Abstract interpretation of cellular signalling networks

Vincent Danos[1,4], Jérôme Feret[2,3], Walter Fontana[1,2], and Jean Krivine[5]

[1] Plectix Biosystems
[2] Harvard Medical School
[3] École Normale Supérieure **
[4] CNRS, Université Paris Diderot
[5] École Polytechnique

Abstract. *Cellular signalling pathways, where proteins can form complexes and undergo a large array of post translational modifications are highly combinatorial systems sending and receiving extra-cellular signals and triggering appropriate responses. Process-centric languages seem apt to their representation and simulation [1–3]. Rule-centric languages such as κ [4–8] and BNG [9, 10] bring in additional ease of expression.*
We propose in this paper a method to enumerate a superset of the reachable complexes that a κ rule set can generate. This is done via the construction of a finite abstract interpretation. We find a simple criterion for this superset to be the exact set of reachable complexes, namely that the superset is closed under swap, an operation whereby pairs of edges of the same type can permute their ends.
We also show that a simple syntactic restriction on rules is sufficient to ensure the generation of a swap-closed set of complexes. We conclude by showing that a substantial rule set (presented in Ref. [4]) modelling the EGF receptor pathway verifies that syntactic condition (up to suitable transformations), and therefore despite its apparent complexity has a rather simple set of reachables.

1 Introduction

Biological signalling pathways are large, natural, quantitative concurrent systems in charge of sending and receiving extra-cellular signals and triggering appropriate responses in the cell —eg differentiation, migration, or growth. They involve multiple proteins, from membrane bound receptors to adapters and relays to transcription factors. As thorough a description as possible of these pathways is key to their understanding and control. Such a task is difficult for a variety of reasons, one being of a purely representational nature. Those networks are highly combinatorial, meaning that their agents can assemble and be modified in a huge number of ways —about 10^{19} unique complexes for the EGF receptor pathway model we consider later. Usual representations based on reactions between structureless entities must inevitably sample down this combinatorial

** Abstraction Project (INRIA, CNRS, and École Normale Supérieure)

complexity, and obtain models which stand in no clear relation to biological facts and are hard to keep abreast of new developments. Regev *et al.* have proposed using π-calculus [11] to avoid the combinatorial explosion besetting differential equations [1, 2]. Other process-based languages have been proposed [3, 12–14]. The rule-based languages κ [4–7] and BNG [9, 10] bring additional ease in the building and modification of models [8].

The object of this paper is to explain and illustrate on a sizable example a method to explore the set of complexes that can be generated by a κ rule set, aka the system's *reachable complexes*. Although κ models can be run with no prior enumeration of reachable complexes [5], a convenient method for describing those can be used to:
- detect dead rules (which is useful when developing large models)
- coarsen rules (ie get rid of superfluous conditions while preserving the underlying qualitative transition system)
- refine rules (eg for kinetic reasons)
- determine whether a rule may activate another (which brings down the cost of stochastic simulations [5])
- generate the underlying ground system (or a truncated version thereof if too large), and equip it with a differential equation semantics for the purpose of fast calibration of a model on available data (not implemented yet).

The very combinatorial nature of signalling systems manifests itself in that computing reachables by transitive closure is unfeasible for any but the simplest networks. Our method works around this problem by defining a finite interpretation of rule sets. This finitisation is based on an approximation of complexes as sets of radius 1 neighbourhoods, which we call *views*, and a pair of adjoint maps to break down complexes into views, and recombine views into complexes. Thus one can generate efficiently a superset of the reachable views, and decide whether the corresponding set of complexes is infinite by detecting repeatable patterns (section 3). This begs the question when the reachable views recombine to form exactly the reachable complexes, not just a super-set. This happens when the set of reachables is closed under swap, an operation whereby pairs of edges of the same type can permute their ends (section 4). We call such sets *local*, and by extension say a model is local if its set of reachables is. The definition of locality for a model is not syntactical, since it is a condition on the set of associated reachables, but one can guarantee locality by placing syntactical restrictions on the model's rule set (section 5). Our EGF receptor network example satisfies that syntactical condition —up to some reachables-preserving transformations— and is local despite its apparent complexity (section 6).

This touches on an interesting and more speculative question. Situations can be expressed in κ which have little to do with biological signalling (eg it is straightforward to represent Turing machines). One would like to think, as the EGF example model indicates, that we have delineated a fragment of κ —that of local rule sets— where natural signalling pathways predominantly fall. What that may mean biologically is briefly discussed in the conclusion together with leads for future work.

$$
\begin{array}{lll}
E ::= \emptyset \mid a, E & \text{(expression)} & s ::= n_\iota^\lambda \quad \text{(site)} \\
a ::= N(\sigma) & \text{(agent)} & n ::= x \in \mathcal{S} \quad \text{(site name)} \\
N ::= A \in \mathcal{A} & \text{(agent name)} & \iota ::= \epsilon \mid m \in \mathbb{V} \quad \text{(internal state)} \\
\sigma ::= \emptyset \mid s, \sigma & \text{(interface)} & \lambda ::= \epsilon \mid i \in \mathbb{N} \quad \text{(binding state)}
\end{array}
$$

Fig. 1. *Syntax.*

$$
E, A(\sigma, s, s', \sigma'), E' \equiv E, A(\sigma, s', s, \sigma'), E' \qquad \frac{i, j \in \mathbb{N} \text{ and } i \text{ does not occur in } E}{E[i/j] \equiv E}
$$
$$
E, a, a', E' \equiv E, a', a, E'
$$

Fig. 2. *Structural equivalence.*

2 κ

We first briefly present a simplified core κ using a process-like notation which facilitates the reachability analysis of the next section. This is in contrast with the equivalent graph-theoretical presentation chosen in Ref. [5] for the definition of the quantitative (stochastic) semantics.

We suppose given a finite set of agent names $\mathcal{A}$, representing different kinds of proteins; a finite set of sites $\mathcal{S}$, corresponding to protein domains and modifiable residues; a finite set of values $\mathbb{V}$, representing the modified states. The syntax of agents and expressions is given in Fig. 1.

An *interface* is a sequence of sites with internal and binding states; specifically one writes x_ι^λ for a site x with internal state ι, and binding state λ. If the binding state is ϵ, the site is *free*; otherwise it is *bound*. On the other hand, if the internal state is ϵ, this means the internal state is left unspecified. In the concrete notation both ϵs are omitted.

An *agent* is given by a name in $\mathcal{A}$ and an interface.

A well-formed *expression* is a sequence of agents such that:
- a site name can occur only once in an interface,
- a binding state occurs exactly twice if it does at all.

We suppose hereafter all expressions to be well-formed.

Sites sharing a same binding state are said to be bound.

The structural equivalence $\equiv$ defined as the smallest binary equivalence between expressions that satisfies the rules given in Fig. 2 stipulates that: neither the order of sites in interfaces, nor the order of agents in expressions matters, and that bindings states can be injectively renamed.

Equivalence classes of $\equiv$ are called *solutions* and one writes $[E]$ for the class of expression E. One says a solution $[E]$ is *reducible* whenever $E \equiv E', E''$ for some non empty expressions E', E''. A *complex* is an irreducible solution.

Complexes and solutions can equivalently be presented graphically. Fig. 3 shows an example for the following expression:

$$
\begin{array}{l}
\mathit{EGF}(r^1), \mathit{EGF}(r^2), \mathit{EGFR}(l^1, r^3, \mathit{Y1048}_p^4, \mathit{Y1148}_p), \mathit{EGFR}(l^2, r^3, \mathit{Y1048}_u, \mathit{Y1148}_p^5), \\
\mathit{GRB2}(\mathit{SH2}^4, \mathit{SH3}^6), \mathit{SOS}(a^6), \mathit{SHC}(\mathit{PTB}^5, \mathit{Y317}_p^7), \mathit{GRB2}(\mathit{SH2}^7, \mathit{SH3})
\end{array}
$$

A *rule* is a pair of expressions E_l, E_r.

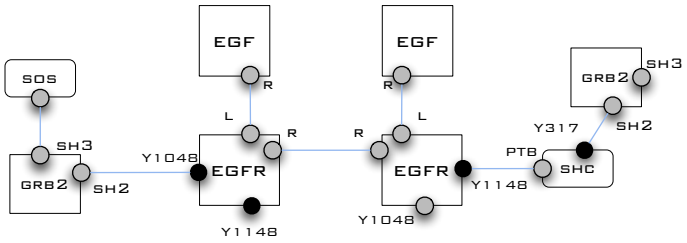

Fig. 3. *A complex from the EGF receptor pathway: sites correspond to protein domains SH2, SH3, ... and modifiable amino-acid residues Y317, Y1048, ... edges correspond to bindings, solid black stands for phosphorylated tyrosine residues.*

$$
\begin{array}{cc}
n_\iota^\lambda \models n_\iota^\lambda & n_\iota^\lambda[n_{\iota_r}^{\lambda_r}] = n_{\iota_r}^{\lambda_r} \\
n_\iota^\lambda \models n^\lambda & n_\iota^\lambda[n^{\lambda_r}] = n_\iota^{\lambda_r} \\
\sigma \models \emptyset & \sigma[\emptyset] = \sigma \\
\dfrac{s \models s_l \quad \sigma \models \sigma_l}{s, \sigma \models s_l, \sigma_l} & s, \sigma[s_r, \sigma_r] = s[s_r], \sigma[\sigma_r] \\
\dfrac{\sigma \models \sigma_l}{N(\sigma) \models N(\sigma_l)} & N(\sigma)[N(\sigma_r)] = N(\sigma[\sigma_r]) \\
E \models \emptyset & E[\emptyset] = E \\
\dfrac{a \models a_l \quad E \models E_l}{a, E \models a_l, E_l} & (a, E)[a_r, E_r] = a[a_r], E[E_r]
\end{array}
$$

Fig. 4. *Definition of matching $\models$ (left), and replacement (right).*

The *left hand side* E_l describes the agents taking part in the event and various conditions on their internal and binding states for the event to actually happen. The *right hand side* E_r describes the rule's effect which is either:
- a *binding* (*unbinding*): E_r (E_l) is obtained by binding two free sites in E_l (E_r),
- or a *modification*: E_r is obtained by modifying some internal state in E_l.

Note that bindings and unbindings are symmetric, while modifications are self-symmetric.

In order to apply a rule E_l, E_r to a solution $[E]$, one uses structural equivalence (Fig. 2) to bring the participating agents at the beginning of the expression, with their sites in the same order as in E_l, and renames bindings to obtain an equivalent expression E' that matches E_l (Fig. 4, left). One then replaces E' with $E'[E_r]$ (Fig. 4, right).

This yields a *transition system* between solutions defined as $[E] \rightarrow_{E_l, E_r} [E[E_r]]$ whenever $E \models E_l$.

Note that sites not occurring in E_l are not constrained in any way, and that matching only uses structural equivalence on E, not E_l.

Our implementation also allows rules for agent creation and deletion. The methods and results presented here can be readily extended to this richer setting.

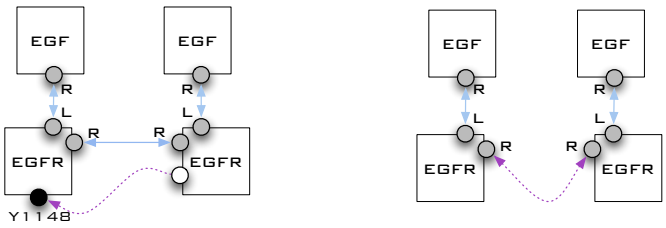

Fig. 5. *Receptor 'cross-phosphorylation' (left): site Y1148 is modified (phosphorylation induced by the receptor kinase domain represented as a solid white circle, and inducing a state change represented as a solid black circle); and receptor 'dimerisation' (right): the dotted edge represent binding, both ends of the link are modified.*

Here is an example of a modification, and a binding (see also Fig. 5):

$$EGF(r^1), EGFR(l^1, r^2, Y1148_u), EGFR(r^2, l^3), EGF(r^3) \rightarrow EGF(r^1), EGFR(l^1, r^2, Y1148_p), EGFR(r^2, l^3), EGF(r^3)$$
$$EGF(r^1), EGFR(l^1, r), EGFR(r, l^2), EGF(r^2) \rightarrow EGF(r^1), EGFR(l^1, r^3), EGFR(r^3, l^2), EGF(r^2)$$

3 Reachability

Let Σ denotes the set of all solutions, and Γ be the set of all complexes. Given R a set of rules and S_0 an initial solution, the set of reachable solutions is given as the least fixpoint in $\wp(\Sigma)$ greater than S_0, written $lfp_{S_0}\textsc{post}$, of the map $\textsc{post}(X) := X \cup \{S' \mid \exists S \in X,\ \exists r \in R,\ S \rightarrow_r S'\}$.

Write $[F] \in [E]$ if there is an expression E' such that $E \equiv F, E'$.

Define the maps $\alpha_c : \wp(\Sigma) \rightarrow \wp(\Gamma)$, $\gamma_c : \wp(\Gamma) \rightarrow \wp(\Sigma)$ as:

$$\alpha_c(X) := \{c \in \Gamma \mid \exists S \in X : c \in S\}$$
$$\gamma_c(Y) := \{S \in \Sigma \mid c \in S \Rightarrow c \in Y\}$$

The pair α_c, γ_c form a Galois connection and the set Γ^* of reachable complexes is $\alpha_c(lfp_{S_0}\textsc{post})$. The most precise counterpart to $\textsc{post}$, $\textsc{post}_c := \alpha_c\,\textsc{post}\,\gamma_c$ can be written $\textsc{post}_c(X) = X \cup \{c \in \Gamma \mid \exists [c_1], \ldots, [c_m] \in X\, \exists r \in R\, \exists S \in \Sigma : [c_1, \ldots, c_m] \rightarrow_r S \wedge c \in S\}$. Clearly $\Gamma^* \subseteq lfp_{\alpha_c(S_0)}\textsc{post}_c$.

This abstraction is not always exact since it does not take into account the number of occurences of a complex in a solution. In practice rules are rarely asking for many occurrences of a same agent or complex, and it is safe to consider that each kind of agent occurs an unbounded number of time in S_0.

One thing that does matter in the application is that Γ^* may be large (even infinite in case of polymerisation) and this is why we set up now a finite approximation of this set. The idea is to only retain from a solution the information which is local to the agents and which we call agents *views* (as in Ref. [15]). Specifically, we replace each binding state in an expression with its associated

$$
\begin{array}{llll}
v & ::= N(\sigma) & \text{(View)} & n ::= x \in \mathcal{S} \quad \text{(site name)} \\
N & ::= A \in \mathcal{A} & \text{(Agent name)} & \iota ::= \epsilon \mid m \in \mathbb{V} \quad \text{(Internal state)} \\
\sigma & ::= \emptyset \mid n_\iota^\lambda, \sigma & \text{(Interface)} & \lambda ::= \epsilon \mid n.N \quad \text{(Binding state)}
\end{array}
$$

Fig. 6. Syntax for views.

$$A(\sigma, s, s', \sigma') \equiv A(\sigma, s', s, \sigma')$$

Fig. 7. Structural congruence.

typed link, ie the site and agent names of the opposite end of the link, and call β the obtained transformation.

An example is:

$$
\begin{array}{l}
\beta(EGF(r^1), EGFR(l^1, r^2), EGFR(r^2, l^3), EGF(r^3)) = \\
EGF(r^{l.EGFR}), EGFR(l^{r.EGF}, r^{r.EGFR}), EGFR(l^{r.EGF}, r^{r.EGFR}), EGF(r^{l.EGFR})
\end{array}
$$

The syntax of views is given in Fig. 6, and the structural congruence which allows to reorder sites in a view is given in Fig. 7.

Operations on solutions transfer naturally to sequences of views. In particular one can define an abstract transition step between sequences of views (Fig. 8) that tests some conditions over the view relation $\models^\sharp$, and either changes the internal state of a site, or adds/removes the appropriate typed links in the binding state of two modified views.

Fig. 9 shows the graphical representation (repetitions are omitted) of the phosphorylation and dimerisation abstract rules (Fig. 5).

Thus one may now define the abstraction that collects the set of views that can be built during a computation sequence.

Define Δ to be the set of views and the map $\alpha : \wp(\Gamma) \to \wp(\Delta)$ as $\alpha(X) := \{[v_i] \mid \exists [c] \in X,\ \beta(c) = v_1, \ldots, v_n\}$. By construction α is a $\cup$-complete morphism of complete lattices, and has therefore an adjoint *concretization* $\gamma : \wp(\Delta) \to$

$$
\begin{array}{cc}
n_\iota^\lambda \models^\sharp n_\iota^\lambda & n_\iota^\lambda[n_{\iota_r}^{\lambda_r}]^\sharp = n_{\iota_r}^{\lambda^r} \\
n_\iota^\lambda \models^\sharp n^\lambda & n_\iota^\lambda[n^{\lambda_r}]^\sharp = n_\iota^{\lambda_r} \\
\sigma \models^\sharp \emptyset & \sigma[\emptyset]^\sharp = \sigma \\
\dfrac{s \models^\sharp s_l \quad \sigma \models^\sharp \sigma_l}{s, \sigma \models^\sharp s_l, \sigma_l} & s, \sigma[s_r, \sigma_r]^\sharp = s[s_r]^\sharp, \sigma[\sigma_r]^\sharp \\
\dfrac{\sigma \models^\sharp \sigma_l}{N(\sigma) \models^\sharp N(\sigma_l)} & N(\sigma)[N(\sigma_r)]^\sharp = N(\sigma[\sigma_r]^\sharp)
\end{array}
$$

$$
\frac{r = E_l \to E_r \quad \beta(E_l) = v_l^1, \ldots, v_l^n \quad \beta(E_r) = v_r^1, \ldots, v_r^n \quad v^i \models^\sharp v_l^i}{[v^1], \ldots, [v^n] \to_r^\sharp [v^1[v_r^1]^\sharp], \ldots [v^n[v_r^n]^\sharp]}
$$

Fig. 8. Abstract semantics.

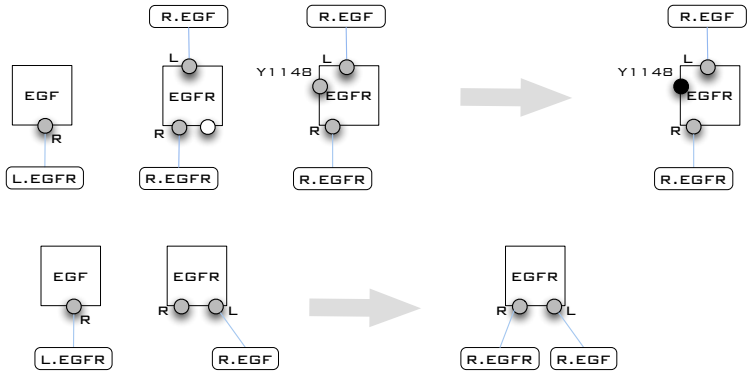

Fig. 9. *The partial views associated to the 'cross-phosphorylation' and 'dimerisation' patterns (multiplicities are not shown); arrows represent the result of the rule action on views (only modified views are shown).*

$\wp(\Gamma)$, defined as $\gamma(Z) = \cup\{X \in \wp(\Gamma) \mid \alpha(X) \subseteq Z\}$, which maps a set of views to the set of complexes that can be built from these, and verifies $\alpha(X) \subseteq Z$ iff $X \subseteq \gamma(Z)$. It is easy to see that α, γ are monotonic, $\alpha\gamma$ is a lower closure operator (anti-extensive, monotonic, and idempotent), and $\gamma\alpha$ an upper closure operator (extensive, monotonic, and idempotent) [16].

Let us consider a couple of examples of upper and lower closure:

$$\gamma\alpha(\{[A(a^1, b^1)]\}) = \{[A(a^n, b^1), \ldots, A(a^{n-1}, b^n)];\ n \in \mathbb{N}\}$$
$$\alpha\gamma(\{[A(a^{b.A}, b^{a.A})], [B(a^{b.A}, b^{a.A})]\}) = \{[A(a^{b.A}, b^{a.A})]\}$$

In the first example the upper operator constructs rings of all lengths; in the second one the typed link $B(b^{a.A})$ has no corresponding *dual* typed link $A(a^{b.B})$ in the view set, so its view cannot be combined with an other one.[6]

Define $\textsc{post}_v(Z) := Z \cup \{u_i \in \Delta \mid \exists v_1, \ldots, v_n \in Z\, \exists r \in R : v_1, \ldots, v_n \rightarrow^\sharp_r u_1, \ldots, u_n\}$. This map is a $\cup$-complete endomorphism and it satisfies $\textsc{post}_c\gamma \leq \gamma\,\textsc{post}_v$ for the pointwise ordering. As a consequence, $\mathit{lfp}_{\alpha(\Gamma_0)}\textsc{post}_v$ the least fixpoint of $\textsc{post}_v$ containing $\alpha(\Gamma_0)$ exists and we can state the soundness of our abstraction as follows: $\mathit{lfp}_{\Gamma_0}\textsc{post}_c \subseteq \gamma(\mathit{lfp}_{\alpha(\Gamma_0)}\textsc{post}_v)$. Thus the views generated by the abstract system reconstruct, via γ, a superset of the generated complexes. Note that while $\mathit{lfp}_{\alpha(\Gamma_0)}\textsc{post}_v$ is certainly finite, its image under γ may not be.

One may wonder how efficient that finite computation is, and how precise. Regarding efficiency, we use decision diagrams [17] to manipulate view sets. To avoid an exponential blow up (the number of views of an agent is exponential in its number of sites), we use 'packing' techniques [18], splitting the interface of agents into smaller subinterfaces, and then considering only relations between

[6] A trickier example is $\alpha\gamma(\{[A(a, b^{a.A})], [A(a^{b.A}, b^{a.A})]\}) = \{[A(a^{b.A}, b^{a.A})]\}$, since a finite chain with a free a must have a free b at the other end.

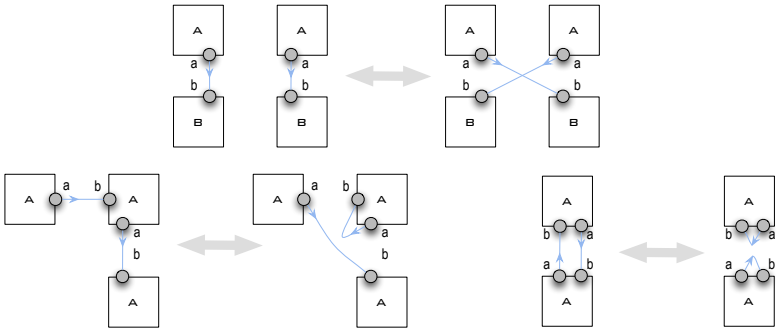

Fig. 10. *The main swap involution (top row); if the names A and B are the same, some agents can be identified (bottom row); orientations are shown for clarity.*

states of sites belonging to a common subset. Our syntactic analysis ensures that the result is unchanged. The next section answers accuracy concerns.

4 Local sets

Say $X \subseteq \Gamma$ is *local* if $X \in Im(\gamma)$, or equivalently $\gamma\alpha(X) = X$. We prove first that for such local sets, the finite interpretation is exact.

Theorem 1. *Consider $Inv \in \wp(\Gamma)$, a set of complexes such that: $\Gamma_0 \subseteq Inv$, $\textsc{post}_c(Inv) \subseteq Inv$, and $Inv = \gamma\alpha(Inv)$, then $\gamma(lfp_{\alpha(\Gamma_0)}\textsc{post}_v) \subseteq Inv$.*

The map $\textsc{post}_v$ is not the most precise counterpart of $\textsc{post}_c$, because the relation $\models^\sharp$ does not require views to be embeddable in complexes. However we shall see below that $\textsc{post}_v\alpha = \alpha\textsc{post}_c\gamma\alpha$ which means that $\textsc{post}_v$ is the most precise counterpart of $\textsc{post}_c$ when applied to abstract elements that are closed with respect to $\alpha\gamma$. Assuming this for the moment we can prove Th. 1.

Proof. We first prove $\textsc{post}_v^n\alpha(\Gamma_0) \in \alpha(\wp(\Gamma))$ and $\textsc{post}_v^n\alpha(\Gamma_0) \subseteq \alpha(Inv)$:
- $\alpha(\Gamma_0) \in \alpha(\wp(\Gamma))$, and $\Gamma_0 \subseteq Inv$, so $\alpha(\Gamma_0) \subseteq \alpha(Inv)$;
- If $Z \in \alpha(\wp(\Gamma))$ and $Z \subseteq \alpha(Inv)$, then $Z = \alpha\gamma(Z)$, so $\textsc{post}_v(Z) = \textsc{post}_v\alpha\gamma(Z)$, and since $\textsc{post}_v\alpha = \alpha\textsc{post}_c\gamma\alpha$, $\textsc{post}_v(Z) = \alpha\textsc{post}_c\gamma(Z) \subseteq \alpha\textsc{post}_c\gamma\alpha(Inv) = \alpha\textsc{post}_c(Inv) \subseteq \alpha(Inv)$. Because $\wp(\Delta)$ is finite, $lfp_{\alpha(\Gamma_0)}\textsc{post}_v = \textsc{post}_v^n\alpha(\Gamma_0)$ for some n, so $lfp_{\alpha(\Gamma_0)}\textsc{post}_v \subseteq \alpha(Inv)$, hence $\gamma(lfp_{\alpha(\Gamma_0)}\textsc{post}_v) \subseteq \gamma\alpha(Inv) = Inv$. □

By setting $Inv = lfp_{\Gamma_0}\textsc{post}_c$, one gets the immediate corollary:

Corollary 1. *If $lfp_{\Gamma_0}\textsc{post}_c \in \gamma(\wp(\Gamma))$, then $lfp_{\Gamma_0}\textsc{post}_c = \gamma(lfp_{\alpha(\Gamma_0)}\textsc{post}_v)$.*

In words, if $lfp_{\Gamma_0}\textsc{post}_c$ is local, the finite interpretation is exact. Likewise, taking Inv to be the local closure of $lfp_{\Gamma_0}\textsc{post}_c$, one obtains the slightly more general result that $\gamma(lfp_{\alpha(\Gamma_0)}\textsc{post}_v)$ is the smallest local set containing $lfp_{\Gamma_0}\textsc{post}_c$, supposing that closure is itself closed under $\textsc{post}_c$ (it does not have to be).

We proceed now to the characterisation of local sets.

We call a *swap* any of the three transformations over solutions given Fig. 10. Note that swapped links have to be of the same A, a, B, b type.

Theorem 2. *$X \in$ is local iff the set of solutions over X is closed under swaps.*

Clearly views are invariant under swaps, which gives the left to right implication. To prove the other implication we introduce an 'assembly grammar' to describe $\gamma(Z)$ for $Z \subseteq \Delta$. This grammar also allows the enumeration (and the counting) of the elements of $\gamma(Z)$. It will also help in proving that $\text{POST}_v\alpha = \alpha\text{POST}_c\gamma\alpha$ by taking $Z \in Im(\alpha)$.

The assembly grammar $\Rightarrow_Z$ where E, E' stand for hybrid expressions (ie expressions mixing ordinary binding states and typed links), σ, σ' for hybrid interfaces, and x is a fresh binding, is given as:
- $E \Rightarrow_Z E'$ if $E \equiv E'$
- $\Rightarrow_Z v$ for $v \in Z$
- $A(a_\iota^{b.B}, \sigma), E \Rightarrow_Z A(a_\iota^x, \sigma), B(b_{\iota'}^x, \sigma'), E$ for $B(b^{a.A}, \sigma') \in Z$
- $A(a_\iota^{b.B}, \sigma), B(b_{\iota'}^{a.A}, \sigma'), E \Rightarrow_Z A(a_\iota^x, \sigma), B(b_{\iota'}^x, \sigma'), E$
- $A(a_\iota^{b.A}, b_{\iota'}^{a.A}, \sigma), E \Rightarrow_Z A(a_\iota^x, b_{\iota'}^x, \sigma), E$

The third clause states that a typed link in a view can be connected to any view taken from Z showing the dual typed link. Similarly, the last two clauses show how dual typed links may be connected to form a link.

We write $\Rightarrow_Z^*$ for the transitive closure of $\Rightarrow_Z$.

Say a hybrid expression c embeds in a complex $[c^*]$ if c is the prefix of a hybrid expression E' obtained from an expression $E \equiv c^*$ by replacing some bindings with their corresponding typed links.

Clearly $\Rightarrow_Z^* c$ implies that c is connected, and $\gamma(Z)$ is the set of all classes $[c]$ such that $\Rightarrow_Z^* c$ and c has no typed links.

Proof (Th. 2, continued). We want $\gamma\alpha(X) \subseteq X$, supposing X is closed under swap. It is enough to prove that whenever $\Rightarrow_{\alpha(X)}^* c$, c embeds in some $[c^*] \in X$. Indeed if $c \in \gamma\alpha(X)$ embeds in some $[c^*] \in X$, then $c \equiv c^*$, since c has no typed links. We prove this by induction on $\Rightarrow_{\alpha(X)}$:
- The base case is by definition of α.
- Suppose c is obtained from c_1 by replacing a typed link $b.B$ with a binding to a view v of type B; by induction we have $[c_1^*]$, and $[v^*] \in X$ containing respectively c_1 and v; we can assume that expressions c_1 and v do not share bindings; therein A, a and B, b must be connected to say B', b, and A', a. One can therefore swap the bindings in the expression c_1^*, v^*, and connect A, a and B, b; since c_1 is connected, it is contained in the post-swap connected component of A which is in X (because X is closed under swap), and so is c.
- Suppose c is obtained from c_1 by fusing two typed links $b.B$ and $a.A$ in two distinct agents; by induction there is $[c_1^*]$ which embeds c_1, if A, a, B, b are connected in $[c_1^*]$ we are done, else consider B', and A' as above (those may be the same agent). Again one can swap the bindings in $[c_1^*]$, and connect A, a and B, b, and their common component after the swap contains c and is in X.
- Suppose c is obtained from c_1 by fusing two typed links $b.B$ and $a.A$ within the same agents; by induction there is $[c_1^*]$ which embeds c_1, if A, a is connected

to A, b we are done, else A, a, A, b are connected in $[c_1^*]$ to say A', b, A'', a (which may belong to the same agent), so one can swap the bindings, and the resulting connected component of A contains c and is in X. □

Using the proof above for the local set $\gamma\alpha(X)$ obtains a stronger statement:

Theorem 3. *If $\Rightarrow^*_{\alpha(X)}$ c, then c embeds in some $c^* \in \gamma\alpha(X)$.*

One may use the grammar $\Rightarrow_Z$ to obtain either an enumeration or a counting of $\gamma(Z)$, when $Z \in Im(\alpha)$ (which is the one case we are interested in in the application). In general, $\Rightarrow_Z$ has either finitely many rewrite sequences or, by a simple combinatorial argument, there must be a sequence of derivations that form a hybrid expression with a path connecting two instances of a typed link $b.B$. By Theorem 3, any such sequence can be completed so as to produce a complex (with no typed links), so one can effectively decide whether $\gamma(Z)$ is infinite (and practically stop the enumeration on derivations showing a repeatable pattern). Note that one may also infer from the same theorem that $\textsc{post}_v(Z) = \alpha\textsc{post}_c\gamma(Z)$ when $Z \in Im(\alpha)$ (the equation does not hold in general).

5 Local rule sets

At this stage, we know that local sets can be exactly counted or enumerated via the abstraction if they are finite, and neatly described if they are not.

Say a rule set R *is local* if given any set of disconnected agents, R generates a local set of complexes.

Proposition 1. *A rule set R is local if the following holds:*
- (acyclicity) complexes in $\gamma\alpha(\Gamma^*)$ are acyclic;
- (local tests) rules only test the views of the agents they modify;
- (non interference) binding rules do not interfere, that is to say:
– whenever $A(a_{\iota_1}, \sigma_1), B(b_{\iota_2}, \sigma_2) \to A(a^1_{\iota_1}, \sigma_1), B(b^1_{\iota_2}, \sigma_2)$
– and $A(a_{\iota_3}, \sigma_3), B(b_{\iota_4}, \sigma_4) \to A(a^1_{\iota_3}, \sigma_3), B(b^1_{\iota_4}, \sigma_4)$
– then $A(a_{\iota_1}, \sigma_1), B(b_{\iota_4}, \sigma_4) \to A(a^1_{\iota_1}, \sigma_1), B(b^1_{\iota_4}, \sigma_4)$

Before we sketch the proof, let us comment on the acyclicity condition which is the only non syntactical one.

Define the *contact map* of a rule set R, written $\chi(R)$, and defined as: a graph with nodes the agent names used in R, with sites those occurring in R, and where sites are connected iff they are bound by some $r \in R$. Note that $\chi(R)$ is not a (graphical) solution, since sites can be connected more than once; rather it is a constraint on generated complexes.

Say a complex is *compatible* with $\chi(R)$ if it projects to it.

Fig. 11 shows the contact map of an early EGF model, and the complex shown Fig. 3 does project to it.

One can test whether a rule set R is acyclic by inspecting $\chi(R)$.

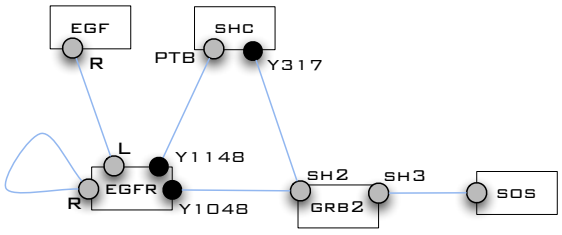

Fig. 11. *The early EGF receptor cascade model contact map; some sites can change their internal states (solid black).*

Proposition 2. *Given R, and Γ_0 compatible with $\chi(R)$, if Γ^* contains a cyclic complex, then there exists $s : \mathbb{Z}_{2n} \to \mathcal{S}$ such that for all $p \in \mathbb{Z}_n$: $s(2p), s(2p+1)$ belong to the same agent in $\chi(R)$, $s(2p) \neq s(2p+1)$, and $s(2p-1), s(2p)$ is an edge in $\chi(R)$.*

In Fig. 11, one sees that the contact map may be cyclic, but every cycle has to use twice the same site, so no complex in this model can be cyclic, provided one picks a compatible initial set of complexes.

Note also that the non-interference condition is only concerning binding operations and there is no comparable constraint on unbinding rules. The intuition is that both agents that want to connect have to do it on the sole basis of their views and are not allowed to communicate prior to context. This is reminiscent of synchronisation in CCS [19].

Proof (Prop. 1 sketch). Given $[E]$ a solution of complexes in Γ^*, and a swap s between links l_1, l_2 in $[E]$, one wants to prove that the obtained $[E^s]$ is still a solution of complexes in Γ^*. Call $[c_1]$, $[c_2]$ the connected components of l_1, l_2 in $[E]$; clearly, it is enough to prove it in the cases when $E \equiv c_1, c_2$ or when $E \equiv c_1 \equiv c_2$.

Suppose that $E \equiv c_1 \equiv c_2$, then l_1, l_2 are connected via a unique path in c_1 (by acyclicity). Suppose they have the same orientation along this path, then $[c_1^s]$ splits in two components, one of them, call it $[d_1]$, containing a cycle; on the other hand, $[d_1] \in \gamma\alpha(\Gamma^*)$, which contradicts acyclicity. So l_1, l_2 must have opposite orientations. Pick a copy of $[c_1]$, say $[c_1']$, with links l_1', l_2'; it is easy to see that swapping (l_1, l_2'), and (l_1', l_2) obtains two copies of $[c_1^s]$, so one can reduce that case to the other one where $E \equiv c_1, c_2$.

So suppose that $E \equiv c_1, c_2$ and pick separate traces leading to solutions which contains respectively $[c_1]$, and $[c_2]$ (there must be some, by definition of Γ^*). Because initial complexes are single agents, both l_1, l_2 have to be created along those traces; consider the last such event in both traces, say $[T_1] \to_{r_1} [S_1]$, and $[T_2] \to_{r_2} [S_2]$ creating respectively l_1, and l_2. By atomicity the views of the agents involved in l_1, say $[A_1]$, $[B_1]$, are the same before and after r_1, except of course for the typed links associated to l_1; the same thing holds of $[A_2]$, $[B_2]$, therefore in $[T_1, T_2]$, one can permute the bindings and apply r_1 to $[A_1]$, $[B_2]$, and r_2 to $[A_2]$, $[B_1]$ (by non interference). Using the final condition namely that

$A(a_u), B(a_u) \to A(a_u^1), B(a_u^1)$
$A(a_u^1), B(a_u^1) \to A(a_p^1), B(a_u^1)$
$A(a_u^1), B(a_u^1) \to A(a_u^1), B(a_p^1)$

(a)

$A(a_u) \leftrightarrow A(a_p)$
$A(a_u), A(a_p) \to A(a_u^1), A(a_p^1)$

(c)

$R(a,b), R(a) \to R(a,b^1), R(a^1)$

(b)

$A(l, x_u, r) \to A(l, x_p, r)$
$A(x_p, r), A(l, x_u) \to A(x_p, r^1), A(l^1, x_u)$
$A(x_p, r^1), A(l^1, x_u) \to A(x_u, r^1), A(l^1, x_p)$
$A(x_u, r^1), A(l^1, x_p) \to A(x_p, r^1), A(l^1, x_u)$

(d)

Fig. 12. Rule systems

tests are local, it is easy to see that all computation steps onward commute to that permutation. □

We illustrate now each condition of Prop. 1 in turn.
Local tests: consider the initial complexes $A(a_u)$ and $B(a_u)$ and the rules in Fig. 12.(a), $A(a_p^1), B(a_p^1)$ is in $\gamma\alpha(\Gamma^*) \setminus \Gamma^*$; indeed the last two rules include non local tests.
Acyclicity: take as an initial complex $R(a,b)$ with the rule in Fig. 12.(b), all R-rings, eg $R(a^1, b^1)$, are in $\gamma\alpha(\Gamma^*) \setminus \Gamma^*$.
Non-interference: consider the initial complex $A(a_u)$ and the rules in Fig. 12.(c), $A(a_u^1), A(a_u^1)$ is in $\gamma\alpha(\Gamma^*)\setminus\Gamma^*$; indeed the rule set does not verify non-interference, since $A(a_u)$ should also be allowed to bind with $A(a_u)$.

In any of the above examples, the finite interpretation could be made exact by suitably extending the agent view radii. In Fig. 12.(d) gives an example which no finite radius approximation can interpret exactly. Indeed it is easy to see that, with $A(l, x_u, r)$ as the only initial complex, all generated chains of length > 1 have exactly one $A(x_p)$; whereas any $< n$ radius abstraction α will have A-chains with no $A(x_p)$, and length $\geq 2n$ in $\gamma\alpha(\Gamma^*)$. We shall refer to this model as the GLO model in the next section.

6 Examples

We have considered three examples:
- the early EGF receptor pathway model [20],
- the early FGF (fibroblast growth factor) receptor pathway model [21],
- and the EGF model of Ref. [4].

Those are referred to hereafter as the EGF, FGF, and SBF models.[7]

Proposition 3. *The sets of reachable complexes in the EGF, FGF and SBF models are local.*

[7] The models and relevant outputs of the analysis used in this proof are available at `www.di.ens.fr/~feret/proplx`.

	EGF	FGF	SBF	GLO
Number of rules	39	42	66	4
Abstraction time	0.08 s	0.06 s	0.08 s	0.01 s
Number of complexes	356	79 080	$\simeq 10^{19}$	∞
Complex counting time	<0.01 s	0.09 s	0.04 s	<0.01 s
Enumeration time	0.06 s	85 s	*	*
Number of complexes (non relational analysis)	14 374	709698	$\simeq 10^{25}$	∞
Decontextualization time	0.17 s	0.25 s	0.88 s	0.01
Local (by conjugation)	*true*	*true*	*true*	*false*

Fig. 13. *Times refer to a run on a 2GHz Intel Centrino Duo, 2G RAM. We also give the number of complexes obtained by using a non-relational analysis to show the loss of precision. We skip the enumeration step when the set of complexes is too large. Recall the GLO model is explicitly designed to be non local.*

All the above models can be shown to be acyclic using their contact maps as shown above for the EGF case. Furthermore the rule sets in these models can be made to verify the other assumptions of Prop. 1. This is done by using two transformations on the rule set.

The first transformation is *decontextualization*. One groups rules that perform the same action. Then, for each group, one computes a Boolean encoding of the set of solutions that 1) may match the left hand side of rules and 2) may be reachable –according to the view-based analysis (Section 3). In good cases redundant conditions in left hand side expressions are revealed and one can simplify the rules. This operation of decontextualization is fully automatic and does not modify the transition system (it does change the kinetics of the system when merged rules have different rates but that is not of concern here).

The second transformation of *conjugation* comes into play to deal with the few non local rules that may remain. One adds rules that are in the transitive closure of the transition system (so that the set of reachable complexes remains the same) and invoke decontextualization again. More precisely, whenever an action can only be applied in a specific context, one looks for sequences of rules that allows to simulate the same action in any other reachable context. This second stage is not automated at the moment.

As noted in the introduction, the use of the view-based abstraction is not limited to proving that complex sets are local. The inverse operation of *contextualization* where one enumerates extensions of complexes in a rule is also useful to get rid of non-contextual rule involving agent deletions, or to extract ground rules from a rule set. Another noteworthy application is the approximation of causations and conflict relations between events as static relations between rules; that is useful for simulation [5], and abstracting those at the level of views accelerates greatly their computation.

7 Conclusion

Biological signalling networks are large and generate combinatorial and high-dimensional transition systems which are computationally unwieldy. We have presented in this paper an abstraction of such systems, as represented as κ rule sets, which is a prerequisite for a certain number of tasks to become feasible. Perhaps the most intriguing finding is that this leads naturally to the definition of a class of *local* networks, a rather weak fragment of the set of all κ systems, where one would not *a priori* expect real models to sit. We could prove that previously and independently constructed models actually fall into that class.

Obviously, more examples need to be studied, before one can claim this is the class of natural signalling networks. Suppose however, for the sake of the argument, that biological networks are indeed predominantly local, one wonders why. Our favourite speculation is that a local network can be brought to process signals reasonably well in a variety of circumstances, placing only low demand on the accuracy of the setup (eg kinetic rates), or the reliability of the signal.

One can unfold the hierarchy of classes which has been left implicit in this paper, by investigating larger radii approximations, which would cover a larger class of networks, although we know that no finite radius approximation can cover all cases (see the GLO example). One has to see if a nice characterisation of say 2-local complex sets can be obtained.

Note that there is no need for our views to be of uniform radii, and one could even refine this classification of dimension sets, using collections of non uniform views. Such a theory, which still needs to be developed, would likely characterize the closure and covering properties one needs for soundness. Our present local views would be just one particularly simple instance that is a good computation trade-off between too poor an abstraction (eg that based on discrete coverings) which is fast but retains little information, and the richer ones we just suggested.

Another tempting avenue for future research is to articulate quantitative extensions of those ideas. Specifically, one can use the multiset version of the abstraction map, and derive an approximate differential or stochastic operational model, to be compared with concrete exact simulations. One is looking for a manifestation of locality at the level of quantitative dynamics.

References

1. Priami, C., Regev, A., Shapiro, E., Silverman, W.: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Information Processing Letters (2001)
2. Regev, A., Shapiro, E.: Cells as computation. Nature **419** (September 2002)
3. Regev, A., Panina, E., Silverman, W., Cardelli, L., Shapiro, E.: BioAmbients: an abstraction for biological compartments. Theoretical Computer Science **325**(1) (2004) 141–167
4. Danos, V., Feret, J., Fontana, W., Harmer, R., Krivine, J.: Rule-based modelling of cellular signalling. In Caires, L., Vasconcelos, V., eds.: Proceedings of the 18^{th} International Conference on Concurrency Theory (CONCUR'07). Lecture Notes in Computer Science (Sep 2007)

5. Danos, V., Feret, J., Fontana, W., Krivine, J.: Scalable modelling of biological pathways. In Shao, Z., ed.: In Proceedings of APLAS 2007. Volume 4807. (2007) 139–157
6. Danos, V., Laneve, C.: Formal molecular biology. Theoretical Computer Science **325**(1) (September 2004) 69–110
7. Danos, V., Laneve, C.: Core formal molecular biology. In: Proceedings of the 12th European Symposium on Programming, ESOP'03. Volume 2618 of LNCS., Springer-Verlag (April 2003) 302–318
8. Danos, V.: Agile modelling of cellular signalling. In: Proceedings of ICCMSE'07. (2007)
9. Blinov, M., Yang, J., Faeder, J., Hlavacek, W.: Graph theory for rule-based modeling of biochemical networks. Proc. BioCONCUR 2005 (2005)
10. Hlavacek, W., Faeder, J., Blinov, M., Posner, R., Hucka, M., Fontana, W.: Rules for Modeling Signal-Transduction Systems. Science's STKE **2006**(344) (2006)
11. Milner, R.: Communicating and mobile systems: the π-calculus. Cambridge University Press, Cambridge (1999)
12. Cardelli, L.: Brane calculi. In: Proceedings of BIO-CONCUR'03, Marseille, France. Volume 180 of Electronic Notes in Theoretical Computer Science., Elsevier (2003)
13. Priami, C., Quaglia, P.: Beta binders for biological interactions. Proceedings of CMSB **3082** (2004) 20–33
14. Danos, V., Krivine, J.: Formal molecular biology done in CCS. In: Proceedings of BIO-CONCUR'03, Marseille, France. Volume 180 of Electronic Notes in Theoretical Computer Science., Elsevier (2003) 31–49
15. Feret, J.: Dependency analysis of mobile systems. In: European Symposium on Programming (ESOP'02). Number 2305 in LNCS, Springer-Verlag (2002)
16. Cousot, P., Cousot, R.: Abstract interpretation and application to logic programs. Journal of Logic Programming **13**(2–3) (1992) 103–179
17. Lee, C.Y.: Representation of switching circuits by binary-decision programs. Bell Systems Technical Journal **38** (1959) 985–999
18. Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., Rival, X.: A static analyzer for large safety-critical software. In: Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI'03), San Diego, California, USA, ACM Press (June 7–14 2003) 196–207
19. Milner, R.: Communication and Concurrency. International Series on Computer Science. Prentice Hall (1989)
20. Blinov, M.L., Faeder, J.R., Goldstein, B., Hlavacek, W.S.: A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. BioSystems **83** (January 2006) 136–151
21. Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O., Heath, J., Gaffney, E.: Simulation and verification for computational modelling of signalling pathways. Proceedings of the 37th conference on Winter simulation (2006) 1666–1674