retrosynthesis algorithm, we need to encode known reactions (for example, to train a machine learning algorithm). This formalism offers an intuitive, visual way to do so by representing reactions as disconnection rules. In practice, and as a lower level description, the disconnection rules and the reactions presented here could be encoded in some graph rewriting language, such as Kappa [22, 17, 23, 4], which is used to model systems of interacting agents, or MØD [30, 2, 3, 4], which represents molecules as labelled graphs (akin to this work), and generating rules for chemical transformations as spans of graphs. In order to formally represent reactions as disconnection rules, we need to rewrite string diagrams (in addition to graphs), the theory for which has been developed in a recent series of articles [6, 7, 8].

The main conceptual contribution of formulating retrosynthesis in layered props is the explicit inclusion of the reaction context into the model. While in the current article we showed how to account for available disconnection rules, reactions and environmental molecules, the general formalism of layered props immediately suggests how to account for other environmental factors (e.g. temperature and pressure). Namely, these should be represented as posets which control the morphisms that are available between the chemical compounds. One idea for accounting for the available energy is via the disconnection rules: the higher the number of bonds that we are able to break in one step, the more energy is required to be present in the environment. Apart from modelling retrosynthesis, another potential use of the reaction contexts is to capture context-dependent chemical similarity. While molecular similarity is a major research topic in computational chemistry [5], the current approaches are based on comparing the molecular structure (connectivity, number of rings etc.) of two compounds, and is therefore bound to ignore the reaction context.

Other advantages of our framework are intuitive representation of the protection-deprotection steps, and hard-wiring of chirality into the formalism. Current retrosynthesis algorithms use molecular complexity measures to guide the algorithm from the complex desired product to the simpler starting materials. Protection-deprotection steps require that a molecule is temporarily made more complex, the reaction is done, and then the resulting molecule made less complex again. Layers can be used to achieve this. Suppose we are in a situation where compounds A and B can react in two different ways, one desirable and the other undesirable. We then move to a layer which only contains the undesirable reaction, and add a functional group to A in a way that there are no non-trivial morphisms with A as the domain. If we additionally find a reaction that achieves said addition of the functional group, we have obtained a protection step.

The clear next step for this work is to implement the morphisms in the layered prop of Section 5 on a computer. As the morphisms are represented by string diagrams, one approach is to use proof formalisation software specific to string diagrams and their equational reasoning, such as [31]. Alternatively, these morphisms could be coded into a programming language like python or Julia. The latter is especially promising, as there is a community writing category-theoretic modules for it [1]. The final aim is to build a machine learning algorithm for retrosynthesis. There are many possible disconnections for every molecule, hence a machine learning algorithm can learn the probabilities of success for each transformation from reaction databases, greatly reducing the numbers of possible routes that need to be searched over.

References

[1] *AlgebraicJulia*. Website. Available at `https://www.algebraicjulia.org/`.

[2] Jakob L. Andersen, Christoph Flamm, Daniel Merkle & Peter F. Stadler (2017): *An intermediate level of abstraction for computational systems chemistry*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 375(2109), p. 20160354, doi:10.1098/rsta.2016.0354. Available at `https://doi.org/10.1098%2Frsta.2016.0354`.

[3] Jakob L. Andersen, Christoph Flamm, Daniel Merkle & Peter F. Stadler (2019): *Chemical Transformation Motifs – Modelling Pathways as Integer Hyperflows*. IEEE/ACM transactions on computational biology and bioinformatics 16(2), pp. 510–523.

[4] Nicolas Behr, Jean Krivine, Jakob L. Andersen & Daniel Merkle (2021): *Rewriting Theory for the Life Sciences: A Unifying Theory of CTMC Semantics*. Theor. Comput. Sci. 884(C), p. 68–115, doi:10.1016/j.tcs.2021.07.026. Available at `https://doi.org/10.1016/j.tcs.2021.07.026`.

[5] Andreas Bender & Robert C. Glen (2004): *Molecular similarity: a key technique in molecular informatics*. Org. Biomol. Chem. 2, pp. 3204–3218, doi:10.1039/B409813G. Available at `http://dx.doi.org/10.1039/B409813G`.

[6] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński & Fabio Zanasi (2022): *String Diagram Rewrite Theory I: Rewriting with Frobenius Structure*. J. ACM 69(2), doi:10.1145/3502719. Available at `https://doi.org/10.1145/3502719`.

[7] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński & Fabio Zanasi (2022): *String diagram rewrite theory II: Rewriting with symmetric monoidal structure*. Mathematical Structures in Computer Science 32(4), p. 511–541, doi:10.1017/S0960129522000317.

[8] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński & Fabio Zanasi (2022): *String diagram rewrite theory III: Confluence with and without Frobenius*. Mathematical Structures in Computer Science 32(7), p. 829–869, doi:10.1017/S0960129522000123.

[9] Shuan Chen & Yousung Jung (2021): *Deep Retrosynthetic Reaction Prediction using Local Reactivity and Global Attention*. JACS Au 1(10), pp. 1612–1620.

[10] Jonathan Clayden, Nick Greeves & Stuart Warren (2012): *Organic chemistry*, second edition. edition. Oxford University Press, Oxford.

[11] Connor W. Coley, Regina Barzilay, Tommi S. Jaakkola, William H. Green & Klavs F. Jensen (2017): *Prediction of Organic Reaction Outcomes Using Machine Learning*. ACS central science 3(5), pp. 434–443.

[12] Connor W. Coley, William H. Green & Klavs F. Jensen (2018): *Machine learning in computer-aided synthesis planning*. Accounts of chemical research 51(5), pp. 1281–1289.

[13] Connor W. Coley, Luke Rogers, William H. Green & Klavs F. Jensen (2017): *Computer-Assisted Retrosynthesis Based on Molecular Similarity*. ACS central science 3(12), pp. 1237–1245.

[14] A. Gilbert Cook & Paul M. Feltman (2007): *Determination of Solvent Effects on Keto—Enol Equilibria of 1, 3-Dicarbonyl Compounds Using NMR*. Journal of chemical education 84(11), p. 1827.

[15] E. J. Corey (1988): *Robert Robinson lecture. Retrosynthetic thinking – essentials and examples*. Chemical society reviews 17, pp. 111–133.

[16] E. J. Corey & Xue min Cheng (1989): *The logic of chemical synthesis*. John Wiley, New York.

[17] Vincent Danos & Cosimo Laneve (2004): *Formal molecular biology*. Theoretical computer science 325(1), pp. 69–110.

[18] Jingxin Dong, Mingyi Zhao, Yuansheng Liu, Yansen Su & Xiangxiang Zeng (2022): *Deep learning in retrosynthesis planning: datasets, models and tools*. Briefings in Bioinformatics 23(1), p. bbab391.

[19] Marco Filice, Jose M. Guisan & Jose M. Palomo (2010): *Recent trends in regioselective protection and deprotection of monosaccharides*. Current Organic Chemistry 14(6), pp. 516–532.

[20] Michael E. Fortunato, Connor W. Coley, Brian C. Barnes & Klavs F. Jensen (2020): *Data augmentation and pretraining for template-based retrosynthetic prediction in computer-aided synthesis planning*. Journal of chemical information and modeling 60(7), pp. 3398–3407.

[21] Hanyu Gao, Thomas J. Struble, Connor W. Coley, Yuran Wang, William H. Green & Klavs F. Jensen (2018): *Using Machine Learning To Predict Suitable Conditions for Organic Reactions*. ACS central science 4(11), pp. 1465–1476.

[22] *Kappa Language*. Available at `https://kappalanguage.org/`.

[23] Jean Krivine (2017): *Systems Biology*. ACM SIGLOG News 4(3), p. 43–61, doi:10.1145/3129173.3129182. Available at `https://doi.org/10.1145/3129173.3129182`.

[24] James Law, Zsolt Zsoldos, Aniko Simon, Darryl Reid, Yang Liu, Sing Yoong Khew, A. Peter Johnson, Sarah Major, Robert A. Wade & Howard Y. Ando (2009): *Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation*. Journal of Chemical Information and Modeling 49(3), pp. 593–602.

[25] Kangjie Lin, Youjun Xu, Jianfeng Pei & Luhua Lai (2020): *Automatic retrosynthetic route planning using template-free models*. Chemical science (Cambridge) 11(12), pp. 3355–3364.

[26] Leo Lobski & Fabio Zanasi (2022): *String Diagrams for Layered Explanations*. To appear in the proceedings of ACT 2022. Available at `https://arxiv.org/abs/2207.03929v1`.

[27] G. Marcou, J. Aires de Sousa, D. A. R. S. Latino, A. de Luca, D. Horvath, V. Rietsch & A. Varnek (2015): *Expert System for Predicting Reaction Conditions: The Michael Reaction Case*. Journal of Chemical Information and Modeling 55(2), pp. 239–250.

[28] Michael R. Maser, Alexander Y. Cui, Serim Ryou, Travis J. DeLano & Yisong Yue (2021): *Multilabel Classification Models for the Prediction of Cross-Coupling Reaction Conditions*. Journal of Chemical Information and Modeling 61(1), pp. 156–166.

[29] Arkadiusz Matwijczuk, Dariusz Karcz, Radosław Walkowiak, Justyna Furso, Bożena Gładyszewska, Sławomir Wybraniec, Andrzej Niewiadomy, Grzegorz P Karwasz & Mariusz Gagoś (2017): *Effect of solvent polarizability on the keto/enol equilibrium of selected bioactive molecules from the 1, 3, 4-thiadiazole group with a 2, 4-hydroxyphenyl function*. The Journal of Physical Chemistry A 121(7), pp. 1402–1411.

[30] *MØD*. Available at `https://cheminf.imada.sdu.dk/mod/`.

[31] P. Sobocinski, P. Wilson & F. Zanasi (2019): *CARTOGRAPHER: a Tool for String Diagrammatic Reasoning*. In M Roggenbach & A Sokolova, editors: *8th Conference on Algebra and Coalgebra in Computer Science (CALCO)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 20:1–20:7. Available at `https://cartographer.id/cartographer-calco-2019.pdf`.

[32] Vignesh Ram Somnath, Charlotte Bunne, Connor W. Coley, Andreas Krause & Regina Barzilay (2020): *Learning graph models for template-free retrosynthesis*. arXiv preprint arXiv:2006.07038.

[33] Felix Strieth-Kalthoff, Frederik Sandfort, Marwin H. S. Segler & Frank Glorius (2020): *Machine learning the ropes: principles, applications and directions in synthetic chemistry*. Chemical Society Reviews 49(17), pp. 6154–6168.

[34] Yijia Sun & Nikolaos V. Sahinidis (2022): *Computer-aided retrosynthetic design: fundamentals, tools, and outlook*. Current Opinion in Chemical Engineering 35, p. 100721.

[35] Umit V. Ucak, Islambek Ashyrmamatov, Junsu Ko & Juyong Lee (2022): *Retrosynthetic reaction pathway prediction through neural machine translation of atomic environments*. Nature communications 13(1), pp. 1186–1186.

[36] Eric Walker, Joshua Kammeraad, Jonathan Goetz, Michael T. Robo, Ambuj Tewari & Paul M. Zimmerman (2019): *Learning To Predict Reaction Conditions: Relationships between Solvent, Molecular Structure, and Catalyst*. Journal of Chemical Information and Modeling 59(9), pp. 3645–3654.

[37] Stuart Warren (1991): *Designing organic syntheses: a programmed introduction to the synthon approach*. John Wiley & Sons.

[38] Stuart Warren & Paul Wyatt (2008): *Organic synthesis : the disconnection approach*, 2nd ed. edition. Wiley, Hoboken, N.J.

[39] Chaochao Yan, Peilin Zhao, Chan Lu, Yang Yu & Junzhou Huang (2022): *RetroComposer: Composing Templates for Template-Based Retrosynthesis Prediction*. Biomolecules 12(9), p. 1325.