Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 299 (2013) 69–84

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

An Algorithm for the Identification of Components in Biochemical Pathways

Giovanni Pardini[1] Paolo Milazzo[2] Andrea Maggiolo-Schettini[3]

Dipartimento di Informatica, Università di Pisa
Largo B. Pontecorvo 3, 56127 Pisa, Italy

Abstract

Biochemical pathways are abstract descriptions of the interactions among the molecular species involved in a process. Different molecular species mentioned in a pathway often represent different states of the same biological entity (e.g. the unbound and bound states of a certain molecule). Hence, a pathway can be seen as a network of interactions between entities which may change state synchronously by means of reactions. We consider such biological entities as pathway components and define a semi-automatic algorithm to infer the components from their interactions described in the pathway. Since this problem is inherently ambiguous, interaction with a domain expert might be needed to resolve any ambiguity that should arise. We apply the algorithm to the identification of components in a model of the EGF signaling pathway from the literature, and discuss possible uses of the component categorization as regards (i) the extraction of subpathways by projecting over a subset of components, and (ii) the automatic translation into finite state automata or process algebra terms.

Keywords: biochemical pathways, modular modeling, formal methods

1 Introduction

Biochemical pathways are networks of interactions between biological entities such as proteins, DNA, RNA and other molecules, taking place inside cells. The interactions constituting a biochemical pathway are typically bindings and unbindings of molecular species, syntheses and degradations of proteins, conformational changes and translocations of molecules (or complexes) from one compartment to another. Each biological entity involved in a pathway usually appears in several different molecular species mentioned in the pathway. For instance, the same protein may appear in its initial form (as just synthesized), but also in an activated form (e.g. phosphorylated) and as a part of some complexes. The forms that a biological entity may take are usually represented as different molecular species, namely with

[1] Email: pardinig@di.unipi.it
[2] Email: milazzo@di.unipi.it
[3] Email: maggiolo@di.unipi.it

http://dx.doi.org/10.1016/j.entcs.2013.11.007

different names. Moreover, complexes (that involve different biological entities) are often associated with a single name, that may not allow the originating biological entities to be easily identified. The reconstruction of the set of biological entities involved in a given pathway is the problem we face in this paper.

In [14] we proposed a modular verification approach based on [13] that allows properties of the pathway to be verified efficiently by applying model checking to an abstraction of the pathway semantics obtained by focusing on the behavior of a subset of the involved biological entities. To this aim we considered a notion of *molecular component* that is the counterpart in a pathway model of the notion of biological entity in the real world. In order to be able to identify molecular components, we assumed molecular species representing complexes to be replaced by as many different species as are the biological entities involved in it. For instance, if a complex C is obtained by the binding of two different proteins A and B, we assumed C to be replaced by C_A and C_B, where C_A is the part of C representing the bound form of protein A, and C_B is the part of C representing the bound form of protein B.

Under such an assumption the reactions of a given pathway can be rewritten in a *normal form* with as many reactants as products. For example, reaction $A, B \to C$ describing the formation of complex C can be rewritten as $A, B \to C_A, C_B$. Moreover, in each obtained reaction we can impose a positional correspondence between reactants and products such that the reactant in the i-th position must be the same biological entity of the product in the same position (as it happens with A and C_A, and with B and C_B).

Once all of the reactions of a pathway are in normal form, it is rather easy to identify molecular components. Such components essentially consist in sets of names of molecular species mentioned in the pathway. Each element of a molecular component (a name) represents a different form of the same biological entity represented by the component. As a consequence, different components do not share any element and all of the names mentioned in the pathway belong to some component. In other words, the set of components of a pathway is a partition of the set of names of biochemical species mentioned in the normal-form pathway.

In this paper we propose an algorithm for transforming pathways into their corresponding normal forms with molecular components specified. Moreover, we show that once the molecular components are identified it is rather easy to perform syntactic transformations aimed at simplifying the visualization of the pathway itself by focusing on a subset of components (namely, on a subset of the involved biological entities). Furthermore, we show that identification of molecular components also allows formal descriptions of the pathway by means of automata or process algebra (such as those in [2,3,6,9,12,19,20]) to be automatically generated. This enables the application of formal methods such as model checking [10,17] and bisimulation [7] to analyse and compare behaviours of pathways.

The proposed normalization algorithm is however not completely automatic, since in general there might be situations in which ambiguities on how to normalize some reactions cannot be avoided. The algorithm is designed to invoke human intervention when one of these ambiguities occurs. When a human intervention is necessary, the algorithm asks the human (a domain expert) to normalize a single reaction. There might be cases in which human intervention is invoked more than once to solve different ambiguities in the same pathway. However, we believe that in most practical cases the need of human intervention will be very limited, if not absent. This belief is supported by our preliminary tests on SBML models from the BioModels database [18].

As an application, we consider the well-known EGF signaling pathway. In particular, we consider the computational model developed by Schoeberl et al. in [21]. On this model we show that our algorithm (implemented in a Java prototype tool) can be successfully applied to infer molecular components from the reactions constituting the pathway. In this case no human intervention is needed, and the molecular components identified by the algorithm correspond to the actual biological entities involved in the pathway. We show also how to consider component-based subpathays and how to automatically generate a set of finite state automata, one for each molecular component.

As related work we mention [16] where the authors propose to use graph-matching techniques for relating different SBML models of the same biological processes, such as those available in public Internet databases. Comparisons between models is formalized by means of abstraction/reduction operations on the reactions of the models.

2 Modelling notation for biochemical pathways

In this section we recall the modeling notation for biochemical pathways presented in [14]. Pathways are networks of biochemical reactions occurring within a cell. Reactions can be influenced by catalysts and inhibitors, which are molecules (proteins) which can stimulate and block the occurrence of reactions, respectively. For the sake of simplicity we do not consider inhibitors in this paper, although they could be dealth with exactly as catalysts.

Given an infinite set of species Σ, a *reaction* has the form

$$R: \; r_1, \ldots, r_n \to p_1, \ldots, p_m \quad \{c_1, \ldots, c_w\}$$

where all r_i, p_i, and c_i are in Σ. Intuitively, $r_1, \ldots, r_n$ denotes the list of reactants, $p_1, \ldots, p_m$ denotes the list of products, and $c_1, \ldots, c_w$ denote the catalysts. Given a reaction R we define $\mathit{re}(R) = \{r_1, \ldots, r_n\}$, $\mathit{pro}(R) = \{p_1, \ldots, p_m\}$, and $\mathit{cat}(R) = \{c_1, \ldots, c_w\}$. We denote the set of species involved in reaction R as $\mathit{species}(R) = \mathit{re}(R) \cup \mathit{pro}(R) \cup \mathit{cat}(R)$.

A normal-form reaction is assumed to have as many products as reactants and also a positional correspondence between reactants and products is assumed. In fact, we consider a species as a "state" of a more abstract biological entity, and a reaction as a synchronized state change of a set of such entities where each r_j and p_j are the states of the corresponding entity before and after the reaction. Formally, a *normal-form* reaction is identified by the syntactic constraint that the number of reactants and products is the same, i.e. $n = m$. In order to link each reactant of a reaction with the corresponding product we assume, for technical reasons, a more general form of correspondence than the positional one. Given a normal-form reaction $R: \; r_1, \ldots, r_n \to p_1, \ldots, p_n$, we define the mapping $\mu_R \subseteq \Sigma \times \Sigma$ between reactants and products as $\mu_R = \{(r_i, p_i) \,|\, 1 \leq i \leq n\}$. Normal form reactions are just aimed at making possible identification of molecular components. Note that catalysts in reactions do not contribute to component identification since they do not change their state when a reaction they catalyse occurs. The same would hold for inhibitors.

A *pathway* P is composed of a set of reactions, $P = \{R_1, \ldots, R_k\}$. Similarly to single reactions, we denote by *species*(P) the set of species occurring in the reactions of the pathway, namely *species*$(P) = \bigcup_{R \in P}$ *species*(R). The notion of normal-form is also extended to pathways, i.e. a pathway is in normal form iff it contains only normal-form reactions. We denote by μ_P the union of all the mappings between reactants and products of pathway reactions, namely $\mu_P = \bigcup_{R \in P} \mu_R$.

The components appearing in a normal-form pathway P are described by a *classification* for P. Formally, a classification $\Gamma \subseteq \Sigma \times \Sigma$ for a pathway P is any equivalence relation such that $\mu_P \subseteq \Gamma$. From a different point of view, components can be identified by the partitioning of the species *species*(P) induced by the classification Γ. In the following, we are interested in the *least* classification, namely the least equivalence relation which contains μ_P. Given a normal-form pathway, its least classification is unique.

2.1 Identification of components

The purpose of this paper is to develop an algorithm to identify the components of a pathway, in order to enable the translation of pathway reactions into equivalent normal-form reactions, with positional correspondence between reactants and products specified. For instance, a pathway $P = \{A, B \to C,\; C \to B, A\}$ could be translated into a normal-form pathway $P' = \{A, B \to C_A, C_B;\; C_A, C_B \to A, B\}$, for some fresh symbols C_A, C_B denoting bound molecules A and B forming complex C, respectively. In this case, the reactants/products mapping is $\mu_{P'} = \{(A, C_A), (B, C_B)\}$. Thus the pathway is composed of the two components $\{A, C_A\}$ and $\{B, C_B\}$ induced by the least classification Γ over $\mu_{P'}$, which is $\Gamma = \{(A, C_A), (C_A, A), (B, C_B), (C_B, B), (A, A), (B, B), (C_A, C_A), (C_B, C_B)\}$.

Notice that, as in the previous example, if the pathway is not in normal form, it may be needed to split one or more species into different subspecies, by introducing new symbols for denoting them. In general, this process may cause the number of components to increase. For example, a single reaction $A \to B$ can be rewritten into any reaction $A_1, \ldots, A_n \to B_1, \ldots, B_n$, for any $n > 0$. Thus, by identifying the components $\{A_i, B_i\}$ for all $i \in \{1, \ldots, n\}$, we can identify any number of components from the single reaction $A \to B$. In general, this process is correct with respect to the real-world behavior. In fact, in our setting, such a single reaction $A \to B$ may actually correspond to a real-world reaction involving any number of components. However, the aim of the algorithm is to infer the "minimal" number of components to allow the pathway to be transformed into normal form. In other words, we are only interested in those components which are "observable" from the context, namely from the set of reactions in which they are involved.

On the other hand, given a normal-form pathway, it may be possible to collapse a set of symbols into a single symbol and still obtain a normal-form pathway with less components. For instance, let us consider pathway $P = \{C_1, C_2 \to D_1, D_2,\ D_1, D_2 \to E_1, E_2\}$, with the two components $\{C_1, D_1, E_1\}, \{C_2, D_2, E_2\}$. By collapsing C_1, C_2 into C, D_1, D_2 into D, and E_1, E_2 into E, such a pathway can be rewritten as $P' = \{C \to D, D \to E\}$, having a single component made of $\{C, D, E\}$.

Therefore, if a pathway is already in normal-form, no splits should happen. On the other hand, if some split occurs, then we expect the final normal-form pathway to be minimal, namely it should not be possible to collapse new symbols and obtain a normal-form pathway with a smaller number of components.

Let P be a normal-form pathway, then P is *minimal* with respect to a classification Γ for P and to a set of new symbols Λ iff there is not any set, among the sets forming the partition induced by Γ, which is made only of symbols from Λ. Formally P is *minimal* w.r.t. Γ and Λ iff $\forall x \in \Lambda. \exists (x, y) \in \Gamma. y \notin \Lambda$. For example, let us consider a pathway $P = \{F \to G_1, G_2\}$. By splitting F into new symbols F_1, F_2, we can obtain the normal-form pathway $P' = \{F_1, F_2 \to G_1, G_2\}$, with the two components $\{F_1, G_1\}, \{F_2, G_2\}$, which is minimal w.r.t. the new symbols F_1, F_2. On the other hand, if we were to split both F into F_1, F_2, F_3, and G_2 into G_{2a}, G_{2b}, we would obtain pathway $P'' = \{F_1, F_2, F_3 \to G_1, G_{2a}, G_{2b}\}$ with components $\{F_1, G_1\}, \{F_2, G_{2a}\}, \{F_3, G_{2b}\}$ and new symbols $\{F_1, F_2, F_3, G_{2a}, G_{2b}\}$. In this case, pathway P'' is not minimal, since sets $\{F_2, G_{2a}\}$ and $\{F_3, G_{2b}\}$ of the classification are composed only of newly-introduced symbols.

3 An algorithm for inferring pathway components

The algorithm is conceptually composed of two alternating phases: in the first phase, it tries to transform the set of reactions into reactions having the same number of reactants and products, but which are not necessarily in normal-form since the correspondence between reactants and products may not be completely specified. Once this phase is completed, if there are any remaining reactions for which either the number of reactants and products differ, or for which the correspondence reactants/products is not completely specified, then the algorithm requires user intervention to resolve the ambiguity (second phase). In such a case, the two phases are repeated, since user input may be used to resolve other ambiguous reactions (and possibly all of them). This procedure is repeated until a normal-form pathway is obtained.

Algorithm 1 (*findComponents*) formalizes the main cycle which (i) tries to resolve automatically a pathway using Algorithm 2 (*transform*), and then, if there are any remaining ambiguities, (ii) asks the user. Algorithm *findComponents* takes a pathway $P = \{R_1, \dots, R_k\}$ as input, and returns a tuple $(\Gamma, \theta_1, \dots, \theta_k)$, where (i)

Algorithm 1 findComponents(P: Pathway)

```
1:  let $P = \{R_1, \ldots, R_k\}$
2:  let $\Gamma = \{(\sigma, \sigma) \mid \sigma \in \mathit{species}(P)\}$
3:  for all $i \in \{1, \ldots, k\}$ do
4:      $\theta_i = \{(\mathit{re}(R_i), \mathit{pro}(R_i))\}$
5:  end for
6:  repeat
7:      let $(\Gamma, \theta_1, \ldots, \theta_k) = \mathrm{transform}(\Gamma, \theta_1, \ldots, \theta_k)$
8:      let $(\theta'_1, \ldots, \theta'_k) = (\theta_1, \ldots, \theta_k)$
9:      if $\exists i, u, v. (u, v) \in \theta_i$ and $(|u| > 1$ or $|v| > 1)$ then
10:         ask user to resolve $\theta_i$, by rewriting it in normal form
                $x_1, \ldots, x_n \to y_1, \ldots, y_n$
            with $\{x_1, \ldots, x_n\} \supseteq \bigcup_{(u,v) \in \theta_i} u$, $\{y_1, \ldots, y_n\} \supseteq \bigcup_{(u,v) \in \theta_i} v$
            and providing a set of replacements $[w_1/z_1, \ldots, w_h/z_h]$
11:         $\forall j \neq i.\ \theta'_j = \theta_j[w_1/z_1, \ldots, w_h/z_h]$
12:         $\theta'_i = \{(x_j, y_j) \mid j \in \{1, \ldots, n\}\}$
13:         $\Gamma' = \Gamma \oplus \{(x_j, y_j) \mid j \in \{1, \ldots, n\}\}$
14:     end if
15: until $\Gamma' = \Gamma$ and $\forall i.\ \theta'_i = \theta_i$
16: return $(\Gamma, \theta_1, \ldots, \theta_k)$
```

$\Gamma \subseteq \Sigma \times \Sigma$ is a classification over *species*(P), and (ii) for each reaction R_i, the set $\theta_i \subseteq \mathcal{P}(\Sigma) \times \mathcal{P}(\Sigma)$ consists of the pairs $\{(u_1, v_1), \ldots, (u_k, v_k)\} \subseteq \mathit{re}(R_i) \times \mathit{pro}(R_i)$ of singleton sets, i.e. for each index j, $u_j = \{x_j\}, v_j = \{y_j\}$ for some symbols x_j and y_j. Each pair x_j, y_j describes the correspondence between reactant x_j and product y_j, namely they are part of the same component. In fact, $\{x_1, \ldots, x_k\} = \mathit{re}(R_i)$, and $\{y_1, \ldots, y_k\} = \mathit{pro}(R_i)$. However, if there are multiple reactants/products from the same component, the algorithm cannot actually univocally match each reactant with each product, since the information provided by components is not sufficient to distinguish different subspecies of the same components. Therefore, whenever there are multiple pairs $(x_1, y_1), \ldots, (x_l, y_l)$ of elements from the same component, it is intended that such a mapping is arbitrarily chosen by the algorithm. In such cases further help from the user is needed, in order to specify the correct mapping between multiple reactants/products which occur within the same component.

As regards Algorithm 1, it initially constructs a classification $\Gamma = \{(\sigma, \sigma) \mid \sigma \in \mathit{species}(P)\}$, namely each symbol is in a different component from any other symbol. As regards each reaction R_i, the set θ_i is initialized with a single tuple $(\mathit{re}(R_i), \mathit{pro}(R_i))$. Conceptually, this means that we initially assume all the reactants to be in the same component as all the products. Set θ_i is going to be iteratively refined by Algorithm 2, by splitting tuples into subtuples whenever some certain associations between reactants and products can be derived. Moreover, algorithm *transform* updates the classification Γ.

A reaction R_i for which there is a tuple $(u, v) \in \theta_i$ such that either u or v still contains more than one element after algorithm *transform* has completed is unre-

Algorithm 2 transform($\Gamma, \theta_1, \ldots, \theta_k$)

```
repeat
    let $\Gamma' = \Gamma$, $\forall i.\, \theta'_i = \theta_i$
    for all $i \in \{1, \ldots, k\}$ do
        for all $(u, v) \in \theta'_i$ do
            if $u = \{\sigma_1\}$ and $v = \{\sigma_2\}$ then
                $\Gamma = \Gamma \oplus \{(\sigma_1, \sigma_2)\}$
            else if $|u| = 0$ or $|v| = 0$ then
                error
            else if $|u| + |v| > 2$ and $\exists x \in u, y \in v.\ x \equiv_\Gamma y$ then
                $\theta_i = \theta_i \setminus (u, v) \cup \{(u \setminus \{x\}, v \setminus \{y\}), (x, y)\}$
            else if $|u| = 1$ and $v = \{y_1, \ldots, y_n\}$ with $n > 1$
            and $\forall x \in u, y \in v.\ x \not\equiv_\Gamma y$ then
                let $x_1, \ldots, x_n$ be fresh names
                $\forall h \in \{1, \ldots, k\}.\ \theta_h = \theta_h[\{x_1, \ldots, x_n\}/u]$
                $\Gamma = \Gamma \oplus \{(x_j, y_j) \mid j \in \{1, \ldots, n\}\}$
            else if $u = \{x_1, \ldots, x_n\}$ with $n > 1$ and $|v| = 1$
            and $\forall x \in u, y \in v.\ x \not\equiv_\Gamma y$ then
                let $y_1, \ldots, y_n$ be fresh names
                $\forall h \in \{1, \ldots, k\}.\ \theta_h = \theta_h[\{y_1, \ldots, y_n\}/v]$
                $\Gamma = \Gamma \oplus \{(x_j, y_j) \mid j \in \{1, \ldots, n\}\}$
            end if
        end for
    end for
until $\Gamma = \Gamma'$ and $\forall i.\ \theta_i = \theta'_i$
return $(\Gamma, \theta_1, \ldots, \theta_k)$
```

solved, and user help is necessary. Therefore, the algorithm assumes that the user resolves the reaction by rewriting it in normal form $x_1, \ldots, x_n \to y_1, \ldots, y_n$. Note that the user is allowed to introduce new symbols. User intervention is requested to resolve a single reaction, even if there are multiple unresolved reactions, then the procedure is performed again. The algorithm repeats the cycle until a fix point is reached, meaning that all components have been identified.

Algorithm *transform* iteratively refines the classification Γ and the reactions $\theta_1, \ldots, \theta_k$ until a fix point is reached. Each iteration involves examining each tuple $(u, v) \in \theta_i$, for each reaction $i \in \{1, \ldots, k\}$. The algorithm distinguishes among five different forms of the tuple (u, v):

- $u = \{\sigma_1\}$, $v = \{\sigma_2\}$, i.e. u and v are singletons. In this case σ_1, σ_2 denote elements of the same component. Formally, this means that Γ can be updated by adding the equality $\sigma_1 \equiv \sigma_2$, denoted by tuple (σ_1, σ_2). (Operator $\Gamma \oplus \Gamma'$ takes two relations Γ and Γ' and constructs the smallest equivalence relation induced by them, i.e. such that $\Gamma, \Gamma' \subseteq \Gamma \oplus \Gamma'$.)
- either $u = \emptyset$ or $v = \emptyset$. As we will see later, it is not possible that both u and v are empty. Therefore, if either one of the two sets are empty then there is an error

since this means that some components either appears or disappears in between the reaction. (This situation is not allowed by our approach.) In principle, this case could also be handled by user intervention. However, user intervention in this case should not be aimed at solving an ambiguity, but at correcting the initial pathway (e.g. by inserting a dummy symbol representing a disappeared species). Such a correction cannot be made automatically, since the reaction to be corrected could be not the one in which the algorithm encounters the error. For example, let $P = \{A, B \to C\,,\ C \to B\}$. The algorithm can encounter the error either on the first reaction or on the second one depending on which reaction it considers first during its execution.

- there exists a pair of elements $x \in u$, $y \in v$, and *at least another element in either u or v*, such that $x \equiv_\Gamma y$, i.e. such that x,y are in the same component. In this case, elements x, y are removed from tuple (u, v) and a new tuple (x, y) is added to θ_i, meaning that there is correspondence between x and y.
- $u = \{\sigma_1\}$, $v = \{y_1, \ldots, y_n\}$ with $n > 1$, and $\forall i.\ \sigma_1 \not\equiv_\Gamma y_i$, i.e. there is one reactant σ_1 and multiple products to be matched, where σ_1 occurs in a different component than all the other elements. The only way to match σ_1 with $y_1, \ldots, y_n$ is to assume that σ_1 actually is a *complex* composed of different elements. That is, we split σ_1 into the new symbols $x_1, \ldots, x_n$, by replacing σ_1 with the set $\{x_1, \ldots, x_n\}$ in all the reactions θ_h, for all $h \in \{1, \ldots, k\}$. (The replacement operation is denoted $\theta_h[\{x_1, \ldots, x_n\}/u]$.) Moreover, each pair of elements x_j, y_j is declared to be occurring in the same component, by adding to Γ the equalities $x_j \equiv y_j$, for all $j \in \{1, \ldots, n\}$.
- $u = \{x_1, \ldots, x_n\}$ with $n > 1$, $v = \{\sigma_2\}$, and $\forall i.\ \sigma_2 \not\equiv_\Gamma x_i$. This case is the symmetrical to the previous one, and is handled analogously to it by replacing σ_2 with new symbols $y_1, \ldots, y_n$, and equating $x_j \equiv y_j$, for all $j \in \{1, \ldots, n\}$.

It is important to notice that the algorithm skips any tuple (u, v) such that $|u| > 1$, $|v| > 1$, and for which no element from u can be matched with any element from v, namely they occur in different components. Notice that this case is actually the case for which user input is requested by algorithm *findComponents*, should any tuple of this kind still be present in the result returned by algorithm *transform*.

The following theorem shows that the *transform* algorithm terminates. As a consequence, the termination of algorithm *findComponents* (in case of ambiguous reaction) clearly depends upon user behavior. For example, if the user could introduce new symbols for an infinite number of times, then the algorithm may never terminate. On the other hand, this is just a theoretical glitch, which does not impede the practical usage of the algorithm, since the automatic phase implemented by algorithm *transform* always terminates.

Theorem 3.1 *Algorithm transform always terminates.*

Proof (Sketch) The outer loop terminates as soon as both the classification Γ and the sets $\{\theta_i\}_i$ do not change from the previous iteration. Therefore, in order for the algorithm to never terminate, it is necessary that either Γ or one of the θ_i is changed at each iteration of the outer loop. Notice that Γ is ever modified by adding new equalities to it, therefore, since $\Gamma \subseteq \mathit{species}(P) \times \mathit{species}(P)$ (unless new species are introduced), there does not exists an infinite chain of updates in which Γ is always different from the previous one. As regards the reactions, either the number of pairs in any θ_i increases, or a symbol in a pair is replaced with a set of new symbols. In the former case we have that the number of pairs increases since one already existing pair is split into two, and this can be done a finite number of times. In latter case the replacement may require new splits to be performed. However, it can be shown a sequence of splits and replacements eventually lead either to reactions in normal form or to a situation in which user intervention is needed. $\square$

We leave as future work to investigate the theoretical properties of our algorithms.

4 Applications

We apply our component identification algorithm to a well-established computational model of the EGF signaling pathway. We consider the model of the MAP kinase cascade activated by surface and internalized EGF receptors, proposed by Schoeberl et al. in [21]. This model includes a detailed description of the reactions that involve active EGF receptors and several effectors named GAP, ShC, SOS, Grb2, RasGDP/GTP and Raf. Moreover, the model describes the activity of internalized receptors, namely receptors that are no longer located on the cell membrane, but on a vesicle obtained by endocytosis and floating in the cytoplasm. Such internalized receptors continue to interact with effectors and to contribute to the pathway functioning, but actually the pathway can be seen as composed by two almost identical branches: the first consisting of the reactions stimulated by receptors on the cell membrane, and the second consisting of reactions stimulated by internalized receptors.

A pictorial representation of the EGF pathway is shown in Figure 1, where the different biological entities involved in the pathway are depicted with different shapes and colors. A diagram representing all of the reactions of the pathway considered in the Schoeberl model is shown in Figure 2. In the figure, species are identified by a short name, but also by a number (in black) in the interval $[1-60]$. Arrows represent reactions, which are also associated with an identifier (in gray) in the interval $[v0-v101]$. Note that the two branches of the pathway are partially combined in the figure. In particular, the representation of most of the species is combined with the representation of its internalized counterpart. In such cases, the number in brackets denotes the number identifying the internalized species. The same holds for reactions: often an arrow denotes both a reaction stimulated by receptors in the cell membrane and the corresponding reaction stimulated by internalized receptors.

The set of reactions constituting the pathway can be directly reconstructed from the diagram in Figure 2. The only non-trivial aspect is related with the presence in the diagram of some reactions in which one reactant is actually acting as a catalyst.

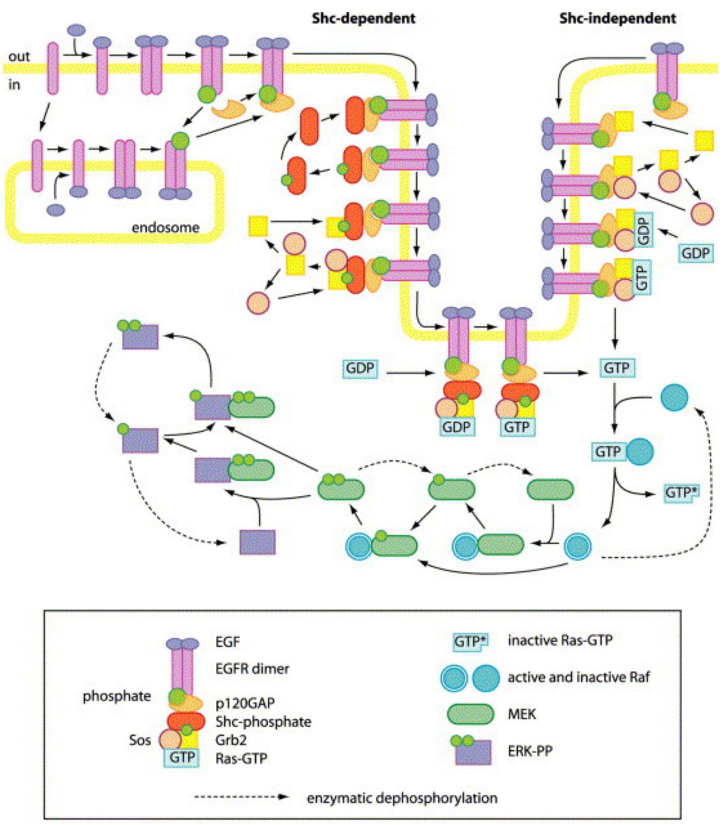

Fig. 1. Pictorial representation of the EGF pathway (from [8])

Component	Species	Component	Species	Component	Species
EGF	1	Grb2	20	ERK	5
EGFi	2	Sos	15	Phosphatase1	1
EGFR	32	RasGTP	15	Phosphatase2	1
GAP	23	Raf	5	Phosphatase3	1
Shc	16	MEK	5		

Table 1
Components identified by the algorithm

For instance, this happens in the case of the reactions involving Raf* and MEK, in which Raf* initially binds MEK and then releases it phosphorylated. We describe these two reactions in the diagram with the following single catalyzed reaction: MEK $\rightarrow$ MEK-P $\{$ Raf* $\}$.

Other species acting as catalysts are MEK-PP, Phosphatase1, Phosphatase2 and Phosphatase3. For the sake of simplicity we assumed also EGF and EGFi to act as catalysts. By applying the same transformation also to the reactions all of these species are involved in, we obtain a pathway constituted by 66 different species and 80 reactions.

Our component identification algorithm (implemented in Java) executes on the considered pathway without the need of human intervention. At the end of the

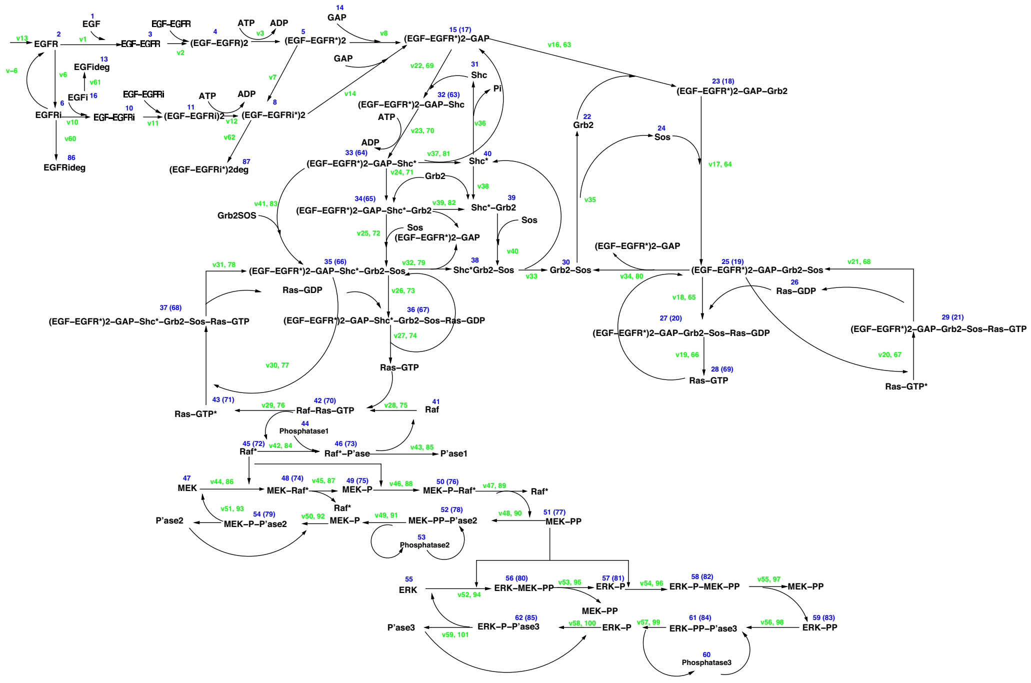

Fig. 2. Schematic representation of the EGF pathway (Schoeberl model, from [21])

execution the obtained normalized pathway is made up of 142 species (76 more than the original pathway) and 14 components are identified. We show in Table 1 the list of components (each given an intuitive name). The table contains also the number of species associated with each component.

Note that the components automatically identified by the algorithm essentially correspond to the biological entities depicted in Figure 1. The only differences with respect to Figure 1 are components Phosphatase1, 2 and 3 (that are ignored in the figure), the distinction of EGFi (internalized version of EGF) from EGF done by our algorithm and the absence of the phosphate entity, that actually we omitted from the model as well as ADP and ATP. The similarity between Table 1 and Figure 1 shows that our algorithm in this case has been able to identify as molecular components the real biological entities involved in the pathway. Components have been identified simply by observing their interactions.

To have an idea on how the pathway changes after normalization we show two diagrams obtained by using the modeling tool CellDesigner [15]. In order to exploit such a tool we needed to adopt SBML (http://www.sbml.org) as notation for pathway description. We used CellDesigner essentially to automatically obtain diagrammatic representations of SMBL descriptions (with species represented as boxes and reactions as arrows). The first diagram is shown in Figure 3 and represents the original pathway. The second diagram is shown in Figure 3 and represents the pathway in normal form. The two diagrams have similar shapes, and the second one (as expected) contains many more species. Although detailed information on the represented species cannot be read in the figures, it is rather clear that the majority

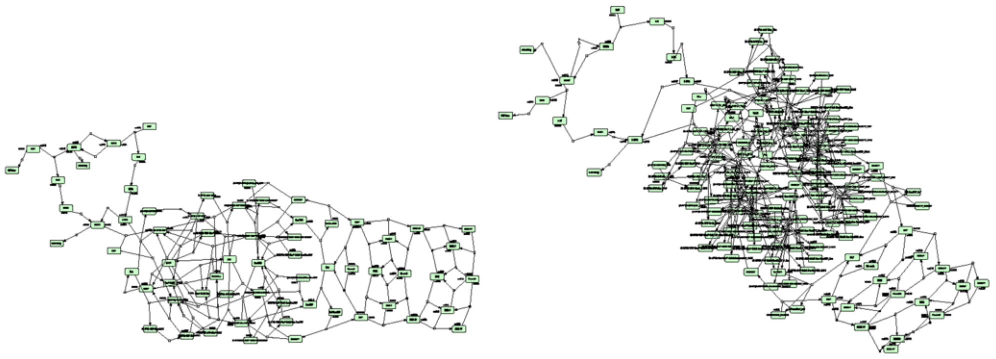
Fig. 3. CellDesigner representations of the EGFR pathway: original pathway (left) and normal form (right)

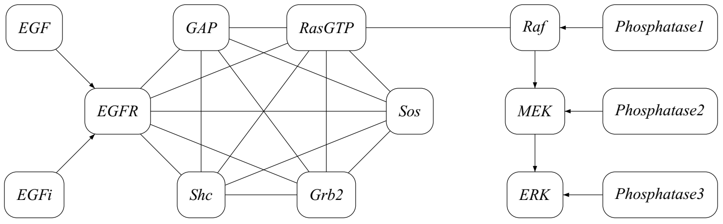

Fig. 4. Interaction graph of the EGF pathway

of the new species introduced in the pathway in normal form are located in the central part of the pathway. Indeed, such a part of the pathway consists of reactions performed by big complexes that are split into new species by our algorithm.

Once the molecular components are identified, it is possible to automatically generate a *component interaction graph*. We actually already did this in [14] where we used a normal-form version of the EGF pathway constructed by hand. A component interaction graph is a graph where nodes represent molecular components of a given pathway, and (possibly directed) edges represent the existence of direct interactions among two components. In other words, we have an undirected edge in the graph connecting nodes representing two components that are both involved in the same reaction as reactants and products. Moreover, we have a directed edge connecting nodes representing two components involved in the same reaction one (the source node) as catalyst and the other (the target node) as reactant and product. The component interaction graph is a very concise representation of the pathway that allows the role of the components and the structure of the pathway to be clarified.

In Figure 4 we can see the component interaction graph of the EGF pathway. Each node of the graph is labeled by the intuitive name of the component that we have chosen. Visually, we can do some simple observations on the component interaction graph. We can identify enzymes like Phosphatase1, Phosphatase2 and Phosphatase3. We can see the first part of the pathway corresponding to the EGF receptor and its interaction with effectors, and its connection to the MEK/ERK cascade through RasGTP and Raf.

The component interaction graph may suggest *subpathways* to be considered,

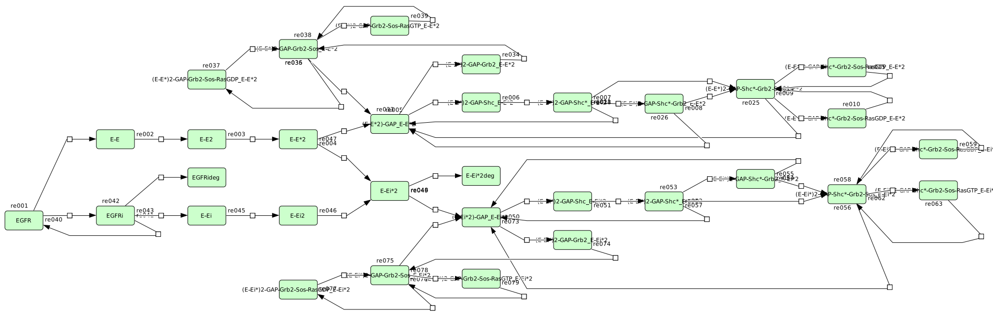

Fig. 5. CellDesigner representation of component EGFR of the EGFR pathway

consisting only of the species and the reactions involving a subset of the components. This could allow the analyst to focus on a particular part of the pathway by forgetting about irrelevant components, species and reactions. For example, one could choose to focus on the subpathway in which only components EGF, EGFi, EGFR, GAP, Shc, Grb2, Sos and RasGTP, or to focus on the MEK/ERK cascade by considering only the two corresponding components, or to focus on the "backbone" subpathway consisting of components EGF, EGFi, EGFR, RasGTP, Raf, MEK and ERK. It is worth noting that subpathways can be constructed automatically.

Another interesting syntactic elaboration consists in constructing the subpathways of the individual components. These subpathways make explicit, and consequently clearer, the sequence of state changes performed by each molecular component. In the example of the EGF pathway we have that four components consist of a single species (see Table 1). Indeed, these correspond to molecules that never change state during the pathway and their subpathways are actually empty. Component EGFi consists of two species since the EGFi molecule can be degraded, and the corresponding subpathway contains only one reaction (degradation of EGFi). All of the other components have a more complex individual dynamics, corresponding to a more complex subpathway. As examples we show the subpathways of components EGFR (Figure 5), RasGTP (Figure 6), Raf and MEK (both in Figure 7).

The subpathways of the individual model components are actually *finite state automata* in which we can assume transitions to be labeled by reactions identifiers. Consequently, we can collect all of these automata into a unique model of concurrent communicating automata (see e.g. [1]) in which synchronizations are driven by transition labels. As a result we obtain a complete automata-based representation of the pathway that we computed automatically by applying our component identification algorithm and the procedure for subpathway construction. Such representation can be used for formal verification of properties (e.g. model checking), for further automatic translations (e.g. into process algebras), and for other kinds of analyses.

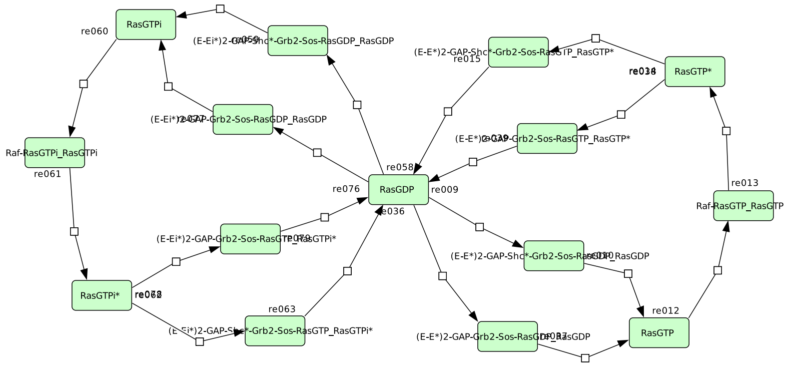
RasGTPi
re060
(E-Ei*)2-GAP-Shc*-Grb2-Sos-RasGDP_RasGDP
(E-E*)2-GAP-Shc*-Grb2-Sos-RasGTP_RasGTP*
re015
RasGTP*
(E-Ei*)2-GAP-Grb2-Sos-RasGDP_RasGDP
(E-E*)2-GAP-Grb2-Sos-RasGTP_RasGTP*
Raf-RasGTPi_RasGTPi
re061
re058
re076
RasGDP
re009
re036
re013
Raf-RasGTP_RasGTP
(E-Ei*)2-GAP-Grb2-Sos-RasGTP_RasGTPi*
(E-E*)2-GAP-Shc*-Grb2-Sos-RasGDP_RasGDP
RasGTPi*
re063
(E-Ei*)2-GAP-Shc*-Grb2-Sos-RasGTP_RasGTPi*
(E-E*)2-GAP-Grb2-Sos-RasGDP_RasGDP
re012
RasGTP

Fig. 6. CellDesigner representation of component RasGTP of the EGFR pathway

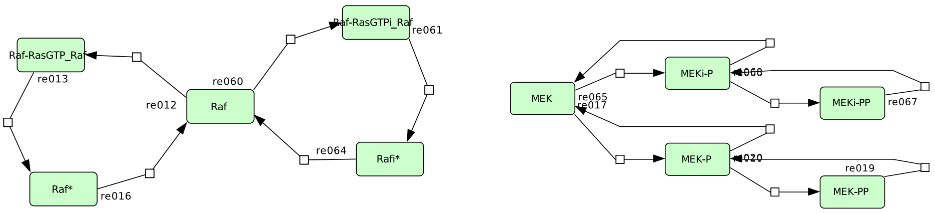

Fig. 7. CellDesigner representation of components Raf (left) and MEK (right) of the EGFR pathway

5 Conclusions and future work

We proposed an algorithm for the identification of molecular components involved in a given pathway. The pathway has to be specified simply as a set of chemical reactions. The algorithm infers molecular components from the interactions between species described by the reactions and gives as result a partition of the set of species in which each element of the partition contains species representing all of the possible states that a molecular component can reach. Moreover, during its execution the algorithm transforms the pathway into a “normal form” in which each reaction has as many products as reactants.

The algorithm is semi-automatic, since ambiguous reactions may require human intervention to be correctly interpreted. However, it seems that in practice human intervention is required quite rarely, as confirmed by some preliminary test we performed on a number of real pathways [18].

As regards future work we plan to develop a web application based on our component identification algorithm, that will accept pathways described by using the SBML language.

Furthermore, we plan to investigate more in deep some theoretical properties of the algorithm and, more in general, of the notion of molecular components. In particular, properties related with computation complexity, confluence of different sequences of human interventions and preprocessing of pathways could be studied.

Finally, we plan to adapt our approach to better work with quantitative aspects of pathways such as reaction kinetics and spatial distribution of molecules. In fact, the transformation of pathways into normal form used as-is in a quantitative context would not preserve the rate of occurrence of reactions. Such a rate depends on the quantities of reactants. Since the algorithm can change the description of reactants to bring reactions into normal form, rates of reactions cannot in general be preserved. Once quantitative aspects will be correctly dealt with, the translation of pathways into stochastic and spatial formalisms [4,5,9,11,20] will be made possible.

References

[1] Alur, R., S. Kannan and M. Yannakakis, *Communicating hierarchical state machines*, in: J. Wiedermann, P. Emde Boas and M. Nielsen, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science **1644**, Springer Berlin Heidelberg, 1999 pp. 169–178.

[2] Barbuti, R., G. Caravagna, A. Maggiolo-Schettini, P. Milazzo and G. Pardini, *The calculus of looping sequences*, in: *Formal Methods for Computational Systems Biology*, Springer, 2008 pp. 387–423.

[3] Barbuti, R., G. Caravagna, A. Maggiolo-Schettini, P. Milazzo and S. Tini, *Foundational aspects of multiscale modeling of biological systems with process algebras*, Theoretical Computer Science **431** (2012), pp. 96–116.

[4] Barbuti, R., A. Maggiolo-Schettini, P. Milazzo and G. Pardini, *Spatial calculus of looping sequences*, Theoretical Computer Science **412** (2011), pp. 5976–6001.

[5] Barbuti, R., A. Maggiolo-Schettini, P. Milazzo, G. Pardini and L. Tesei, *Spatial p systems*, Natural Computing **10** (2011), pp. 3–16.

[6] Barbuti, R., A. Maggiolo-Schettini, P. Milazzo and S. Tini, *Compositional semantics and behavioral equivalences for p systems*, Theoretical Computer Science **395** (2008), pp. 77–100.

[7] Barbuti, R., A. Maggiolo-Schettini, P. Milazzo and A. Troina, *Bisimulation congruences in the calculus of looping sequences*, in: *Theoretical Aspects of Computing-ICTAC 2006*, Springer, 2006 pp. 93–107.

[8] Breitling, R. and D. Hoeller, *Current challenges in quantitative modeling of epidermal growth factor signaling*, FEBS Letters **579** (2005), pp. 6289 – 6294.

[9] Calder, M., S. Gilmore and J. Hillston, *Modelling the influence of rkip on the erk signalling pathway using the stochastic process algebra pepa*, in: *Transactions on computational systems biology VII*, Springer, 2006 pp. 1–23.

[10] Calzone, L., F. Fages and S. Soliman, *Biocham: an environment for modeling biological systems and formalizing experimental knowledge*, Bioinformatics **22** (2006), pp. 1805–1807.

[11] Cardelli, L. and P. Gardner, *Processes in space*, Theoretical Computer Science **431** (2012), pp. 40–55.

[12] Ciocchetta, F. and J. Hillston, *Bio-pepa: A framework for the modelling and analysis of biological systems*, Theoretical Computer Science **410** (2009), pp. 3065–3084.

[13] Drábik, P., A. Maggiolo-Schettini and P. Milazzo, *Modular Verification of Interactive Systems with an Application to Biology*, Scientific Annals of Computer Science **21** (2011), pp. 39–72.

[14] Drábik, P., A. Maggiolo-Schettini and P. Milazzo, *Towards modular verification of pathways: fairness and assumptions*, in: G. Ciobanu, editor, *Proceedings 6th Workshop on Membrane Computing and Biologically Inspired Process Calculi*, EPTCS **100**, 2012, pp. 63–81.

[15] Funahashi, A., Y. Matsuoka, A. Jouraku, H. Kitano and N. Kikuchi, *CellDesigner: a modeling tool for biochemical networks*, in: *WSC '06: Proceedings of the 38th conference on Winter simulation* (2006), pp. 1707–1712.

[16] Gay, S., S. Soliman and F. Fages, *A graphical method for reducing and relating models in systems biology*, Bioinformatics **26** (2010), pp. i575–i581.

[17] Heath, J., M. Kwiatkowska, G. Norman, D. Parker and O. Tymchyshyn, *Probabilistic model checking of complex biological pathways*, Theoretical Computer Science **391** (2008), pp. 239–257.

[18] Maggiolo-Schettini, A., P. Milazzo and G. Pardini, *Application of a semi-automatic algorithm for identification of molecular components in SBML models*, Submitted.

[19] Pardini, G., R. Barbuti, A. Maggiolo-Schettini, P. Milazzo and S. Tini, *A compositional semantics of reaction systems with restriction*, in: *The Nature of Computation. Logic, Algorithms, Applications*, Springer, 2013 pp. 330–339.

[20] Phillips, A. and L. Cardelli, *Efficient, correct simulation of biological processes in the stochastic pi-calculus*, in: *Computational Methods in Systems Biology*, Springer, 2007, pp. 184–199.

[21] Schoeberl, B., C. Eichler-Jonsson, E. D. Gilles and G. Müller, *Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors*, Nature biotechnology **20** (2002), pp. 370–375.