

A Rule-Based Approach for Automated Generation of Kinetic Chemical Mechanisms

Olivier Bournez¹, Guy-Marie Côme², Valérie Conraud²,
Hélène Kirchner¹, and Liliana Ibănescu¹ *

¹ LORIA & INRIA Lorraine

615, rue du Jardin Botanique, BP 101

F-54602 Villers-lès-Nancy Cedex, France

{Olivier.Bournez, Helene.Kirchner, Mariana-Liliana.Ibanescu}@loria.fr

² Département de Chimie Physique des Réactions (DCPR), INPL-ENSIC

1, rue Grandville, BP 451,

F-54001, Nancy, France

{Guy-Marie.Come, Valerie.Conraud}@ensic.inpl-nancy.fr

Abstract. Several software systems have been developed recently for the automated generation of combustion reactions kinetic mechanisms using different representations of species and reactions and different generation algorithms. In parallel, several software systems based on rewriting have been developed for the easy modeling and prototyping of systems using rules controlled by strategies. This paper presents our current experience in using the rewrite system ELAN for the automated generation of the combustion reactions mechanisms previously implemented in the EXGAS kinetic mechanism generator system. We emphasize the benefits of using rewriting and rule-based programming controlled by strategies for the generation of kinetic mechanisms.

1 Introduction

Combustion reactions are widely present in our everyday life, taking place in engines, burners and industrial chemical reactors, to produce mechanical or thermal energy, and also to incinerate pollutants or to manufacture chemical substances. The optimal design and operation of efficient, safe and clean chemical reactors, engines, burners, incinerators is highly desirable.

The design of combustion processes has mainly been carried out by using rather empirical models, while fundamental design, based on scientific principles, becomes more and more the main research goal (see e.g. [7,16]). However, the generation of detailed fundamental kinetic mechanisms for the combustion of a mixture of organic compounds in a large temperature field requires to consider several hundred chemical species and several thousands of elementary reactions [7]. An automated procedure is the only convenient and rigorous way to write such large mechanisms.

* Work supported by Peugeot Citroën Automobiles.

A number of software systems have been developed for this purpose. A non exhaustive list of software systems for automatic generation of detailed kinetic combustion mechanisms is the following: MAMOX [19], NetGen [20], EXGAS [7,22], COMGEN [17]. These systems are¹ implemented using traditional imperative programming, using rather ad hoc data-structures and procedures for the representation and transformations of molecules (e.g. boolean adjacency matrices and matrices transformations). Flexibility is often absent or limited to menu systems, whereas the actual use of these systems, during validation of generated mechanisms by chemists, as well as during their final use for conception of industrial chemical processes, requires modifications, activations or deactivations of involved rules according to new experimental data, reactor conditions, or chemist expertise. Furthermore, existing systems, are limited, sometimes by their technology based on ad hoc structures, to acyclic species, or mono-cyclic species, whereas combustion mechanisms often involve aromatic species, i.e. polycyclic species.

We present in this paper an alternative approach based on rule-based programming and strategies. Rule-based systems have gained considerable interest with the development of efficient compilers. Now, systems like ASF+SDF [13], Maude [5,6], Cafe-OBJ [9], or ELAN [3,4] are used for various applications like constraint solving, protocol verification, modeling of biological systems, and more. This paper presents a system, named GasEl, based on ELAN system, that generates kinetic mechanisms of fuel combustion. This is one of the objectives of a research project that involves two teams from Nancy, France: one team of computer scientists from LORIA² and a team of chemists from DCPR³ that developed the kinetic mechanism generator system EXGAS [7,22].

ELAN has some good properties for the generation of kinetic mechanisms: chemical reactions are naturally expressed using conditional rules, themselves easily understood by chemists; ELAN matching power allows for retrieving patterns in chemical species, thanks to the capability of handling multiset structures through the use of associative and commutative functions; ELAN provides a strategy language and strategy constructors to define control on rules, which appears as essential for designing generation mechanisms in a flexible way; thanks to its efficient compiler, ELAN can handle a large number of rules application (several thousands per second) and is thus well-suited to the computational complexity of the modeling.

Of course, some technical difficulties remain. One of them is that cyclic molecules are easily represented by graphs whereas ELAN can only do term rewriting. Another one is elimination of redundancies that requires intelligent search in huge data sets. The paper presents the solutions that we adopted.

The idea that rewriting techniques can be applied to chemistry is not new. Actually, even the chemical metaphor has been exploited to define computational

¹ as far as literature says

² LORIA is the Lorraine Laboratory for Research into Information Technology and its Applications

³ DCPR is the Department of Physical Chemistry of Reactions

models [2]. Indeed, kinetic elementary mechanisms looks like graph rewriting rules. However, at least in this context, existing rewriting softwares are not used in practice for the automatic generation of kinetic combustion mechanisms. One reason is that pure graph rewriting is not sufficient: one needs for example to rewrite not only graphs, but associative commutative forests of graphs, with an explicit possibility of controlling rule applications, as their order is sometimes really important or dependent of external conditions such as temperature. Furthermore, the type of involved rules is particular, in the sense that chemical rules obey to external properties such as conservation laws. We hope this paper is providing some hints to help understand the kind of rewriting which is needed in this type of industrial applications.

The paper is organized as follows: in Section 2, we position the problem of generation of kinetic mechanisms in the whole context of its use by chemists and industrial partners, and we give a brief description of the chemical problem complexity. In Section 3 we give an example of what a detailed kinetic mechanism is for a specific molecule and we give the general procedure for its generation. Section 4 presents a description of the problems and the corresponding solutions we adopted in GasEl. Conclusions and discussions are presented in Section 5.

2 Chemical Problem Description

In this section we give a short presentation of the chemists' challenges in the generation of detailed kinetic mechanisms [7,16].

Mastering the combustion reactions necessitates the elaboration and the validation of a *reaction model*; the description of the whole process is given in Figure 1:

1. The *experimental sequence* produces the experimental results for a given molecule of hydrocarbon, and consists of two phases:
 - (a) *experiments* done in laboratory reactors;
 - (b) *acquisition and treatment of physical chemical data*.
2. The *modeling sequence* has as input a model of the same molecule of hydrocarbon and gives simulation results obtained after two steps:

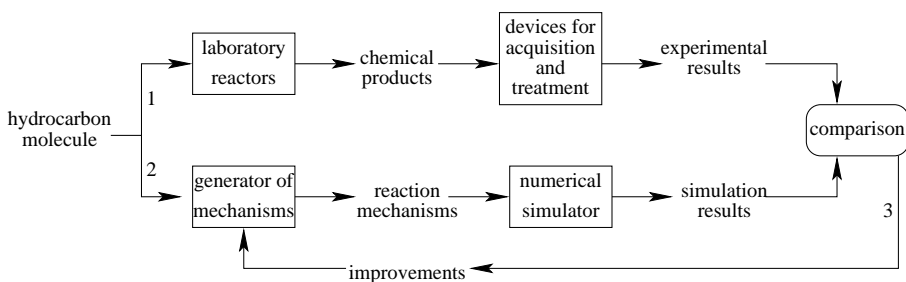


Fig. 1. Elaboration and validation process for a reaction mechanisms model

- (a) *generation of reaction mechanisms* by a generator of mechanisms (e.g. EXGAS);
 - (b) *numerical simulation* by a numerical simulator (e.g. CHEMKIN II [12]).
3. The *analysis and adjustment process* consists of:
- (a) *comparison* between the experimental results and the simulation results;
 - (b) *improvement of mechanisms generator* by introducing new generic reactions or/and fitting generic parameters.

Our system, GasEl, is intended to be a tool in the modeling sequence, a generator of detailed kinetic mechanisms for a specific area: the oxidation and combustion of fuels (e.g. Diesel fuels, petrols).

From a chemistry point of view, the complexity of the problem is induced by the composition of the following aspects:

- The *structural complexity* of hydrocarbon molecules, that can correspond to rather general graphs (planar graphs, trees and also 3D graphs) with many symmetries to be considered.
- The *complex composition of fuels or petrols*. For example, a fuel for a racing car is composed by a mixture of 98 different molecules, and the order of magnitude for the fuel of a normal petrol car is 300 different molecules, and for Diesel fuels is 3000 different molecules.
- The *combinatorial explosion* in the number of chemical species and elementary reactions modeling kinetic mechanisms. For example, 479,206 reactions and 19,052 species are considered in the simulation of tetradecane pyrolysis reported in [8], and there are 1,527 reactions and 404 species in the combustion reaction model generated with EXGAS for the n-heptane [10].
- The *duration of the validation process*. Usually, in 2–3 year, a chemical kinetic experimental PhD thesis validates experimental results for only a few molecules (1 to 2), in a couple of reactors. In the same amount of time, a PhD thesis related to modeling usually validates reaction mechanisms for only 2 or 3 molecules using and requiring several (5 to 10) previously validated experimental results for those molecules.
- The *complexity of numerical simulation*, based on partial derivative equations.
- The *multidisciplinary research* needed, including expertise from thermodynamics, quantum mechanics, transport phenomena, fluid mechanics, chemical reactors, numerical methods, experimental techniques, etc. See for example [7] for a presentation of the variety of mechanisms involved in the modeling of combustion of petrols.

3 Automated Generation of Mechanisms: Primary Mechanism

The purpose of an automated generator of detailed kinetic mechanisms is to take as input one or more hydrocarbon molecules and the reaction conditions and to

```

REACTION MECHANISMS FOR C C C C
Unimolecular Initiations
  C C C C ----->  C e +  C(e)C C
  C(C)C C ----->  C(e)C +  C(e)C
Bimolecular Initiations
  C C C C +  O=O ----->  C(e)(C C C) +  O(e)O
  C(C)C C +  O=O ----->  C(e)C C +  O(e)O
Oxidation reactions
  C(e)C +  O=O ----->  C=C +  O(e)O

```

Fig. 2. Fragment of the GasEl output for n-butane: e denote a free electron and molecules with free electrons are free radicals

give as output the list of elementary reactions applied and the corresponding thermodynamic and kinetic data.

For example, for the combustion of the n-butane, C_4H_{10} , 778 reactions are generated by EXGAS, and 164 new species are obtained. A fragment of the GasEl corresponding output is given in Figure 2.

For every reaction, a module, not discussed in this paper, needs to calculate using specific rules associated kinetic parameters (usually encoded by 3 real numbers) and, for every species associated thermodynamic parameters (14 coefficients of some polynoms in EXGAS) [7].

The generated detailed kinetic mechanisms can be the result of several phases: the “primary mechanism” can be followed by the “secondary mechanism”, usually based on lumping techniques not considered in this paper, to get even more complete descriptions of the involved mechanisms [7].

In the primary mechanism a set of ten reaction patterns should be applied to an initial mixture of molecules. A complete description of the involved reactions patterns is out of the scope of this paper, but the chemistry-like presentation from Figure 3 gives the flavor of the transformations needed to be encoded.

We mention that every reaction pattern is actually also guarded by “chemical filters”, i.e. chemical conditions of applications, not mentioned here, even if several of them are currently implemented: they include considerations on the number of atoms in involved molecules or free radicals, the type of radicals or the type of bonds, etc. Some of them are discussed in [7].

From a computer science point of view, primary mechanism can be seen as the result of several phases (see Figure 4):

1. The initiation phase: *unimolecular* and *bimolecular initiation* reactions (reaction patterns 1 and 2 in Figure 3) are applied to initial reactants, i.e. to the initial mixture of molecules. Let RS_1 be the set of all free radicals that can be obtained.
2. The propagation phase: a set of generic patterns of reactions (reaction patterns 3–8 in Figure 3) are applied to all free radicals in RS_i to obtain a new set RS_{i+1} of free radicals. RS_{i+1} consists in all free radicals of RS_i plus those that can be obtained by these reactions.

This is iterated until no new free radical is generated.

1	unimolecular initiation (ui)	$A-B \rightarrow \bullet A + \bullet B$
2	bimolecular initiation (bi)	$A=B + S-T \rightarrow \bullet A-B-T + \bullet S$
3	addition of free radicals to oxygen (ad)	$O=O + \bullet R \rightarrow \bullet O-O-R$
4	isomerisation of free radicals (is)	$\bullet A-B-TH \rightarrow AH-B-T\bullet$
5	unimolecular decomposition of free radicals by beta-scission (bs)	$\bullet A-B-T \rightarrow A=B + \bullet T$
6	unimolecular decomposition of hydroperoxy-alkyl free radicals to cyclic ethers (cy)	$\bullet A-B-O-OH \rightarrow A-B + \bullet OH$ <div style="text-align: center;"> $\begin{array}{c} \backslash / \\ O \end{array}$ </div>
7	oxidation of free radicals (ox)	$O=O + \bullet A-BH \rightarrow A=B + \bullet O-OH$
8	metathesis (me)	$\bullet R + A-H \rightarrow R-H + \bullet A$
9	combination of free radicals (co)	$\bullet R + \bullet S \rightarrow R-S$
10	disproportionation of free radicals (di)	$\bullet A-B-T + \bullet R \rightarrow A=B + R-T$

Fig. 3. Reaction patterns of primary mechanism given by emphasizing patterns like a simple (—) or double(=) bond, a free radical ($\bullet A$), a specific atom (O, H). Symbols different from atom symbols (C, O, H) are variables and can be instantiated by any radical

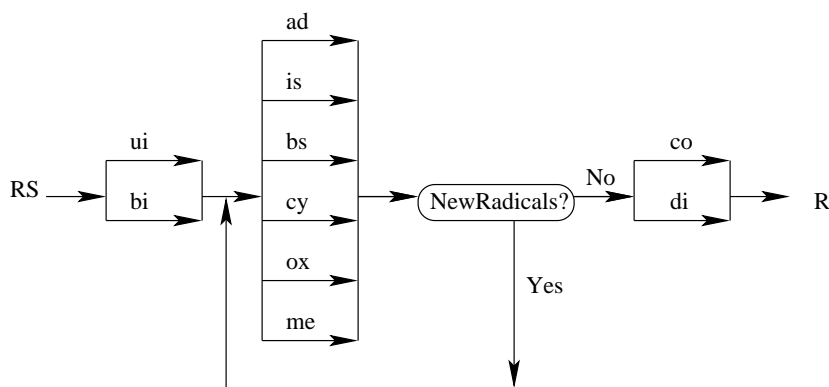


Fig. 4. The Primary Mechanism

3. The termination phase: *combination* and *disproportionation* reactions (reaction patterns 9 and 10 in Figure 3) are applied to free radicals of $\bigcup_i RS_i$ to get a set R of molecules.

4 GasEl — System Description

According to Tomlin et al. [21], a mechanism generation program should have the following features:

- generated chemical species should be stored in a form that can be easily manipulated and its notation must be unique and non-ambiguous (canonical);
- it should generate a given species of reaction only once;

- it should be able to filter out those reactions which are obviously unimportant.

The classical techniques which are known in literature to solve these issues and which are actually used in current existing softwares are the following: molecules are represented by graphs, mostly restricted to the acyclic case, or at least to the mono-cyclic case, and internally represented by their adjacency matrices, i.e. boolean square matrices. One system uses something different here: in EXGAS system, molecules are represented as tree-like structures, i.e. special symmetry-factorized trees with maximal sharing, for which some theoretical canonical representation results have been established, but only for acyclic-species [22]. For adjacency matrices, procedures to detect equalities between different graph representations of a same molecule are based on Morgan algorithm [14], or extensions (see for e.g. [11,24]), sometimes using topological indices [18]. With respect to existing systems, the solutions implemented in GasEl are original in several aspects, described in this paper.

Taking advantage from the experience of the team that developed EXGAS, the issues that we have to solve are the following:

1. We need a good *internal representation* for chemical objects: chemical species (i.e. molecules and free radicals) and mixture of species (see Section 4.1): a difficulty here is that we are using a term-rewriting system and that cyclic molecules correspond to graphs.
2. We need a way to *test* if two representations of a molecule are or not representations of a same molecules (see Section 4.3). This is vital for example to detect termination of propagation phase in primary mechanism.
3. We need a way to express into computational concepts various *chemists expertise*, like the description of reactions pattern of primary mechanisms (see Section 4.4), or the description of chemical filters (e.g. conditions on application/non-applications of rules according to temperature domain) (see Section 4.7).
4. We need to allow *flexibility and ease of modification* of the rules that are used for the generation of mechanisms, according to chemist expertise or according to new experimental data (see Section 4.7).

The rest of this paper is devoted to describe how GasEl fulfills in a original way all these requirements.

4.1 Representation of Species

We need a way to represent molecules and free-radicals. We need a priori to fix an external representation for inputs/outputs of the system, as well as a representation for the internal computations. Unlike what is classically done in existing software systems [1], we propose to use the same notation for both.

As ELAN is a term rewriting system, we use the linear notation called SMILES presented in [23]. This notation is compact and well-suited because acyclic graphs are represented as trees. We briefly recall the principles of this representation:

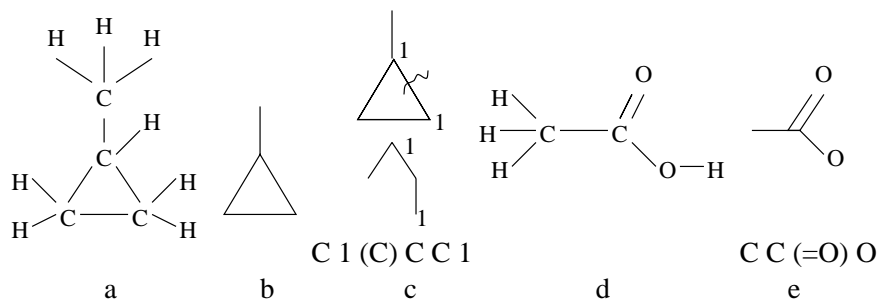


Fig. 5. Representations of molecules: a. molecular graph of methylcyclopropane; b. corresponding hydrogen-depleted molecular graph; c. corresponding SMILES notation; d. molecular graph of Acetic acid; e. corresponding hydrogen-depleted molecular graph and SMILES notation

1. Molecules are represented as hydrogen-suppressed molecular graphs.
2. If the hydrogen-suppressed molecular graph has cycles, we transform it into a tree applying the following rule to every cycle: choose one fresh digit and one single or aromatic bond of the cycle, break the bond and label the 2 atoms with the same digit.
3. Choose a root of the tree, and represent it like a concatenation of the root and the list of its sons.

For example, the term C 1 (C) C C 1 represents methylcyclopropane and the term C C (=O)O represents acetic acid: see Figure 5.

In Figure 6 we give the signature of our notation:

- atoms are represented by their atomic symbols;
- the sort `symbol` is extended to the concatenation of the sort `symbol` and `int` in order to code cycles;

```

C, c, O, o, H   : symbol;      /* Atom specification */
@@             : (symbol int) symbol; /* Labels for cycle closure specification */

-, =, #, ':'    : bond;      /* Bond specification */

e              : radical;
@             : (symbol) radical; /* Molecules and radicals specification */
@@           : (symbol radical_list) radical;

@             : (radical) radical_list; /* List of radicals specification */
@@           : (bond radical) radical_list;
(@)          : (radical_list) radical_list;
(@) @       : (radical_list radical_list) radical_list (AC);

```

Fig. 6. GasEl signature of molecules and free radicals notation

- single, double, triple and aromatic bonds are represented by the symbols “_”, “=”, “#”, “:”, respectively and belong to sort `bond`;
- a molecule is represented as a root and the list of sons that belongs to sort `radical_list`;
- the user definition of the the list of sons has a particular form, inspired by the chemical notation SMILES (see e.g. [23]) and is defined using an associative-commutative operator;
- a special symbol `e` of sort `radical` is introduced for the representation of free radicals (a free radical `•R` is a molecule in which an atom has a free electron).

4.2 Representation of Mixtures of Species

We need a way to represent mixtures of molecules and free radicals. We propose to benefit here from the Associative Commutative matching possibilities of ELAN: in ELAN, a mixture has the sort `reactif` using the following signature

```
@      : (radical) reactif;
@ + @  : (reactif reactif) reactif (AC);
```

For example, `C C C C + O=O + C C (= O)O` is a term that represents a mixture of n-butane, oxygen and acetic acid.

4.3 Equality Test for Species

As pointed out by Tomlin et al. in [21], chemical species should be stored in a mechanism generation program in a canonical form: i.e. a unique and non-ambiguous form. Classical canonicity algorithms in this context are either based on transformations applied to the adjacency matrices of the molecule [11,14,24], or restricted to the acyclic case when considering other ad hoc data structures [22].

In our term-based representation, a molecule is a tree. Different choices of the root induce different trees representing the same molecule. We call these trees the *visions* of the molecule: see Figure 7.

Testing if two terms represent the same molecule is done in 2 steps:

1. An operator `AllVis` generates all the visions of a molecule by choosing every node of the tree representing the molecule to be the root.
2. Two terms M_1 and M_2 are equivalent if M_1 and M_2 have the same number of atoms and if M_1 is a vision of M_2 :

$$\text{is_eq}(M_1, M_2) \Rightarrow \text{true if } \text{no_at}(M_1) = \text{no_at}(M_2) \text{ and } M_1 \in \text{AllVis}(M_2)$$

Our algorithm is exponential. However, recall that no polynomial algorithm is known for the graph isomorphism testing problem. The problem is proved to be **NP** but the question if it is complete or polynomial is a well known open problem [15].

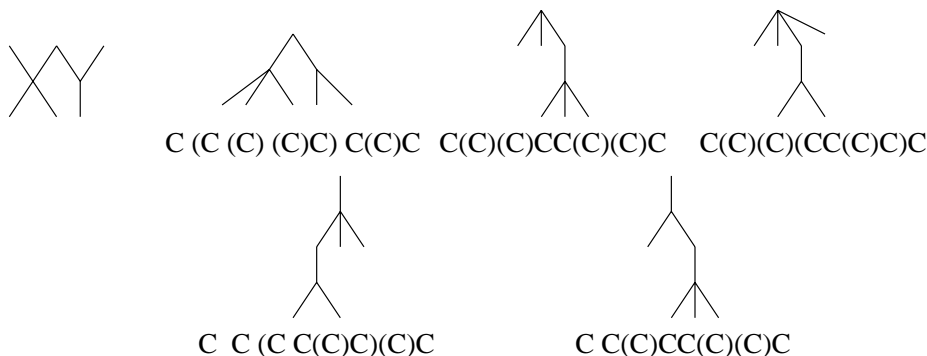


Fig. 7. Visions of ISO-octane: all the distinct visions modulo commutativity

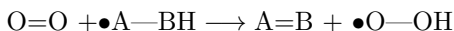
The current version is clearly not optimal. Improvements are currently investigated, in particular classical chemical techniques based on topological indices [18] can be considered.

4.4 Encoding Chemical Reactions

Generic elementary reactions are transformations applied to a mixture of molecules and free radicals.

We need to express the ten elementary reactions of Figure 3. *Addition of free radicals to oxygen, oxidation of free radicals, combination of free radicals and disproportionation of free radicals* are expressed directly as rewrite rules: the generic reaction, expressed as a graph transformation is encoded directly into a term rewrite rule.

For example, the generic reaction of oxidation of free radicals is the following:



This means, chemically-speaking, that a molecule of oxygen, $\text{O}=\text{O}$, abstracts an H atom in the β position with respect to the radical point, with the formation of the free radical $\bullet\text{O}-\text{OH}$ and an unsaturated molecule (a molecule with a double bond).

This corresponds, in other words, to the graph transformation given in Figure 8 and is coded directly by the following ELAN rewrite rule:

```
[ox] O=O + A(e)(-B x) y => A (=B x) y + O(e)O
      if no_H(B x) >= 1
      end
```

where A and B are variables of sort `symbol`, x and y are variables of sort `radical_list` and the function `no_H(R)` returns the number of implicit hydrogen atoms in the root of R.

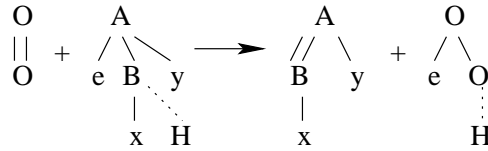
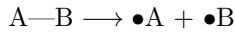


Fig. 8. Oxidation of free radicals: a dotted line correspond to an (implicit) hydrogen atom

4.5 Exhaustive Application of Reactions to a Given Molecule

The other generic chemical reactions require more work. We take the example of *unimolecular initiation*. An unimolecular initiation consists in breaking a single bond of a molecule and is represented, in the acyclic case, as the following reaction pattern:



Breaking a C—C bond which does not belong to a cycle corresponds to the graph transformation given in Figure 9.

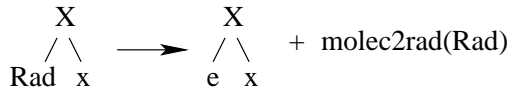


Fig. 9. Unimolecular initiation

Similarly to Section 4.4, we obtain the following ELAN rewrite rule:

```
[ui] X (Rad) x => X(e) x + molec2rad(Rad) end
```

where X is a variable of sort `symbol`, `Rad` is a variable of sort `radical` and `x` is a variable of sort `radical_list`. The operator `molec2rad()` transforms its argument into a free radical and is defined by two ELAN rewrite rules:

```
[] molec2rad(X Rad) => X (e) Rad end
>[] molec2rad(X (Rad) x) => X (e) (Rad) x end
```

The power of the associative-commutative matching of ELAN allows us to give generic ELAN rewrite rule [ui] for the unimolecular initiation that will be applied to all sons of root X .

The previous rewrite rules have to be applied everywhere inside the terms and based on the semantics of the strategy language of ELAN, to apply a named rewriting rule (e.g. [ui]) to all sub-terms we adopted the following technique:

1. Apply the operator `AllVis` to the molecule `X (Rad) x`.
2. Apply the generic ELAN rewrite rules for unimolecular initiation to every vision of the molecule given by the `AllVis` operator.

For cyclic molecules, we have to simulate (a restricted type of) graph-rewriting using term-rewriting.

To deal with cyclic molecules in the *unimolecular initiation*, we need first to test if the bond to be broken is on a cycle or not; in the affirmative case one single free radical is generated by the fusion operator; if the bond to be broken is not on a cycle, the rewrite rule is similar to the acyclic case. Therefore, we need to add the following rule:

```
[ui_cycle] X (Rad) x => fusion(X(e) x, molec2rad(Rad))
      where labels_1 := () get_cycle_label (X x)
      where labels_2 := () get_cycle_label (Rad)
      if common(labels_1,labels_2)
      end
```

and to change previous rule [ui] in

```
[ui_CC] X (Rad) x => X(e) x + molec2rad(Rad)
      where labels_1 := () get_cycle_label (X x)
      where labels_2 := () get_cycle_label (Rad)
      if not(common(labels_1,labels_2))
      end
```

4.6 Encoding the Primary Mechanism

Using the power of strategies of ELAN, the primary mechanism is defined in a natural way. This corresponds to the concatenation of three strategies corresponding to each phase, `tryInit` for the initiation phase, `tryPropag` for the propagation phase and `tryTermin` for the termination phase:

```
[] mec_prim    => tryInit; tryPropag; tryTermin end
```

The user defined strategies `tryInit` and `tryTermin` are easily expressed using the ELAN choice strategy operator `dk` applied to the strategies (the ELAN rewrite rule) defining the generic reactions. The `dk` operator (*dont know choose*) takes all strategies given as arguments and returns, for each of them the set of all its results.

```
[] tryInit    => dk(ui, bi) end
```

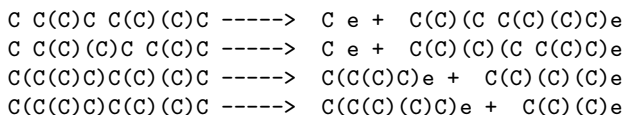
The output of the initiation phase applied to ISO-octane is illustrated in Figure 10.

Strategy `tryPropag` is defined as the iteration of one step of propagation using the ELAN strategy iterator `repeat*`:

```
[] tryPropag => repeat*(propagOne) end
```

Strategy `repeat` iterates the strategy until it fails and returns the terms resulting from the last unfailling call of the strategy. Strategy `propagOne` is defined in a similar way as `tryInit` using a `dk` operator applied to the generic reactions of the propagation phase.

Unimolecular Initiations



Bimolecular Initiations

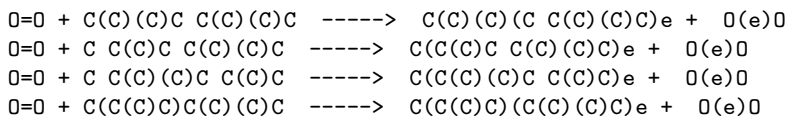


Fig. 10. Initiation reactions of ISO-octane combustion

4.7 Flexibility and Control of Application of Elementary Reactions

The previous techniques allows a great flexibility through the power of matching and of the strategy language of ELAN: for examples

- Chemical filters, such as testing if a free radical is β , $\beta\mu$, μ , or Y is easily encoded by matching against corresponding patterns.
- Modifying the set of applied generic chemical reactions in primary mechanism correspond to a direct and natural modification of the corresponding strategy.
- Activating/Deactivating application of generic chemical reactions to cycles corresponds to a simple modification of strategy: replace for example `dk(ui_cycle,ui_CC)` by `dk(ui_CC)`.

5 Conclusions and Discussions

In this paper, we describe our experience in building a system, named `GasEl`, based on system ELAN, for the automatic generation of chemical kinetic combustion mechanisms.

The main innovative feature of our system, compared to the existing ones (`MA-MOX` [19], `NetGen` [20], `EXGAS` [7,22]), is that it can handle polycyclic molecules.

Compared to `EXGAS`, `GasEl` is much more flexible thanks to a modular design, to the rule-based formalism and to the ELAN strategies language.

The outcomes of the `GasEl` project, that involve chemists from the team that developed `EXGAS` system, and computer scientists from the team that developed ELAN, are twofold:

Chemists got a new system that extends the generation of combustion mechanisms to (poly)cyclic species. The fact that `GasEl` is now considered to be the successor of `EXGAS` shows that they understand the benefits from the presented approach. Further work is to provide them with a friendly interface, allowing to visualize cyclic molecules, to easily introduce new rules and design new strategies, or to connect this tools more directly with existing computer softwares, in order for example to get automatic association of thermodynamic and kinetic parameters to reactions and reactants.

Computer scientists got in modeling of the generation of detailed kinetic mechanisms a real challenge for testing the benefit from using rewriting and rule-based programming controlled by strategies. This experience teaches several things. First, that existing chemical software, at least in the context of automatic generation of chemical kinetic combustion mechanisms, do not use really rewriting techniques, but rather ad hoc structures and techniques. Second, that one reason that explains this fact is that the type of things involved is not pure term-rewriting, nor pure graph-rewriting, nor something directly already present in existing software: as the paper shows, we want “easily-controllable molecule transformations”, with the always present needs of testing if a given generated molecule or radical is already obtained. As molecules can be cyclic, and hence correspond to rather general graphs, what we need corresponds to a kind of associative and commutative forest of graphs rewriting controlled by strategies. Since, no existing rewriting software is currently able to deal directly with this kind of features, and taking the advantage from our experience on the ELAN system, we actually developed the application using ELAN, i.e. associative and commutative term rewriting controlled by strategies.

Of course this approach has drawbacks: for e.g. the techniques that are used to emulate graph rewriting can clearly be improved in efficiency. However, as the discussion shows, when repositioning this system in chemist world (see for e.g. the number of processes that can be validated in a PhD) this computer-scientist efficiency issue may not be such an important drawback.

Moreover our approach revealed three further interesting points. First, the time for the development of the tool (two years now) can be compared to the time required for the development of similar features in EXGAS system, even if of course we are using their experience. Second, even if we can not rewrite graphs in ELAN, we have associative and commutative matching and strategies, and hence, we have almost all the required features. And third, and this is the main point, we believe this work has given hints about what is really required to address this type of industrial motivated applications.

Future work includes significant computer tests and chemists validation and, of course offering all the features of EXGAS system, providing a friendly interface, or connecting GasEl with other computer systems, but also to better understand this latter point. For example, this is clear that chemical rules always obey some conservation laws. Can we characterize abstractly chemical rules? Can we characterize the associated rewriting theory?

References

1. John M. Barnard. Structure representation. In Paul von Rague Schleyer, editor, *The Encyclopedia of Computational Chemistry*, pages 2818–2826. John Wiley & Son Ltd, 1998.
2. Gérard Berry and Gérard Boudol. The Chemical Abstract Machine. In *Proceedings of POPL'90*, pages 81–94. ACM, 1990. Also as research report INRIA 1133.

3. Peter Borovanský, Horațiu Cîrstea, Hubert Dubois, Claude Kirchner, Hélène Kirchner, Pierre-Etienne Moreau, Quang-Huy Nguyen, Christophe Ringeissen, and Marian Vittek. *ELAN V 3.6 User Manual*. LORIA, Nancy (France), september 2002.
4. Peter Borovanský, Hélène Kirchner, Pierre-Etienne Moreau, and Christophe Ringeissen. An overview of ELAN. In Claude Kirchner and Hélène Kirchner, editors, *Second Workshop on Rewriting Logic and its Applications WRLA'98*, volume 15 of *Electronic Notes in Theoretical Computer Science*, Pont-à-Mousson (France), 2000. Elsevier Science Publishers. URL: <http://www.elsevier.nl/locate/entcs/volume15.html>.
5. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and J.F. Quesada. Towards Maude 2.0. In Kokichi Futatsugi, editor, *WRLA2000, the 3rd International Workshop on Rewriting Logic and its Applications, September 2000, Kanazawa, Japon*. Electronic Notes in Theoretical Computer Science, 2000.
6. Manuel Clavel, Steven Eker, Patrick Lincoln, and José Meseguer. Principles of Maude. In José Meseguer, editor, *Proceedings of the first international workshop on rewriting logic*, volume 4, Asilomar (California), September 1996. Electronic Notes in Theoretical Computer Science.
7. Guy-Marie Côme. *Gas-Phase Thermal Reactions. Chemical Engineering Kinetics*. Kluwer Academic Publishers, 2001.
8. M.J. De Witt, D.J. Dooling, and Linda J. Broadbelt. Computer generation of reaction mechanisms using quantitative rate information: Application to long-chain hydrocarbon pyrolysis. *Ind. Eng. Chem. Res.*, 39(7):2228–2237, 2000.
9. K. Futatsugi and A. Nakagawa. An overview of CAFE specification environment – an algebraic approach for creating, verifying, and maintaining formal specifications over networks. In *Proceedings of the 1st IEEE Int. Conference on Formal Engineering Methods*, 1997.
10. Pierre-Alexandre Glaude. *Construction automatique et validation de modèles cinétiques de combustion d'alcane et d'éthers*. PhD thesis, Institut National Polytechnique de Lorraine - ENSIC - Nancy, France, 1999.
11. Ovidiu Ivanciuc. Canonical numbering and constitutional symmetry. In Paul von Rague Schleyer, editor, *The Encyclopedia of Computational Chemistry*, pages 167–182. John Wiley & Son Ltd, 1998.
12. R. J. Kee, F. M. Rupley, and J. A. Miller. CHEMKIN II. Technical Report SAND89-8009B, Sandia National Laboratories, Livermore, 1993.
13. Paul Klint. A meta-environment for generating programming environments. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2(2):176–201, 1993.
14. H.L. Morgan. The generation of a unique machine description for chemical structures - a technique developed at Chemical Abstracts Service. *J. Chem. Doc.*, 5:107–113, 1965.
15. Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
16. M.J. Pilling, editor. *Low-temperature Combustion and Auto-ignition*, volume 35 of *Comprehensive Chemical Kinetics*. Elsevier, Amsterdam, 1997.
17. Artur Rakiewicz and Thanh N. Truong. Application of chemical graph theory for automated mechanism generation. *accepted for publication*, 2002.
18. Milan Randić. Topological indices. In Paul von Rague Schleyer, editor, *The Encyclopedia of Computational Chemistry*, pages 3018–3032. John Wiley & Son Ltd, 1998.
19. E. Ranzi, T. Faravelli, P. Gaffuri, and A. Sogaro. Low-temperature combustion: automatic generation of primary oxydation reaction and lumping procedures. *Combustion and Flame*, 102:179, 1995.

20. Roberta G. Susnow, Anthony M. Dean, William H. Green, P. Peczak, and Linda J. Broadbelt. Rate-based construction of kinetic models for complex systems. *The Journal of Physical Chemistry A*, 101:3731–3740, 1997.
21. Alison S. Tomlin, Tamás Turányi, and Michael J. Pilling. *Low-temperature Combustion and Autoignition*, volume 35 of *Comprehensive Chemical Kinetics*, chapter 4. Mathematical tools for the construction, investigation and reduction of combustion mechanisms, pages 293–437. Elsevier, Amsterdam, 1997.
22. Valérie Warth, Frédérique Battin-Leclerc, René Fournet, Pierre-Alexandre Glaude, Guy-Marie Côme, and Gérard Scacchi. Computer based generation of reaction mechanisms for gas-phase oxidation. *Computers and Chemistry*, 24:541–560, 2000.
23. David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Science*, 28:31–36, 1988.
24. David Weininger, Arthur Weininger, and Joseph L. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Science*, 29:97–101, 1989.