Chemical Graphs, Chemical Reaction Graphs, and Chemical Graph Transformation

ARTICLE *in* ELECTRONIC NOTES IN THEORETICAL COMPUTER SCIENCE · MARCH 2005

DOI: 10.1016/j.entcs.2004.12.033 · Source: DBLP

Francesc Rossello
University of the Balearic Islands

Gabriel Valiente
Polytechnic University of Catalonia

Chemical Graphs, Chemical Reaction Graphs, and Chemical Graph Transformation

Francesc Rosselló[1]

Department of Mathematics and Computer Science
Research Institute of Health Science (IUNICS)
University of the Balearic Islands
E-07122 Palma de Mallorca

Gabriel Valiente[2]

Department of Software
Technical University of Catalonia
E-08034 Barcelona

Abstract

Chemical reactions are described by edge relabeling graph transformation rules, in which a substrate chemical graph is transformed into a product chemical graph by breaking existing bonds and creating new bonds between atoms. These edge relabeling graph transformation rules are themselves chemical graphs, where the order of a bond before the chemical reaction is distinguished from the order of the bond after the chemical reaction. The approach is illustrated by an implementation on top of the PerlMol collection of Perl modules for computational chemistry.

Key words: chemical graph, chemical reaction, explicit chemical reaction, graph transformation, edge relabeling, PerlMol

1 Introduction

The aim of this note is to illustrate the approach initiated by the authors in [13] to model chemical reactions by edge relabeling chemical graph transformation rules, while describing an implementation on top of PerlMol [14].

[1] Partially supported by the Spanish DGES and the EU program FEDER, project BFM2003-00771 ALBIOM.

[2] Partially supported by the Spanish CICYT, project MAVERISH (TIC2001-2476-C03-01) and by the Ministry of Education, Science, Sports and Culture of Japan through Grant-in-Aid for Scientific Research B-15300003 for visiting JAIST (Japan Advanced Institute of Science and Technology).

PerlMol is a collection of Perl modules for computational chemistry which include an object-oriented representation of molecules, a suite of file I/O modules, and a powerful pattern-matching engine.

A molecule is represented in PerlMol as a `Chemistry::Mol` object that contains a group of `Chemistry::Atom` objects and a group of `Chemistry::Bond` objects. A molecule object can be constructed by adding atoms and bonds to an empty molecule.

```
use Chemistry::Mol;
my $mol = Chemistry::Mol->new;
$mol->new_atom(symbol=>"C");
$mol->new_atom(symbol=>"C");
...
$mol->new_bond(atoms=>[$mol->atoms(1),$mol->atoms(2)]);
...
```

A molecule object can also be constructed by reading in the contents of a file or parsing a string that describes the molecule. PerlMol supports the most common formats used in computational chemistry, including the MDL molfile format [3], the PDB format [19], and SMILES strings [16,17,18].

```
use Chemistry::Mol;
use Chemistry::File::MDL;
my $mol = Chemistry::Mol->read("file.mol");
```

PerlMol also provides for fast subgraph isomorphism. A molecule object can be matched in PerlMol to a substructure of a larger molecule. The pattern molecule is a `Chemistry::Pattern` object, which is a subclass of `Chemistry::Mol`, and it provides a `match` method to find all occurrences of the pattern in a given molecule.

```
use Chemistry::Mol;
use Chemistry::File::SMILES;
my $s = ’C=CC=C.C=C’;
my $patt = Chemistry::Mol->parse($s,format=>’smiles’);
$s = ’C1=CCC=C1.C1=CCC=C1’;
my $mol = Chemistry::Mol->parse($s,format=>’smiles’);
while ($patt->match($mol)) {
    @map = $patt->atom_map;
    ...
}
```

This is perhaps the most important feature of PerlMol for graph transformation. Further, PerlMol provides a standalone `awk`-like program, called `mok`, for matching molecular regular expressions written in the SMARTS language [9]. See also [15] for more details on the PerlMol project.

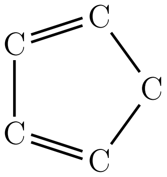
2 Chemical Graphs

Chemical descriptions can be made at different levels of resolution: a molecular descriptor uniquely identifies a molecule in a chemical database; a molecular formula indicates the number of each type of atom in a molecule; a constitutional formula or chemical graph also indicates which pairs of these atoms are bonded; and a structural formula also indicates those stereochemical distinctions that are required to uniquely identify a molecule.

Chemical graphs are just graph-based descriptions of molecules, with nodes representing the atoms, each one of them labeled by the type (the name of the corresponding element), and edges representing the bonds, with a non-negative weight describing the order of the bond (0 for a non-existent bond, 1 for a single bond, 2 for a double bond, etc.).

Definition 2.1 *A* ***chemical graph*** *is a weighted graph* (V, E, μ)*, with* (V, E) *an undirected graph (without multiple edges or self-loops) all whose nodes are labeled by means of chemical elements, and* $\mu : E \to \mathbb{N}$ *a weight function. The* ***valence*** *of a node in a chemical graph is the total weight of the edges incident to it.*

Chemical graphs can be represented as `Chemistry::Mol` objects in PerlMol. Notice that non-existent bonds need not be explicitly represented unless they are needed for modeling a chemical reaction. For instance, the 1,3-cyclopentadiene (C_5H_6) molecule,



can be represented in PerlMol as follows.

```
use Chemistry::Mol;
use Chemistry::File::SMILES;
my $s = ’C1=CCC=C1’;
my $mol = Chemistry::Mol->parse($s,format=>’smiles’);
```

This results in a `Chemistry::Mol` object containing five `Chemistry::Atom` objects and also five `Chemistry::Bond` objects. Notice that hydrogen atoms and the corresponding bonds are not explicitly represented in PerlMol. However, the number of implicit hydrogen atoms bond to a given atom and can be accessed by the `hydrogens` method, and the sum of the bond orders in which the atom participates, including all implicit hydrogens, can be accessed by the `valence` method that provides the `Chemistry::Atom` class.

```
my $impl = $atom->hydrogens;
my $val = $atom->valence;
```

3 Chemical Reaction Graphs

A chemical reaction is the change produced by two or more molecules acting upon each other. In a chemical reaction, *substrate* molecules are transformed into *product* molecules.

Chemical reactions consist of breaking, forming and changing bonds in sets of molecules. Therefore, a chemical reaction can be represented as the transformation of the chemical graph representing the reaction's substrate into the chemical graph representing the product. This transformation will satisfy a set of specific conditions. First, the number and type of the atoms in the substrate and the product must be the same, and therefore the transformation must induce the identity on the set of labeled nodes. Besides, and for simplicity, we shall restrict ourselves in this paper to chemical reactions where each individual atom has the same valence in the substrate and in the product: from the point of view of graphs, this corresponds to ensure that the total weight of edges incident to each node remains constant after the transformation.

There are about 700 chemical reactions which have come to be recognized and referred to by name within the chemistry community [8,10,11,12]. Most of them can be classified according to the pattern of atomic rearrangement, into the following four main classes.

Combination reactions Two or more substrates combine to form a single product, according to the pattern: $A+B \to AB$. For example, sodium and chlorine react to form sodium chloride: $2\,\mathrm{Na} + \mathrm{Cl}_2 \to 2\,\mathrm{NaCl}$.

Decomposition reactions A single substrate is decomposed or broken down into two or more products, according to the pattern: $AB \to A + B$. For example, calcium carbonate breaks down into calcium oxide and carbon dioxide: $\mathrm{CaCO}_3 \to \mathrm{CaO} + \mathrm{CO}_2$.

Displacement reactions One of the substrates is displaced into another one, according to the pattern: $A + BC \to AC + B$. They are also called *single displacement* reactions. For example, sodium and water displace to sodium hydroxide and hydrogen: $2\,\mathrm{Na} + 2\,\mathrm{H}_2\mathrm{O} \to 2\,\mathrm{NaOH} + \mathrm{H}_2$.

Exchange reactions One of the substrates is exchanged by another one, according to the pattern: $AB + CD \to AD + CB$. They are also called *double displacement* reactions. For example, hydrochloric acid and sodium hydroxide exchange to sodium chloride and water: $\mathrm{HCl} + \mathrm{NaOH} \to \mathrm{NaCl} + \mathrm{H}_2\mathrm{O}$.

In a more detailed representation, at the level of the constitutional formula or the structural formula, structural change of these chemical reactions can be modeled by superimposing the reactant and the product to match up the atoms and bonds that are unchanged in the transformation. A formalism called *imaginary transition structures* was introduced in [5,6,7] to model chemical reactions, where the chemical graphs representing the reactions' substrate and product are superimposed topologically, and the bonds are then

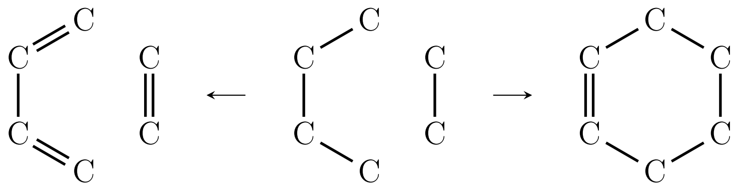


Fig. 1. Double-pushout transformation rule for the Diels-Alder reaction of Example 3.1.

distinguished and classified into three categories: *out-bonds* (bonds appearing only in the substrate molecules), *in-bonds* (bonds appearing only in the product molecules), and *par-bonds* (bonds appearing in both the substrate and the product molecules). Imaginary transition structures can be seen as double-pushout transformation rules [2] over chemical graphs: the left-hand side, context, and right-hand side are chemical graphs with set of labeled nodes corresponding to the atoms in its molecules; the left-hand side graph has edges representing out-bonds and par-bonds, the context graph has edges representing par-bonds only, and the right-hand side graph has edges representing in-bonds and par-bonds. This is, essentially, the view of chemical reactions advocated in [1].

Example 3.1 *Consider, for example, the Diels-Alder reaction [4], one of the most important reactions in organic chemistry, which is described by the double-pushout transformation rule shown in Fig. 1. The substrate of the reaction, 1,3-butadiene* (C_4H_6) *and ethylene* (C_2H_4)*, is combined to form cyclohexene* (C_6H_{10})*.*

In [13], explicit chemical reactions in a metabolic pathway are described by edge relabeling graph transformation rules. Chemical reaction graphs are just a representation of explicit chemical reactions.

Definition 3.2 *A* ***chemical reaction graph*** *is a structure* (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 following conditions:*

- *There is no* $e \in E$ *such that* $\sigma(e) = \pi(e) = 0$.
- *For every* $v \in V$*, if* $e_1, \dots, e_k$ *are the edges incident to it, then*

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

Example 3.3 *Consider again the Diels-Alder reaction of Example 3.1, which is described by the chemical reaction graph shown in Fig. 2. An edge* e *is labeled by* $\sigma(e) : \pi(e)$*. For instance, a label of the form* $1 : 0$ *next to an edge* e *means that* $\sigma(e) = 1$ *and* $\pi(e) = 0$*; that is, an existing bond is broken by the reaction.*

A chemical reaction graph is represented as a single `Chemistry::Reaction`

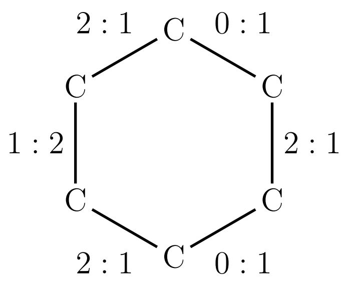


Fig. 2. Chemical reaction graph for the Diels-Alder reaction of Examples 3.1 and 3.3.

object, which is a subclass of `Chemistry::Mol`. The constructor takes a substrate `Chemistry::Mol` object, a product `Chemistry::Mol` object and a map of substrate atoms to product atoms, represented as a Perl hash.

```
use Chemistry::Reaction;
my $x = 'C=CC=C.C=C';
my $s = Chemistry::Pattern->parse($x,format=>'smiles');
$x = 'C1=CCCCC1';
my $p = Chemistry::Pattern->parse($x,format=>'smiles');
my %m;
$m{$s->atoms(1)} = $p->atoms(6);
$m{$s->atoms(2)} = $p->atoms(1);
$m{$s->atoms(3)} = $p->atoms(2);
$m{$s->atoms(4)} = $p->atoms(3);
$m{$s->atoms(5)} = $p->atoms(4);
$m{$s->atoms(6)} = $p->atoms(5);
my $react = Chemistry::Reaction->new($r,$p,\%m);
```

In a `Chemistry::Reaction` object, the order of a bond before the reaction (in the substrate molecule) is distinguished from the order after the reaction (in the product molecule). These are both represented as a single `Chemistry::Reaction::Bond` object, which is a subclass of `Chemistry::Bond` and they can be accessed by the `before` and `after` methods that provides the `Chemistry::Reaction::Bond` class.

The substrate and product molecules, as well as the map of substrate atoms to product atoms, can be recovered by the `substrate`, `product`, and `map` methods that provides the `Chemistry::Reaction` class.

4 Chemical Graph Transformation

The chemical reaction described by a chemical reaction graph, often takes place within the context of a larger molecule. In the general case of reversible chemical reactions, two different forms of chemical graph transformation neeed to be distinguished.

A chemical reaction graph can be applied to a molecule if the chemical

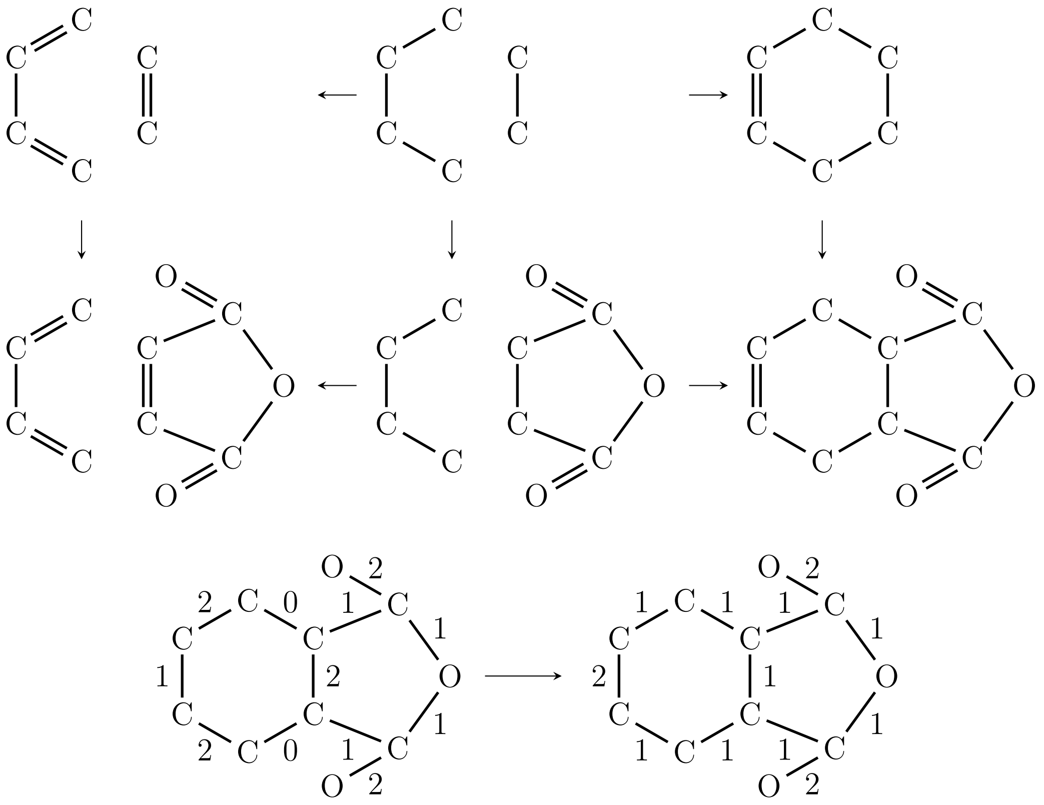


Fig. 3. Forward application of the Diels-Alder reaction of Example 3.1 as a double-pushout transformation (above) and also as an edge relabeling graph transformation (below).

reaction graph is a subgraph of the molecule. In a forward reaction, the substrate molecule is a subgraph of the molecule to be transformed while in a reverse reaction, it is the product molecule which is a subgraph of the molecule to be transformed. Non-reversible chemical reactions can only be applied forward.

Definition 4.1 *A chemical graph (V, E, μ) is a* ***subgraph*** *of a chemical graph (V', E', μ') if $V \subseteq V'$, $E \subseteq E'$ and for all edges $e \in E$, $\mu(e) \leqslant \mu'(e)$. An explicit chemical reaction (V, E, σ, π)* ***can be forward applied*** *to a chemical graph (V', E', μ) if (V, E, σ) is a subgraph of (V', E', μ) and in such a case, the* ***forward application*** *of (V, E, σ, π) to (V', E', μ) at (V, E, σ) results in a chemical graph (V', E', μ'), where $\mu'(e) = \mu(e)$ for all edges $e \in E' \setminus E$ and $\mu'(e) = \pi(e)$ for all edges $e \in E$. In a similar way, an explicit chemical reaction (V, E, σ, π)* ***can be reverse applied*** *to a chemical graph (V', E', μ) if (V, E, π) is a subgraph of (V', E', μ) and in such a case, the* ***reverse application*** *of (V, E, σ, π) to (V', E', μ) at (V, E, π) results in a chemical graph (V', E', π'), where $\pi'(e) = \pi(e)$ for all edges $e \in E' \setminus E$ and $\pi'(e) = \mu(e)$ for all edges $e \in E$.*

Example 4.2 *The forward application of the Diels-Alder reaction of Exam-*

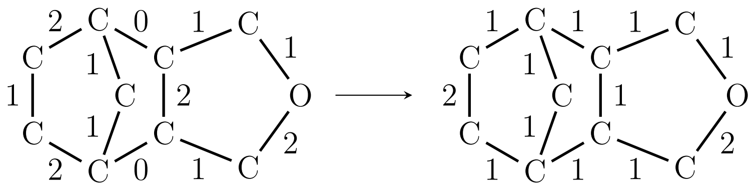


Fig. 4. Forward application of the Diels-Alder reaction of Example 3.1, as an edge relabeling graph transformation, to two copies of the same molecule.

ple 3.1 to 1,3-butadiene (C_4H_6) *and dihydro-2,5-furandione* ($C_4H_4O_3$) *to form 1,3-isobenzofurandione* ($C_8H_8O_3$)*, is shown in Fig. 3 as both a double-pushout transformation and an edge relabeling graph transformation.*

A molecule can also react with itself in a chemical reaction.

Example 4.3 *The forward application of the Diels-Alder reaction of Example 3.1 to two copies of the 1,3-cyclopentadiene* (C_5H_6) *molecule to combine into a single dicyclopentadiene* ($C_{10}H_{12}$) *molecule, is shown in Fig. 4.*

A chemical reaction graph can be forward or reverse applied to a chemical graph in two different ways, by means of the `forward` and `reverse` methods that provides the `Chemistry::Reaction` class. On the one hand, a chemical reaction graph can be forward or reverse applied at the first subgraph found of the chemical graph to be transformed which is isomorphic to the substrate or product chemical graph of the reaction.

```
if ($react->product->match($mol)) {
    @map = $react->atom_map;
    $mol->reverse($react,@map);
}
```

On the other hand, a chemical reaction graph can be forward or reverse applied at each subgraph of the chemical graph to be transformed which is isomorphic to the substrate or product chemical graph of the reaction.

```
while ($react->substrate->match($mol)) {
    @map = $react->atom_map;
    $mol->forward($react,@map);
}
```

5 Conclusion

The approach to the description of chemical reactions by edge relabeling graph transformation rules, is illustrated by the implementation in Perl of a `Chemistry::Reaction` class that takes advantage of the object-oriented representation of molecules, the interface with the most common data formats used in computational chemistry, and the powerful pattern matching capabilities of the PerlMol collection of Perl modules for computational chemistry.

The `Chemistry::Reaction` class will be incorporated into the PerlMol collection of Perl modules for computational chemistry. A preliminary version of `Chemistry::Reaction` is available from the authors.

Future work includes the extension of the chemical reaction graph formalism to take compounds formed by ionic (instead of covalent) bonding, stereochemistry, and chirality into account; the automatic determination of the map of substrate atoms to product atoms in a chemical reaction graph; the encoding of all named chemical reactions in a database of chemical reaction graphs; and the development of a chemical graph transformation system based on the `Chemistry::Reaction` class.

Acknowledgments

The authors would like to acknowledge detailed comments by Ivan Tubert-Brohman on a preliminary version of this paper.

References

[1] Benkö, G., C. Flamm and P. F. Stadler, *A graph-based toy model of chemistry*, Journal of Chemical Information and Computer Sciences **43** (2003), pp. 1085–1093.

[2] Corradini, A., U. Montanari, F. Rossi, H. Ehrig, R. Heckel and M. Löwe, *Algebraic approaches to graph transformation. Part I: Basic concepts and double pushout approach*, in: G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations*, World Scientific, 1997 pp. 163–246.

[3] Dalby, A., J. G. Nourse, W. D. Hounshell, A. K. I. Gushurst, D. L. Grier, B. A. Leland and J. Laufer, *Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited*, Journal of Chemical Information and Computer Sciences **32** (1992), pp. 244–255.

[4] Fringuelli, F. and A. Taticchi, "The Diels-Alder Reaction: Selected Practical Methods," John Wiley & Sons, Chichester, England, 2002.

[5] Fujita, S., *Description of organic reactions based on imaginary transition structures. Part 1–5*, Journal of Chemical Information and Computer Sciences **26** (1986), pp. 205–242.

[6] Fujita, S., *Description of organic reactions based on imaginary transition structures. Part 6–9*, Journal of Chemical Information and Computer Sciences **27** (1987), pp. 99–120.

[7] Fujita, S., "Computer-Oriented Representation of Organic Reactions," Yoshioka Shoten, Kyoto, 2001.

[8] Hassner, A. and C. Stumer, “Organic Syntheses based on Name Reactions,” Tetrahedron Organic Chemistry Series **22**, Pergamon Press, Oxford, England, 2002, 2nd edition.

[9] James, C. A., D. Weininger and J. Delany, *Daylight theory manual* (2004), available at `http://www.daylight.com/dayhtml/doc/theory/`.

[10] Laue, T. and A. Plagens, “Named Organic Reactions,” John Wiley & Sons, Chichester, England, 1998.

[11] Li, J. J., “Name Reactions: A Collection of Detailed Reaction Mechanisms,” Springer-Verlag, Berlin, 2003, 2nd edition.

[12] Mundy, B. P. and M. G. Ellerd, “Name Reactions and Reagents in Organic Synthesis,” John Wiley & Sons, Chichester, England, 1988.

[13] Rosselló, F. and G. Valiente, *Analysis of metabolic pathways by graph transformation*, in: *Proc. 2nd Int. Conf. Graph Transformation*, Lecture Notes in Computer Science **0000** (2004), pp. 000–000.

[14] Tubert-Brohman, I., *Perl and chemistry*, The Perl Journal **8** (2004), pp. 3–5.

[15] Tubert-Brohman, I., *PerlMol—Perl modules for molecular chemistry* (2004), available at `http://www.perlmol.org/`.

[16] Weininger, D., *SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules*, Journal of Chemical Information and Computer Sciences **28** (1988), pp. 31–36.

[17] Weininger, D., *SMILES. 3. DEPICT. Graphical depiction of chemical structures*, Journal of Chemical Information and Computer Sciences **30** (1990), pp. 237–243.

[18] Weininger, D., A. Weininger and J. L. Weininger, *SMILES. 2. Algorithm for generation of unique SMILES notation*, Journal of Chemical Information and Computer Sciences **29** (1989), pp. 97–101.

[19] Westbrook, J. D. and P. M. D. Fitzgerald, *The PDB format, mmCIF formats and other data formats*, in: P. E. Bourne and H. Weissig, editors, *Structural Bioinformatics*, Methods of Biochemical Analysis **44**, John Wiley & Sons, Hoboken, New Jersey, 2003 pp. 161–179.