Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 127 (2005) 157–166
www.elsevier.com/locate/entcs

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.

Keywords: Chemical graph, chemical reaction, explicit chemical reaction, graph transformation, edge relabeling, PerlMol.

[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).

doi:10.1016/j.entcs.2004.12.033

1 Introduction

The aim of this note is to illustrate the approach initiated by the authors in [11] to model chemical reactions by edge relabeling chemical graph transformation rules, while describing an implementation on top of PerlMol [12].

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 [17], and SMILES strings [14,15,16].

```
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::Pattern;
use Chemistry::File::SMILES;
my $s = 'C=CC=C.C=C';
my $pat = Chemistry::Pattern->parse($s,format=>'smiles');
$s = 'C1=CCC=C1.C1=CCC=C1';
my $mol = Chemistry::Mol->parse($s,format=>'smiles');
while ($pat->match($mol)) {
  @map = $pat->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 [7]. See also [13] for more details on the PerlMol project.