Double Pushout Rewriting in Chemistry

Handout

Georg Fischer, Hannah Gschwentner

February 3, 2022

1 Introduction

This Handout is about Double Pushout Graph Rewriting (DPO) which is an important algebraic approach to graph transformation. We can formulate DPO on a purely categorical level. Pushout is the concept of gluing graphs defined by algebraic constructions in the category of graphs, which allow deletion and preservation of vertices and edges. In short, rewriting has two steps: deletion and addition.
The idea of all graph transformation approaches is the production based modifications of graphs, where, in an informal way, a production $p : L \to R$ of graphs L and R, meaning RHS and LHS, defines a correspondence between elements of L and R.
If we apply the production to a given graph G, we have to find an occurrence of L in G called match m. So a match is a graph morphism $m : L \to G$, where G is a graph and m maps edges and vertices from L to G s.t the graphical structure is preserved. This determines which edges and vertices have to be preserved, deleted or created.

1.1 First example

Remember that a graph G is a pair (V, E), where V is the set of vertices and E the set of edges. Furthermore we have label functions $l_v : V \to L_V$ and $l_e : E \to L_E$, where L_V and L_E are fixed label alphabets.
In the next example we see a diagram with a graph, a production and a match. This is just an informal diagram, where non of the productions or objects belong to a category and it is not commutative as we would conclude from our lecture. Just to have an idea of graph rewriting and the concept of gluing graphs.

Example 1.1.1. Given a graph G_1, a production $p_1 : L_1 \to R_1$ and a graph homomorphism called match $m_1 : L_1 \rightarrow G_1$. The match m_1 maps edges and vertices from the graph L_1 to G_1 such that the graphical structure is preserved. Lets have a look at our diagram example:

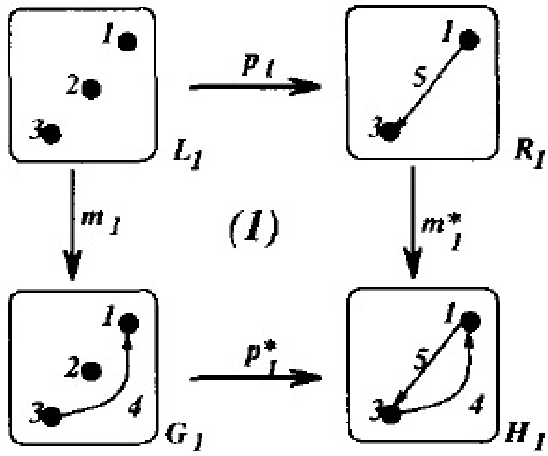


Figure 1: First example

In L_1 we have three vertices $\{1, 2, 3\}$ and the application of the production p_1 preserves the vertices $\{1, 3\}$ and connects them with an edge 5 in R_1. In this context, same numbers mean the same object. Then we have to find a match $m_1 : L_1 \rightarrow G_1$, which is an occurrence of L_1 in G_1.
As mentioned before rewriting has two steps: deletion and addition.

Deletion:

Now we apply p_1 to G_1 at m_1. Every element in G_1 that has no corresponding element in R_1 and matches an element in L_1 has to be deleted. In this case we delete vertex 2 of G_1.

Addition:

The corresponding element of R_1 in L_1 has to be added to G_1. In this example we have to add edge 5 in H_1.

2 Pushouts

In this section we want to introduce graph rewriting in a formal way. Form now on we are situated in the category **Graph**.
Let **Graph** be the category of labeled graphs as objects and arrows as graph morphisms. For the labeled graphs there are given two fixed alphabets L_V and L_E, one for the vertices the other for the edges. Therefore, a labeled graph over the alphabets (L_V, L_E) is a tuple $G = (V, E, s_G, t_G)$, where $s_G : E \to V$ is the source function and $t_G : E \to V$ the target function.
To get a double pushout, you will see, we have to define a production as a span, because a span is a pair of graph morphisms with common source. This structure we will need for the double pushout in order to achieve the two commuting squares.

Definition 2.0.1 (span)**.** A span in a category $\mathcal{C}$ from an object X to an object Y and S in $\mathcal{C}$ is defined as a pair of graph morphism as we see in the diagram below:

$$X \overset{f}{\leftarrow} S \overset{g}{\rightarrow} Y$$

Then we can define the production in the DPO approach in particular.

Definition 2.0.2 (production)**.** A production p in a category $\mathcal{C}$ is defined as a span:

$$L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R$$

Because we are in the category of graphs, we define also a graph production.

Definition 2.0.3 (graph production)**.** A production p and a pair of injective graph morphisms $l : K \to L$ and $r : K \to R$ is called a graph production $p : (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$.

In this context, where we have given a production $p : (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$, a graph morphism $m : L \to G$, where G is a graph, is called a **match**.

If we obtain a new object by deleting some elements in L and replace it with R, we call this a derivation of a match along a production.

Definition 2.0.4 (direct derivation)**.**
Let be G a graph, p a production and $m : L \to G$ a match then a direct derivation from G to H exists

if and only if the diagram

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ \downarrow m & & \downarrow & & \downarrow m^* \\ G & \xleftarrow{l^*} & D & \xrightarrow{r^*} & H \end{array}$$

can be constructed and both squares are pushouts in the category **Graph**.

In DPO direct derivations are modeled by gluing conditions of graphs that are characterized as pushouts. In the definition above K is a common interface graph and this production consists of two gluing diagrams of graphs and total graph morphisms.
Looking at the diagram from the definition we see at the bottom, that we have the co-production $p^* : G \rightsquigarrow H$, which relates the derived graphs with the graph, we have given. D is called the context graph, which we gain by deleting all elements of G, which have a preimage in L, but does not have one in K (inverse gluing operation).

In order to avoid problematic situations a match m which must fulfill the gluing condition, which is separated in two parts: the dangling condition and the identification condition!

Dangling Condition

If the production p specifies the deletion of a vertex of G, then also all incident edges of G from the vertex have to be deleted.

Identification Condition

The element, which should be deleted by the application of the production, must have only one preimage in L

In graph rewriting it can happen that we find a match m of L, but the production p is not applicable, because no context exists. This happens if either the dangling condition or the identification condition is violated. In conclusion, if non of the conditions is satisfied by a match, the double pushout rewriting cannot occur.

- if the production removes a vertices, without removing all edges connected to this vertex In the first diagram, the edge c in D is not removed
- if two graph elements which are not preserved are identified by the match m

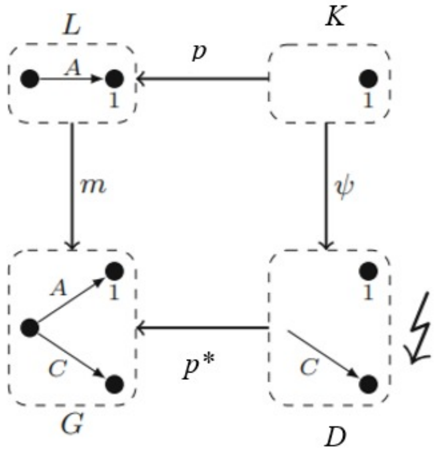


Figure 2: Dangling Condition

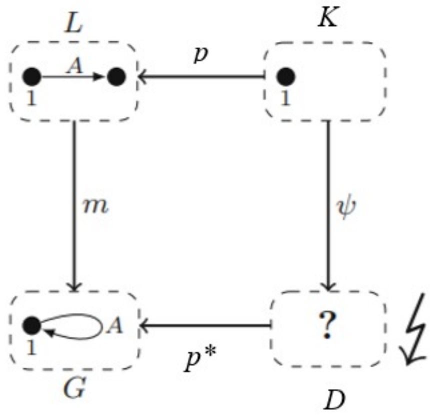
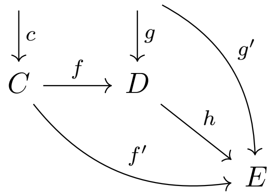


Figure 3: Identifiction Condition

Summarized, the gluing condition ensures that the application of p deletes only the elements which are specified by the production.

The next step is to define **pushouts**.

Definition 2.0.5 (pushout)**.** Let $\mathbf{C}$ be a category, $b : A \to B, c : A \to C$ two arrows of $\mathbf{C}$ like in the diagram, then the triple (D, g, f) is called a pushout over b and c if



- $g \circ b = f \circ c$ (Commutativity)

- for all objects D' and arrows $g' : B \to D'$ and $f' : C \to D'$, with $g' \circ b = f' \circ c$, there exists a unique arrow $h : D \to D'$ such that $h \circ g = g'$ and $h \circ f = f'$.

Remark 2.0.6. A pushout complement of the arrows b, g in the diagram above is a triple (C, c, f) such that (D, g, F) is a pushout of b, c. Then C is called a pushout complement object of b, g.

There always exists a pushout of to arrows in **Graphs**. For the edges and vertices it can be computed component wise as it can be easily seen by looking at the figure in the above definition.
In the double-pushout rewriting approach the construction of the DPO diagram cannot be ensured by the existence of a match for a production. The first pushout depends on the pushout complement for $(K \overset{l}{\to} L \overset{m}{\to} G)$ whether it exists or not. If the gluing condition is satisfied, the complement exists.

Proposition 2.0.7. The pushout complement object in **Graph** of two morphisms $b : A \to B$ and $c : B \to D$ exists if and only if the gluing condition is satisfied. Furthermore, the pushout complement is unique up to isomorphism if the morphism b is injective.

2.1 Construction of DPO

In this chapter we concentrate on the construction of the double pushout, where we need everything we introduced until now.
For the construction we refer to the diagram below to get a better impression. To apply a **production** p to graph G, we have to find an occurrence of L in G, which is our **match** m. To get a **double pushout** m has to satisfy the gluing conditions. The next step is to delete all items of L which not interface graph K from G. So we have to find a graph D and morphisms d and l^* s.t. square is a **pushout**. Then D is the pushout complement object of (l, m). Last, we embed R into D, which is expressed by the right pushout.

$$\begin{array}{ccccc} L & \xleftarrow{\;l\;} & K & \xrightarrow{\;r\;} & R \\ \downarrow{\scriptstyle m} & & \downarrow{\scriptstyle d} & & \downarrow{\scriptstyle m^*} \\ G & \xleftarrow[l^*]{} & D & \xrightarrow{\;r^*\;} & H \end{array}$$

Another short example of double pushout:

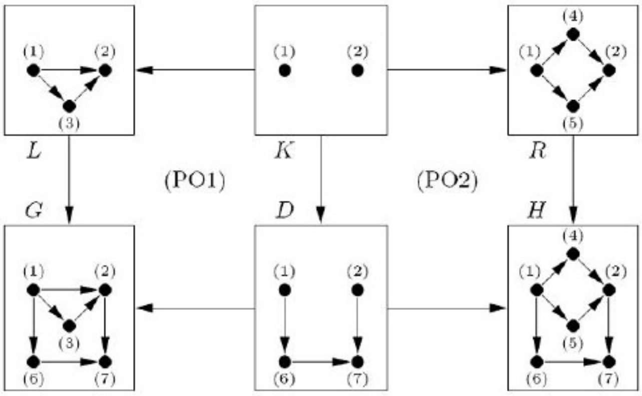


Figure 4: Double pushout

Above we see the production, where the vertices 1 and 2 are always preserved Then the match from L to G fullfills the gluing condition because the graph elements are preserved and no vertex is deleted so no incident edge had to be deleted. Both diagrams $PO1$ and $PO2$ commutes, we conclude they are clearly pushouts because both squares commute and if fulfills the other requirements as well. Furthermore, so we can see G and H are just the addition of L and D or R and D.

3 Chemical graphs, Chemical reactions

3.1 Chemical graph

In order to use this introduced way to rewrite graphs in chemistry we have to define first, what a chemical graph is and how a chemical reaction can be modeled as a graph transformation. Let us first take a look at the chemical graphs:

Definition 3.1.1. (Chemical graph) A **chemical graph** is a weighted graph (V, E, μ) which satisfies:

- (V, E) define an **undirected graph**
- vertices are labeled after **chemical elements**
- $\mu : E \to \mathbb{N}$ is a **weight function**
- the **valence** of a node is defined as the total weight of its incident edges.

3.2 Chemical reaction

With this defintion in mind we can introduce, what a **chemical reaction** is.
A chemical reaction is the change produced by two or more molecules acting upon each other so that substrate molecules are transformed into product molecules.
In the theory of pushouts, a chemical reaction can be interpreted as a production. But because it is a special production it has to satisfy two conditions:

- The **number** and **type** of atoms has to remain the same
- It consists of or is a combination out of the following four types of reactions:
  - **combination reaction**: Two substrates form one product (A+B $\to$ AB)
  - **decomposition reaction**: One substrate forms two products (AB $\to$ A+B)
  - **displacement reaction**: One of the substrates is displaced into another (A+BC $\to$ AC+B)
  - **exchange reaction**: One of the subtrates is exchanged by another (AB+CD $\to$ AD+CB)

3.3 Chemical reaction graph

We can finally formally introduce the graph description of a chemical reaction.

Definition 3.3.1. (Chemical reaction graph) A chemical reaction graph is a tuple (V, E, σ, π) with (V, E, σ) and (V, E, π) chemical graphs, called the substrate and the product chemical graphs respectively, satisfying the conditions:

- There is no edge $e \in E$ such that $\sigma(e) = \pi(e) = 0$. (else there would neither exist an edge e in the substrate or the product graph)
- For every vertex $v \in V$, if $e_1, ..., e_k$ are the edges incident to it , then

$$\sigma(e_1) + ... + \sigma(e_k) = \pi(e_1) + ... + \pi(e_k) \geq 1.$$

3.4 Diels-Alder Reaction

Now we want to describe the Diels-Alder reaction, one of the most important reaction in organic chemistry, by using the double pushout rewriting we established, in order to make the given definitions a bit more understandable.
At first we define a production for the Diels-Alder reaction.

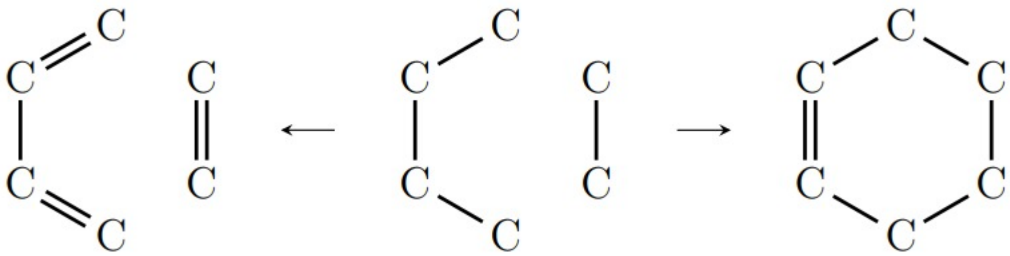


Figure 5: Production

This shows that the Diels-Alder reaction is just a combination reaction and is truly a chemical reaction in our defintion.
To see the Diels-Alder reaction as a double-pushout transformation, the 1, 3-cyclopentadiende (C_5H_6) molecule will match to 1, 3-butadiene (C_4H_6) and dihydro-2, 5-furandione ($C_4H_4O_3$) will match to the 1, 3-isobenzofurandione ($C_8H_8O_3$) (seen on the left handside in figure **??**).

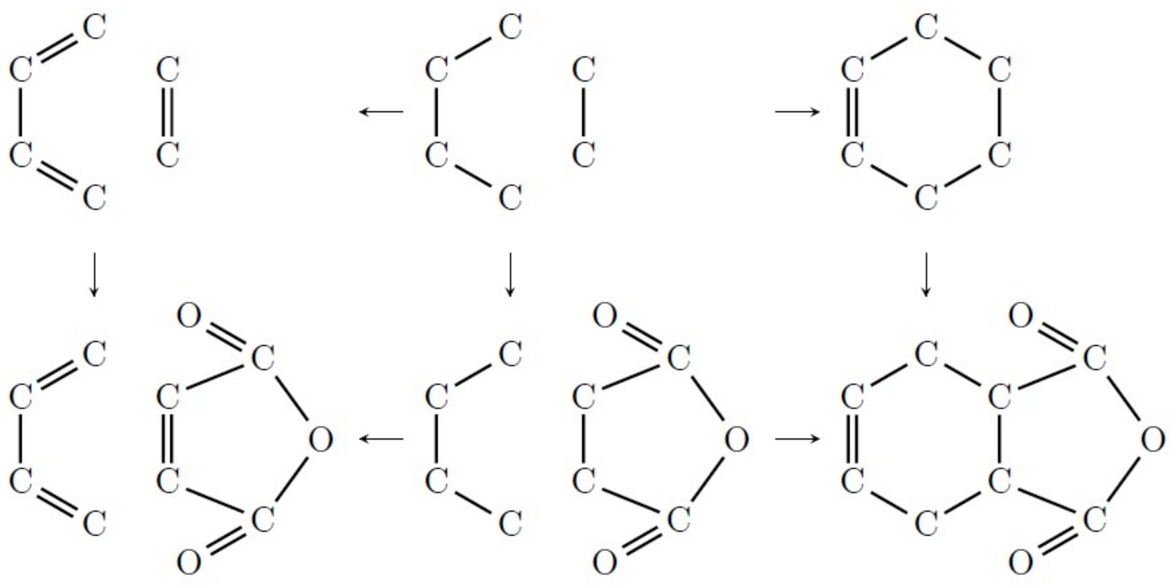


Figure 6: Double pushout for the Dies-Alder reaction

We end up with a double pushout graph for the Diels-Alder reaction.

4 Application and outlook

In order to use this rewriting technique in practise it was implemented in a program called Perl, under PerlMol. This is used to find structures in big molecules, in order to give an idea on how it could react.
The big problem, that remains to be solved, is, that the theory/program disregards:

- Ionic bonding (bonding between elements with sharply different electronegativity)
- Stereochemistry (three-dimensional orientation of elements in the structure)
- Chirality (distinguishability of structure from its mirror image)

and is therefore flawed in some aspects or in other words has potential to grow.