The κ-Lattice: Decidability Boundaries
for Qualitative Analysis in Biological Languages

G. Delzanno[(1)]   C. Di Giusto[(2)]   M. Gabbrielli[(2)]   C. Laneve[(2)]   G. Zavattaro[(2)]

(1)   Dip. di Informatica e Scienze dell'Informazione, Università di Genova, Italia
(2)   Dip. di Scienze dell'Informazione, Università di Bologna, Italia

Abstract

The κ-calculus is a formalism for modelling molecular biology where molecules are terms with internal state and sites, bonds are represented by shared names labelling sites, and reactions are represented by rewriting rules. Depending on the shape of the rewriting rules, a lattice of dialects of κ can be obtained. We analyze the expressive power of some of these dialects by focusing on the thin boundary between decidability and undecidability for problems like reachability and coverability. This analysis may be used, for instance, for excluding the genesis of dangerous substances.

1   Introduction

In recent years we are witnesses of an increasing interest in applications of specification languages used in concurrency as formal models of biological systems. Languages like Petri nets, term rewriting, and process calculi are becoming common idioms for fostering the cooperation between researchers working in biology and computer science [1, 5, 9, 12, 14, 17, 27, 18, 19, 28, 31] . Qualitative analysis like reachability [12, 27] and symbolic model checking [16], and static analysis like abstract interpretation [8] can be used for validation and optimization (e.g. detection of dead rules and dependencies) of models that are used by biologists for experiments in silico (e.g. stochastic simulations). However, general purpose decision procedures are not always applicable to validate formal models of biological systems. Indeed, the level of granularity used in modelling biological mechanisms can dramatically influence the expressive power of the resulting formal languages, as in the case of the passage from basic chemistry (that may be modelled by Petri nets) to bio-chemistry (that requires binding sites, thus becoming Turing-complete) [36]. For this reason, as in other applications of concurrency, an important foundational issue is the study of dialects for which qualitative analysis is computable in an effective way and the

isolation of minimal fragments in which it is proved to be impossible. In this paper, we investigate the boundary between decidability and undecidability of qualitative analysis of biological systems. As a formal model for our analysis, we consider the κ calculus [9]. κ is a formalism for modelling molecular biology where molecules are terms with internal state and with sites, bonds are represented by names that label sites, and reactions are represented by rewriting rules. For example, $EGFR[tk^0](1^z)$ represents a molecule of species $EGFR$ that is not phosphorilated – the internal state tk is 0 – and that is bond to another molecule – its site 1 is labelled with a name z. The reaction in Fig. 1 defines the first step of the *Receptor Tyrosine Kinase* (RTK) growth factor *EGF* (a dimeric form of *EGF* binds two receptors *EGFR*, thus phosphorylating the tyrosine kinase site – *tk* switches from 0 to 1). This reaction is rendered by the following κ rule:

$$EGF(1^x + 2^y), EGF(1^x + 2^z), EGFR(1^y),$$
$$EGFR[tk^0](1^z) \rightarrowtail EGF(1^x + 2^y), EGF(1^x + 2^z), \quad (1)$$
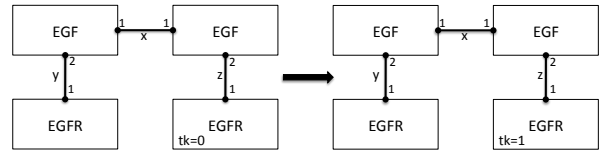$$EGFR(1^y), EGFR[tk^1](1^z)$$

Figure 1. Representation of the κ-rule (1)

A recent contribution turns out to be rather close to the present one [8]. Using abstract interpretation (abstracting away from the multiplicity of molecules – always considered unbounded – and from the exact structure of molecular complexes) authors design an efficient algorithm for computing the set of reachable complexes in a fragment of κ with a *local rule set* and over-approximating the set of reachable complexes in the general case.

As a matter of fact, classical problems, such as reachability and coverability, turn out to be undecidable in κ.

Therefore one is either compelled to design approximated analyses or to study these properties in dialects of κ. We choose the second direction, thus yielding a number of precise analyses that do not abstract away either from the multiplicity of molecules or from the exact structure of complexes. To this aim, we consider a number of κ dialects that, as we discuss in the following, take inspiration from biological phenomena such as the *molecular self-assembly* [33] or the *DNA branch migration* [26]. These dialects are ordered into a lattice by the sublanguage relation – see Figure 2 disregarding the ovals. Let us unravel the lattice with the restrictions imposed to κ to obtain the sublanguages κ^{-n}, κ^{-d}, and κ^{-d-u}. The calculus κ^{-n} follows by removing any form of destruction of molecules (the molecules never decrease). This fragment naturally models those systems where molecules always keep their "identity" even when they are part of a complex because, for example, they can subsequently dissociate from the complex. This is the case of polymers, that is chemical structures obtained by joining monomers that react on complementary surfaces. A simple polymerisation – the linear bidirectional one, where the complementary surfaces of monomers are two (that we respectively call l and r in the following) – is modelled by the following κ^{-n} rules:

$$A(r), A(l) \;\rightarrowtail\; A(r^x), A(l^x) \qquad (2)$$
$$A(r^x), A(l^x) \;\rightarrowtail\; A(r), A(l) \qquad (3)$$

The reaction (2) defines polymerization (the creation of a bond between two monomers with free complementary surfaces); (3) defines depolymerization (the destruction of the bond, but not of the monomers). The additional restriction yielding κ^{-d} is the one that disallows the removal of bonds (depolymerizations are forbidden). This restriction is inspired by molecular self-assembly, which is a process where molecules, initially unbound, adopt a defined arrangement. The DNA-origami method is a popular example of self-assembly that allows to create arbitrary two-dimensional shapes, such as Borromean rings [22], using DNA. In κ^{-d} self-assembly is directly enforced because bonds cannot be broken. The last dialect along this axis, called κ^{-d-u}, is obtained by considering molecules without internal states. In several cases such states are not useful. An example is the DNA self-assembly governed by the Watson-Crick complementary base pairing [32]. We also consider two other subcalculi that forbid destructions of molecules and bonds: κ^{-d-i} and κ^{-d-u-i}. These dialects are obtained from κ^{-d} and κ^{-d-u}, respectively, by restricting reductions to those that never verify the connectedness of reactants. For example, the polymerization (2) is a reaction of this type. It turns out that the Watson-Crick complementary base pairing may be defined in κ^{-d-u-i}.

Our analysis also takes into account a different axis. In [7] a new reaction rule has been introduced, called *ex-*

change. According to this reaction, the interaction between two molecules may flip a bond from one to the other. For example, the reader may consider the case where a thief molecule T may connect to a third site of the monomer A and steals the polymer connected to the site l of A:

$$T(t + s), A(h) \;\rightarrowtail\; T(t^x + s), A(h^x) \qquad (4)$$
$$T(t^x + s), A(h^x + l^y) \;\rightarrowtail\; T(t^x + s^y), A(h^x + l) \;(5)$$

(reaction 5 is an example of bond flipping). Bond flipping allows us to model other interesting DNA systems, such as those based on branch migration used to create, for instance, a nanoscale biped walking along a DNA strand [34]. The calculi including bond flipping are made evident with the superscript $+bf$. Finally, we consider also a more liberal form of flipping, called free flipping (see Figure 3), in which flipping can occur also between two unbound molecules. With free flipping, the thief molecule T can steal the polymer to a monomer without previously connecting to it:

$$T(s), A(l^y) \;\rightarrowtail\; T(s^y), A(l) \qquad (6)$$

For all of the 14 dialects of κ we investigate three problems: the *Reachability Problem* (RP), the *Simple Coverability Problem* (SCP) and the *Coverability Problem* (CP). The RP is the decision problem associated to the existence of a derivation (simulation) from an initial solution to a target. As shown in [12, 16, 27], this problem is of high relevance for validation of formal models of biological systems. The SCP is the decision problem associated to the existence of a derivation from an initial solution to a target with given components, regardless of their multiplicity. SCP is a generalization of the decision problem associated to the static analysis considered in [8]. Finally, CP is the decision problem associated to the existence of a derivation from an initial solution to a target that contains given components: CP is a generalization of RP that can naturally be used to formulate structural properties of biological networks without need of specifying an entire target solution.

Our results about the (un)decidability of RP, SCP, and CP in the κ lattice are illustrated in Figure 2. The undecidability results are proved by modelling Turing complete formalisms in the calculi, while the decidability results are proved by reduction to decidable properties in finite state systems or Petri-nets. As far as the undecidability results are concerned, the most surprising one is the undecidability of CP in κ^{-d-u}. We prove that this very poor fragment of κ – in which molecules have no state and bonds cannot be neither destroyed nor flipped – is powerful enough to encode 2 Counter Machines [25], a Turing complete formalism. It is also interesting to observe that this result about κ^{-d-u} relies on the possibility to test at least the presence of bonds. In fact, κ^{-d-u-i} is no longer Turing complete because CP is decidable for this fragment (CP allows one
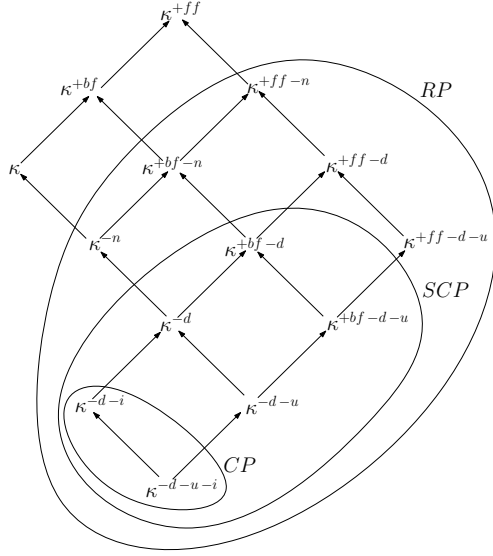
Figure 2. The κ lattice and the (un)decidability of RP, SCP, CP

to test whether a certain complex, for instance representing the termination of a computation, can be produced). While the dialects that include $\kappa^{-d\,-u}$ are Turing complete, many of them retain decidable SCP and/or RP properties. These facts, apparently contrasting with Turing universality of the calculi, are consequences of the following monotonic properties: reactions cannot decrease either (i) the total number of molecules in the solution or (ii) the size of the complexes in the solution. In the calculi satisfying the form of monotonicity (i) we show that it is possible to compute an upper-bound to the number of molecules in the solutions of interest for the analysis of RP. In this way, we reduce our analysis to a finite state system. For the calculi satisfying the form of monotonicity (ii) we show that it is possible to compute an upper-bound to the size of the complexes in the solutions of interest for the analysis of SCP. In this case, even if it is not possible to reduce to a finite state system (because there is no upper-bound to the number of instances of the complexes in the solutions of interest), we can reduce to Petri-nets in which reachability and coverability are decidable.

The paper is organised as follows: Section 2 recalls κ, its fragments and the needed terminology. Section 3 discusses the separation results between the fragments of κ. Section 4 discusses related contributions in literature. Section 5 concludes with few final remarks.

2   Preliminaries

This section introduces κ and its dialects, together with the terminology that is necessary in the sequel.

κ-**calculi.** Two countable sets of *species* $A, B, C, \ldots$, and of *bonds* $x, y, z, \ldots$ are assumed. Species are sorted according to the number of sites $a, b, c, \ldots$ and fields $h, i, j, \ldots$ they possess. Sites may be either bound to other sites or unbound, i.e. not connected to other sites. The configuration of sites are defined by partial maps, called *interfaces* and ranged over by $\sigma, \rho, \ldots$. The interfaces associate to sites either a bond or a special empty value ε, which models the fact that the site is unbound. For instance, if A is a species with three sites, $(2 \mapsto x; 3 \mapsto \varepsilon)$ is one of its interfaces. This map is written $2^x + 3$ (the ε is always omitted). We notice that this σ does not define the state of the site 1, which may be bound or not. Such (proper) partial maps are used in reaction rules in order to abstract from sites that do not play any role in the reactions (similar for evaluations, see below). In the following, when we write $\sigma + \sigma'$ we assume that the domains of σ and σ' are disjoint. The functions $dom(\cdot)$ and $ran(\cdot)$ return the domain and the range of a function. Fields represent the internal state of a species. The values of fields are also defined by partial maps, called *evaluations*, ranged over by $u, v, \ldots$. For instance, if A is a species with three fields, $\{1 \mapsto 5; 2 \mapsto 0; 3 \mapsto 4\}$, shortened into $1^5 + 2^0 + 3^4$, is a possible evaluation. We assume there are finitely many internal states, that is every field is mapped into a finite set of values. As for interfaces, $u + v$, we implicitly assume that the domains of u and v are disjoint.

Definition 1. *A molecule $A[u](\sigma)$ is a term where u and σ are a total evaluation and a total interface of A. Solutions, ranged over by $S, T, \ldots$, are defined by:* $S ::= A[u](\sigma) \mid S, S$. *Bonds in solutions occur at most twice; in case bonds occur exactly twice the solution is* proper. *A pre-solution is a sequence of terms $A[u](\sigma)$ where u and σ are partial functions and bonds occur at most twice. A pre-solution is* proper *if (similarly as before) bonds occur exactly twice. The set of bonds in S is denoted* bonds(S).

In the rest of the paper the composition operator "," is assumed to be associative, so $(S, S'), S''$ is equal to $S, (S', S'')$ (therefore parentheses will be always omitted).

Let $\sigma \leq \sigma'$ if $dom(\sigma) = dom(\sigma')$ and, for every i, if $\sigma(i) \neq \varepsilon$ then $\sigma(i) = \sigma'(i)$ (the two interfaces may differ on sites mapped to the empty value ε by σ as σ' may map such sites to bonds).

Reactions have the shape $\mathsf{L} \rightarrowtail \mathsf{R}$, where L and R are pre-solutions called reactants and products, respectively. The general shape of reactions is defined in the next definition. Following [7], we extend the definition of [9] with exchange reactions, thus the calculus is an extension of the κ-calculus.[1]

Definition 2. *Reactions of the $\kappa^{+\mathit{ff}}$ calculus – the κ calculus with free flipping rules – are either* creations C, *or*

---

[1] Another difference with [9] is that we allow newly produced molecules unbound from existing ones.

destructions *D, or* exchanges *E.*
The format of creations is

$$A_1[u_1](\sigma_1), \ldots, A_n[u_n](\sigma_n)$$
$$\rightarrowtail A_1[u_1'](\sigma_1'), \ldots, A_n[u_n'](\sigma_n'),$$
$$B_1[v_1](\phi_1), \ldots, B_k[v_k](\phi_k)$$

where, for every i, $dom(u_i) = dom(u_i')$, $\sigma_i \leq \sigma_i'$, *and* v_i
and ϕ_i *are total. Reactants and products are proper.*
The format of destructions is

$$A_1[u_1](\sigma_1), \ldots, A_n[u_n](\sigma_n)$$
$$\rightarrowtail A_{i_1}[u_{i_1}'](\sigma_{i_1}'), \ldots, A_{i_m}[u_{i_m}'](\sigma_{i_m}')$$

where $i_1, \ldots, i_m$ *is an ordered sequence in* $[1 \ldots n]$, *for
every* i_j, $dom(u_{i_j}) = dom(u_{i_j}')$, $\sigma_{i_j} \geq \sigma_{i_j}'$, *and if* $i_j \notin$
$\{i_1, \ldots, i_m\}$ *then* σ_{i_j} *is total. Reactants and products are
proper.*
The format of exchanges is

$$A[u](a^x + \sigma), B[v](b + \rho) \rightarrowtail A[u'](a + \sigma), B[v'](b^x + \rho)$$

where $ran(\sigma) = ran(\rho)$.

Creations may change state, produce new bonds between
two unbound sites, or synthesise new molecules. Destruc-
tions behave the other way around. Exchanges are remi-
niscent of the π calculus because they define a migration
of a bond from one reactant to the other. We distinguish
two types of exchanges: the one occurring between con-
nected molecules, called *(connected) bond flipping*, and the
one occurring between disconnected molecules, called *free
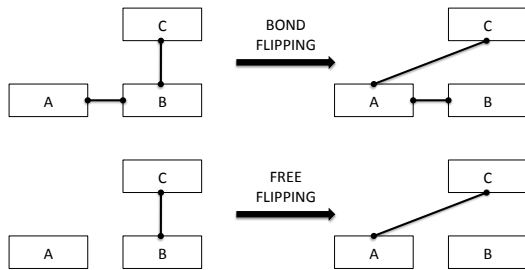(bond) flipping*. These are illustrated below:

Figure 3. Bond flipping and free flipping

The operational semantics of κ^{+ff} calculus uses the fol-
lowing two definitions:

- the *structural equivalence* between solutions, denoted
  $\equiv$, is the least one satisfying (we remind that solutions
  are already quotiented by associativity of ","):
    - $S, T \equiv T, S$;
    - $S \equiv T$ if there exists an injective renaming $\imath$ on
      bonds such that $S = \imath(T)$.

- $A_1[u_1 + u_1'](\sigma_1 \circ \imath + \sigma_1'), \ldots, A_n[u_n + u_n'](\sigma_n \circ \imath + \sigma_n')$ is an $(\imath, u_1' \cdots u_n', \sigma_1', \cdots, \sigma_n')$ *instance* of
  $A_1[u_1](\sigma_1), \ldots, A_n[u_n](\sigma_n)$ if $\imath$ is an injective renam-
  ing on bonds and the maps $u_j + u_j'$ and $\sigma_j \circ \imath + \sigma_j'$ are
  total with respect to the species A_j.

Definition 3. *The reduction relation of the* κ^{+ff} *calculus,
written* $\rightarrow$, *is the least one satisfying the rules:*

- *let* $\mathsf{L} \rightarrowtail \mathsf{R}$ *be a reaction of* κ^{+ff}, S *be an* $(\imath, \widetilde{u}, \widetilde{\sigma})$-
  instance of L, *and* T *be an* $(\imath, \widetilde{u'}, \widetilde{\sigma'})$-*instance of* R.
  Then $S \rightarrow T$;

- *let* $S \rightarrow T$ *and* $(bonds(T) \setminus bonds(S)) \cap bonds(R)$
  $= \emptyset$, *then* $S, R \rightarrow T, R$;

- *let* $S \equiv S'$, $S' \rightarrow T'$, *and* $T' \equiv T$, *then* $S \rightarrow T$.

The κ^{+ff} calculus groups several sub-calculi that have in
turn simpler formats of rules. We have already depicted in
Figure 2 the fragments we study. We move from κ^{+ff} along
two different axes:

1. we restrict reactions by letting $i_m = n$ in destruc-
   tions (forbidding cancellations of molecules), the su-
   perscript $-n$; removing destructions, the superscript
   $-d$; removing destructions and considering species
   with emptyset of sites (removing fields), the super-
   script $-d - u$; removing destructions, fields, and such
   that no bond occurs in the left-hand side of creations
   and exchanges, except the flipping one, the superscript
   $-d - u - i$;

2. we restrict exchanges by allowing bond-flipping only,
   the superscript $+bf$, and by removing exchanges, no
   superscript $+bf$ or $+ff$.

Some of the combinations are empty. For example, a calcu-
lus without checks of bonds and with cancellation of bonds
is meaningless as, in order to remove one bond, it is neces-
sary to test its presence first. The reader may refer to the in-
troduction for formalisations of relevant biological systems
written in these calculi.
Decision problems for qualitative analysis. A first basic
qualitative property is whether a solution eventually pro-
duces "something relevant" or not. Clearly this "something
relevant" can be defined in a variety of ways. In this paper
we consider its formalisation in terms of reachability and
coverability, two standard properties which have been ex-
tensively investigated in many concurrent formalisms. Few
preliminary notions are required.

Definition 4 (Complex). *Given a proper solution, a* com-
plex *is a sub-solution that is connected (there is a path
of bonds connecting every pair of molecules therein) and
proper. Two complexes in a solution are* equal *if they are
structurally equivalent.*

4

Let $S(S)$ be the set of different complexes in S; let also $\to^*$ be the transitive and reflexive closure of $\to$.

Definition 5. RP: *the reachability problem of T from a proper solution S checks the existence of R such that $S \to^* R$ and $R \equiv T$;*

SCP: *the simple coverability problem of T from a proper solution S checks the existence of R such that $S \to^* R$ and $S(R) = S(T)$ and $R \equiv T, T'$, for some T';*

CP: *the coverability problem of T from a proper solution S checks the existence of R such that $S \to^* R$ and $R \equiv T, T'$, for some T'.*

3 (Un)Decidability Results for κ dialects

In this section we study the (un)decidability of RP, SCP, and CP in the κ lattice of Figure 2. The overall results represented in that figure are the consequences of theorems that we detail in the remainder of this section. For each decidability region – one for RP, one for SCP, and one for CP – we prove that the corresponding property is decidable in the top language of the region and undecidable in the bottom language(s) among those not included in the region.

We separate the presentation of our results in two subsections, the first one is devoted to decidability, the latter to undecidability.

3.1 Decidability results

The proofs of decidability follow by reduction to decidable problems in either finite state systems or Place/Transition Petri nets (P/T nets). These nets are an interesting infinite state model for the representation and analysis of parallel processes because they retain several decidability problems, such as reachability or coverability [15]. We recall here the basic notation, for a full description of this computational model see [29].

Definition 6. *A P/T net is a tuple $N = (S, T, F, m_0)$, where S and T are finite sets, called* places *and* transitions, *respectively, such that $S \cap T = \emptyset$. A finite multiset over the set S of places is called a* marking, *and m_0 is the initial marking. F is the transition function associating to each transition t two markings called the* pre-set *and the* post-set *of t.*

The marking m of a P/T net can be modified by means of transitions firing: a transition with pre-set m' and post-set m'' can fire if $m' \subseteq m$; upon transition firing the new marking of the net becomes $(m \setminus m') \cup m''$ where $\setminus$ and $\cup$ are the difference and union operators for multisets, respectively.

Our first positive result is for the κ^{+ff-n} fragment.

Theorem 7. *RP is decidable in κ^{+ff-n}.*

Proof. We reduce RP to the reachability problem in a finite state system. Let $\mathcal{R}$ be a set of κ^{+ff-n} reactions and let S and T be two proper solutions. We notice that, in order for $S \to^* T$, all intermediary solutions traversed by the computation must have a number of molecules which is less or equal to the number n_T of molecules in T. This is because in κ^{+ff-n} it is not possible to delete molecules.

Let $\mathcal{A}$ be the set of species occurring either in S or in a rule of $\mathcal{R}$. Let also $\text{set}^T(\mathcal{A})$ be the set of (proper) solutions with a number of molecules less than n_T. This set is finite *up-to structural equivalence* because the number of sites and fields of species is finite, the values of fields is finite, and the possible combinations of bonds is finite, as well. By mapping every solution R to its canonical representative in the structural equivalence class, called $[R]$, we can build a *finite state system* FSS_T such that, by Definition 3, given two solutions in $\text{set}^T(\mathcal{A})$, $R \to R'$ if and only if $[R] \to [R']$. We conclude the proof by observing that $S \to^* T$ if and only if $[S] \to^* [T]$, and this latter property is decidable in FSS_T. $\square$

The next result is about the decidability of SCP in κ^{+bf-d}. This follows from the property that, in κ^{+bf-d}, the connectedness of two molecules can never be broken.

Lemma 8. *Let S and T be two proper solutions of the κ^{+bf-d} calculus such that $S \to T$. If there exists a path of bonds connecting two molecules in S – i.e. the two molecules are* connected *– then the two molecules are still connected in T (possibly with a different path).*

Proof. Bonds can only be created and flipped in κ^{+bf-d}. In particular, in this last case, a flip occurs if the affected molecules – not only the reactants – are already connected (see the top picture of Figure 3). This entails the property of the lemma. $\square$

Theorem 9. *SCP is decidable for κ^{+bf-d}.*

Proof. We reduce to the *target marking reachability* problem for P/T nets, which is decidable [4]. This problem amounts to checking, given a P/T net P and a target marking m_t, whether a marking m is reachable in P such that $m(p) = 0$ for every place p such that $m_t(p) = 0$, and $m(p') \geq m_t(p')$ for every other place p'.

Let $\mathcal{R}$ be a set of κ^{+bf-d} reactions and S, T and R be proper solutions such that $S(T) = S(R)$ and $R \equiv T, R'$, for some solution R'. Let n_T be the maximum number of molecules of a complex in T.

As a consequence of Lemma 8, if $S \to^* R$, then the complexes occurring in every intermediary solution traversed by the computation have a number of molecules smaller or equal to n_T.

Let $\mathcal{A}$ be the set of species occurring either in S or in a rule of $\mathcal{R}$, and let $\text{SET}^T(\mathcal{A})$ be the set of complexes composed of at most n_T molecules belonging to the species in

5

$\mathcal{A}$. As in the proof of Theorem 7, this set $\mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$ is finite if taken up-to structural equivalence.

We define the following P/T net. The places are the elements of $\mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$. We build the transitions in two steps. Given a rule $\rho : \mathsf{L} \rightarrowtail \mathsf{R}$, we first define RED_ρ as the least set containing all reductions $S_1, \cdots, S_n \rightarrow S'_1, \cdots, S'_m$ such that:

i) S_i and $S'_j \in \mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$ for every i and j;

ii) the reduction is obtained by applying Definition 3 that instantiates ρ with a proof-tree PT,

iii) for every i, S_i is directly involved in the reduction (i.e. at least one molecule of its is an instance of a term in L in the unique leaf of PT).

Condition (iii) ensures that set RED_ρ is finite up to structural equivalence. Indeed, we have that n is less or equal than the number of terms in L, m is less or equal than the number of terms in R, and $\mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$ is finite. For each rule ρ and each reduction $S_1, \cdots, S_n \rightarrow S'_1, \cdots, S'_m$ in RED_ρ we build a P/T transition with pre-set $[S_1], \ldots, [S_n]$ and post-set $[S'_1], \ldots, [S'_m]$. Let m_S and m_T be the initial and final markings corresponding to S and T, respectively. The above P/T net faithfully reproduces the possible computations of S that traverse solutions retaining complexes composed of at most n_T molecules. This allows us to reduce SCP of S to the target marking reachability of m_T in the above P/T net, which is decidable. $\qquad\square$

Our last decidability result regards κ^{-d-i}.

Theorem 10. *CP is decidable in κ^{-d-i}.*

Proof. We reduce to the coverability problem in P/T net. Let $\mathcal{R}$ be a set of κ^{-d-i} reactions and S, T and R be proper solutions such that $R \equiv T, R'$ for some solution R'. Let n_T be the maximum number of molecules of a complex in T.

As in the proof of Theorem 9, let $\mathcal{A}$ be the set of species occurring either in S or in a rule of $\mathcal{R}$, and let $\mathsf{SET}^T(\mathcal{A})$ be the set of complexes composed of at most n_T molecules belonging to the species in $\mathcal{A}$. The set $\mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$ is finite.

We define the following P/T net. Places are elements of $\mathsf{SET}^{T,+}_{\underline{\underline{\equiv}}}(\mathcal{A})$ that extends $\mathsf{SET}^T_{\underline{\underline{\equiv}}}(\mathcal{A})$ with the places $\widehat{A}[u](\sigma)$, for every species $A \in \mathcal{A}$, every evaluation u, and with partial functions σ mapping every site to ε (properly speaking, $\widehat{A}$ is not a molecule because σ cannot be partial). Note that the number of places is finite because the additional places $\widehat{A}[u](\sigma)$, with respect to the P/T net already discussed in Theorem 9, is finite (A is taken from the finite set $\mathcal{A}$, the possible evaluations u are finite and similarly for σ).

Transitions are defined in two steps. Given a rule $\rho : \mathsf{L} \rightarrowtail \mathsf{R}$, we first define RED^+_ρ as the least set containing all reductions $S_1, \cdots, S_n \rightarrow S'_1, \cdots, S'_m$ such that:

i) S_i and S'_j are complexes composed only of molecules belonging to species in $\mathcal{A}$ (possibly with size greater than n_T);

ii) the reduction is obtained by applying Definition 3 that instantiates ρ with a proof-tree PT,

iii) for every i, S_i is directly involved in the reduction (i.e. at least one molecule of its is an instance of a term in L in the unique leaf of PT).

Note that, unlike the proof of Theorem 9, RED^+_ρ can be infinite as we do not impose any restriction to the sizes of S_i. Nevertheless, it is possible to group transitions in RED^+_ρ into finitely many different groups. For every $S_1, \cdots, S_n \rightarrow S'_1, \cdots, S'_m$ in RED^+_ρ, we let a transition with pre-set given by the following places

- $[S_i]$ if S_i has no more than n_T molecules;

- $\widehat{A_1}[u_1](\sigma_1), \cdots, \widehat{A_{m'}}[u_{m'}](\sigma_{m'})$ if S_i has more than n_T molecules and the molecules of S_i that participate to the reduction (the ones that instantiate the terms in L in the unique leaf of PT) are

$$A_1[u_1](\sigma_1 + \rho_1), \cdots, A_{m'}[u_{m'}](\sigma_{m'} + \rho_m)$$

with $\varepsilon \notin ran(\rho_i)$ and $\{\varepsilon\} = ran(\sigma_i)$

and post-set given by the following places

- $[S'_j]$ if S'_j has no more than n_T molecules;

- $\widehat{A_1}[u_1](\sigma_1), \cdots, \widehat{A_{m'}}[u_{m'}](\sigma_{m'})$ if S'_j has more than n_T molecules and the molecules of S'_j that participate to the reduction (the ones that instantiate the terms in R in the unique leaf of PT) are

$$A_1[u_1](\sigma_1 + \rho_1), \cdots, A_{m'}[u_{m'}](\sigma_{m'} + \rho_m)$$

with $\varepsilon \notin ran(\rho_i)$ and $\{\varepsilon\} = ran(\sigma_i)$.

The set of transitions is finite because both the pre-sets and the post-sets use places in $\mathsf{SET}^{T,+}_{\underline{\underline{\equiv}}}(\mathcal{A})$ and their cardinality is less or equal to the number of terms in L and R, respectively.

Let m_S and m_T be the initial and final markings corresponding to S and T, respectively. This P/T net *does not* faithfully represent all the complexes that can be produced by computations starting from S. In fact, while every complex with cardinality less than n_T is represented by a place, this is not the case for complexes bigger than n_T. When such a complex is created, the net removes the structure of bonds, and considers only the states and the free sites of its molecules. However, this information is sufficient for the coverability analysis in the κ^{-d-i}-calculus because, by Lemma 8, the size of a complex cannot decrease,

thus complexes larger than n_T cannot directly produce the complexes of interest for the analysis but can only trigger reactions necessary in order to reach such complexes. As in the κ^{-d-i}-calculus the bond names cannot be tested in the reactants of a reaction, the loss of this information for large structures is not problematic.

This construction allows us to reduce the coverability problem for κ^{-d-i} to the coverability of the marking m_T in P/T net, which is decidable. $\qquad\square$

3.2 Undecidability results

Our undecidability results follow by reducing to undecidable problems such as the halting problem for 2 Counter Machines (2CMs), which is a Turing equivalent formalism. A 2CM [25] is a machine with *two registers* R_1 and R_2 holding arbitrary large natural numbers and a *program P* consisting of a finite sequence of numbered instructions of the following type:

- $j : \mathsf{Succ}(R_i)$: increments R_i and goes to the instruction $j+1$;

- $j : \mathsf{DecJump}(R_i, l)$: if the content of R_i is not zero, then decreases it by 1 and goes to the instruction $j+1$, otherwise jumps to the instruction l;

- $j : \mathsf{Halt}$: stops the computation and returns the value in the register R_1.

A state of the machine is given by a tuple (j, v_1, v_2) where i indicates the next instruction to execute (the program counter) and v_1 and v_2 are the contents of the two registers. The user has to provide the initial state of the machine. In the rest of the paper, we consider 2CMs in which registers are initially set to zero and where the instruction 0 is Halt. Our first negative result is for reachability of a solution in κ.

Theorem 11. *RP is undecidable in κ.*

Proof. We reduce the termination problem for 2CMs to RP. Let M be a 2CM with n instructions. To encode it in κ we use five species:

1. P is the program counter; it retains one field with values in $[0, \ldots, n]$ and no site;

2. Z_1 and Z_2, both with one site, represent the value 0;

3. R_1 and R_2, both with two sites, represent the unity to be added to or removed from registries.

Let $j, l \in [0..n]$ and let $i \in \{1, 2\}$. The encoding $\llbracket \cdot \rrbracket_\kappa$ is defined in Figure 4.

It turns out that the 2CM halts if and only if the solution $P[1^0], Z_1(1), Z_2(1)$ is reachable from the initial state. Therefore we conclude that RP is undecidable in κ. $\quad\square$

$\llbracket j : \mathsf{Succ}(R_i) \rrbracket_\kappa \ =$

$$\begin{cases} P[1^j], Z_i(1) \rightarrowtail P[1^{j+1}], Z_i(1^x), R_i(1^x + 2) \\ P[1^j], R_i(2) \rightarrowtail P[1^{j+1}], R_i(2^x), R_i(1^x + 2) \end{cases}$$

$\llbracket j : \mathsf{DecJump}(R_i, l) \rrbracket_\kappa \ =$

$$\begin{cases} P[1^j], Z_i(1) \rightarrowtail P[1^l], Z_i(1) \\ P[1^j], Z_i(1^x), R_i(1^x + 2) \rightarrowtail P[1^{j+1}], Z_i(1) \\ P[1^j], R_i(2^x), R_i(1^x + 2) \rightarrowtail P[1^{j+1}], R_i(2) \end{cases}$$

$\llbracket j : \mathsf{Halt} \rrbracket_\kappa \ =$

$$\begin{cases} P[1^j], Z_1(1), Z_2(1) \rightarrowtail P[1^0], Z_1(1), Z_2(1) \\ P[1^j], Z_i(1^x), R_i(1^x + 2) \rightarrowtail P[1^j], Z_i(1) \\ P[1^j], R_i(2^x), R_i(1^x + 2) \rightarrowtail P[1^j], R_i(2) \end{cases}$$

Figure 4. Enconding 2CMs in κ.

Theorem 12. *SCP is undecidable in κ^{-n}.*

Proof. We reduce the termination problem for 2CMs to SCP in κ^{-n}. For this, we modify the encoding of 2CMs used for κ in the previous theorem as follows:

- a binary field to the species R_1 and R_2 is added. When this field is zero, the molecule is considered garbage, otherwise it is a valid one.

Without loss of generality, we assume that the two registers are incremented at least once.

The encoding $\llbracket \cdot \rrbracket_{\kappa^{-n}}$ is defined in Figure 5.

Namely, the increment refines the previous encoding by setting to 1 the field of the new R; the decrement, rather than removing one molecule at the end of the register, which is not allowed in κ^{-n}, removes the bond and resets the field to zero; the halt operation turns every molecule R to garbage. We now observe that any solution that contains the complexes in the target solution

$$T = P[1^0], Z_1(1), Z_2(1), R_1[1^0](1+2), R_2[1^0](1+2)$$

encodes a halting configuration. Thus, termination of a 2CM can be reduced to SCP for the corresponding κ^{-n} encoding and for the target solution T. $\quad\square$

We observe that, without using fields and destructions, as in κ^{-d-u}, it is not possible to reuse the encoding scheme of Theorems 11 and 12. In the following theorem we prove that the use of creations in molecules without fields is sufficient to make the CP problem undecidable.

Theorem 13. *CP is undecidable in κ^{-d-u}.*

7

$[\![j : \mathsf{Succ}(R_i)]\!]_{\kappa^{-n}} \; =$

$$\begin{cases} P[1^j], Z_i(1) \rightarrowtail P[1^{j+1}], Z_i(1^x), R_i[1^1](1^x + 2) \\ P[1^j], R_i(2) \rightarrowtail P[1^{j+1}], R_i(2^x), R_i[1^1](1^x + 2) \end{cases}$$

$[\![j : \mathsf{DecJump}(R_i, l)]\!]_{\kappa^{-n}} \; =$

$$\begin{cases} P[1^j], Z_i(1) \rightarrowtail P[1^l], Z_i(1) \\ P[1^j], Z_i(1^x), R_i[1^1](1^x + 2) \rightarrowtail \\ \qquad\qquad P[1^{j+1}], Z_i(1), R_i[1^0](1 + 2) \\ P[1^j], R_i(2^x), R_i[1^1](1^x + 2) \rightarrowtail \\ \qquad\qquad P[1^{j+1}], R_i(2), R_i[1^0](1 + 2) \end{cases}$$

$[\![j : \mathsf{Halt}]\!]_{\kappa^{-n}} \; =$

$$\begin{cases} P[1^j], Z_1(1), Z_2(1) \rightarrowtail P[1^0], Z_1(1), Z_2(1) \\ P[1^j], Z_i(1^x), R_i[1^1](1^x + 2) \rightarrowtail \\ \qquad\qquad P[1^j], Z_i(1), R_i[1^0](1 + 2) \\ P[1^j], R_i(2^x), R_i[1^1](1^x + 2) \rightarrowtail \\ \qquad\qquad P[1^j], R_i(2), R_i[1^0](1 + 2) \end{cases}$$

Figure 5. Encoding 2CMs in κ^{-n}.

Proof. We define an encoding of 2CMs by using constructions on species with emptysets of fields. Instructions are implemented by species P_j with a site 1 that may be bound to a molecule of species D. When this happens, the instruction is disabled. A further species $Halt$ with no sites will represent a terminating state. Registers are implemented by grids of increasing height (see Figure 6). The first column
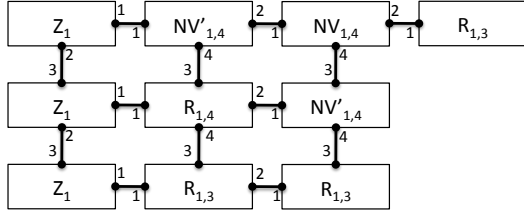

Figure 6. Grid representing the register R_1.

consists of Z_i molecules with three sites; the other nodes of the grid, called *register molecules*, are either $R_{i,j}$ molecules or $NV_{i,j}$ or $NV'_{i,j}$ molecules, $i \in \{1, 2\}$ and j ranging over instruction numbers, all retaining 4 sites. The meaningful part of the grid is the topmost row: the number of molecules $R_{i,j}$ therein represents the value of the corresponding register while the other rows represent previous values (we add a new row when performing a decrement). For instance, the register in Figure 6 contains the value 1 obtained after two increments –performed by the instruction with index

3–, two decrements –performed by the instruction with index 4–, and a subsequent increment –performed by the instruction with index 3. The full definition of the rules for incrementing and decrementing the registers can be found in Appendix A, here we describe their overall behaviour.

The encoding of $[\![j : \mathsf{Inc}(R_{i,j})]\!]_{\kappa^{-d-u}}$ increases the topmost row of the grid with a molecule $R_{i,j}$. The encoding of $[\![j : \mathsf{DecJump}(R_i, l)]\!]_{\kappa^{-d-u}}$ is more complex. The key idea is to copy the topmost row of the grid (from left to right according to the graphical representation of the grid in Figure 6) reducing, if possible, the number of molecules $R_{i,j'}$. This is obtained replacing the first encountered $R_{i,j'}$ molecule with the molecule $NV'_{i,j}$. If there is no such molecule available, all molecules in the new topmost row will be of species $NV_{i,j}$ (i.e. the kind of molecule used to copy molecules of species $NV_{i,j'}$ or $NV'_{i,j'}$). The copy records j in the second index of the register molecules: in this way, when the copy is finished (i.e. the new instance of Z_i is produced) it is possible to release the molecule representing the next instruction, that is P_{j+1} in case the decrement succeeded, P_s otherwise. The encoding $[\![j : \mathsf{Halt}]\!]_{\kappa^{-d-u}}$ simply produces the $Halt$ molecule.

The encoding satisfies the following property: the 2CM halts if and only the solution $P_1(1), Z_1(1 + 2 + 3), Z_2(1 + 2 + 3)$ in the corresponding κ^{-d-u} encoding can produce molecule $Halt$. Thus, termination of 2CMs is reduced to CP with target solution $T = Halt$. Therefore the undecidability of CP in κ^{-d-u}. $\qquad\square$

Our last negative result is for the fragment $\kappa^{+ff-d-u}$.

Theorem 14. *SCP is undecidable in $\kappa^{+ff-d-u}$.*

Proof. (Sketch) We proceed as described in Theorem 13 assuming, without loss of generality as in Theorem 12, that the two registers are incremented at least once. The new construction adds rules that come into play in case the $Halt$ molecule is produced. In fact, the molecule $Halt$ triggers the "destruction" of the grids representing the registers: one molecule is produced for each end of each bond, and the bond is passed to such new molecule. Following this approach we know a priori the exact structure of the structures that will be available at the end of the computation in case of 2CM termination. $\qquad\square$

4 Related work

In this section we discuss some related works by first focusing on formal models specifically proposed for describing biological systems and then considering more generally the fields of term/graph rewriting and process calculi.

As we said in the Introduction, the closest work to this contribution is [8] where a syntactic restriction entailing a form of SCP is proposed. This restriction – κ with local

rule sets – is orthogonal to the ones proposed in this paper. It does not cover the reachability analysis of finite structures with recurrent patterns, such as finite polymers. In these cases, the analysis in [8] yields an over-approximation of the reachable complexes. How much reasonable is this over-approximation is not clear.

Apart from κ, the literature reports several proposals for describing (and reasoning on) biological systems, which use a variety of formal tools, including process calculi, term/graph rewriting, (temporal) logic, and rule based languages. However, the expressive power of most of these formalisms is the one of Petri nets. Therefore, the decidability of reachability and coverability problems is an immediate consequence of the corresponding results on Petri nets.

Formalisms whose expressive power is similar to κ, miss results analogous to those contained in this paper. For example, the biochemical abstract machine Biocham [17, 18] is a rule-based model similar to κ. However reactions are constrained to specify completely the reagent solution, unlike κ where reactions partially specify reactants and products. It is worth noticing that the Biocham constraint do not allow finite descriptions of rules creating polymers of arbitrary length. As a consequence, when considering purely qualitative aspects, i.e. removing kinetic quantities, the Biocham can be reduced to a classical Petri net [17]. Another rule-based model for describing and analysing biological processes is Pathway Logic [14, 31]. This model is based on rewrite logic, which allows to describe biological entities and their relations at different levels of abstractions and granularity by using elements of an algebraic data type (to describe states) and rewrite rules (to describe transitions between states and therefore behaviours). Even though Pathway Logic models of biological processes are developed in Maude system, which is Turing complete, yet the analysis of biological systems uses the, so called, Pathway Logic Assistant for representing models in terms of Petri Nets [31]. Therefore, also in this case, the relevant decidability results derive from the analogous results on Petri nets. This is the case also for the model used in [19].

A different model, based on graph transformation has been proposed by Blinov et al. [1]. However, in this case, the relevant properties (e.g. membership of a given species in a reaction network) are semi-decidable and we are not aware of suitable restrictions on the general model that ensure decidability for some of them.

As regards the fields of term/graph rewriting and process calculi, we have not find results from which we can derive immediately those we have obtained for κ. In particular, for term rewriting systems, the reductions to Petri net reachability can be applied to decide reachability for associative-commutative ground term rewriting (AC) [24] and for Process Rewrite Systems (PRS) [23]. However, AC and PRS are more expressive than Petri nets, but strictly

less expressive than Turing machines [23]. On the other hand our positive results are given for fragments of κ that are Turing-complete. As such, the set of derivatives of a κ solution may not be a regular set of terms. Thus, decision procedures based on tree automata like those proposed in fragments of non-ground term rewriting [6, 10, 20, 30] cannot be applied to the κ-lattice.

Decidability results for reachability in process calculi like Mobile Ambients, Boxed Ambients, and Bio-ambients are given in [2, 3, 4, 11, 35]. These results are obtained for fragments (or for weak semantics) that ensure the monotonicity of the generated ambient structures. In addition they consider process calculi (Mobile/Boxed/Bio Ambients) which operate on tree-like structures and without fresh name generation. This contrasts with the dialect of κ of Figure 2, that operate on (possibly cyclic) graph-structures and admit dynamic creation of new names (bonds).

Concerning Graph Rewriting Systems (GRS) there exist folk theorems about reachability that state its undecidability in full-fledged GRS and its decidability for GRS in which rules do not add new nodes. We are not aware of (un)decidability results for decision problems like reachability and coverability in graph rewriting systems with features similar to those considered in our κ-lattice. The only specific results we are aware of are those given for reachability in context-free graph grammars [13] and for coverability in GRS that are *well-structured* with respect to the graph minor relation [21]. However, we consider here more general rules than those of context-free graph grammars. Furthermore, we do not see how to apply the decision procedure proposed in [21] to languages in the κ-lattice that, in general, do not enjoy strict compatibility with respect to the graph minor ordering.

5 Conclusions

We have investigated three decidability problems for several κ dialects. These problems allow one to check whether, starting from a given initial solution, a sequence of reactions described in the κ formalism produces a solution having some specific features. Hence our results, summarized in Figure 2, can be seen as a first step in the direction of qualitative analysis of κ calculus. Besides presenting techniques for qualitative analysis, we also characterise the computational power of κ-like biologically inspired models. In this respect, the main result is that we can remove bond and molecule destruction and the internal state of molecules from κ without losing Turing completeness (see the modelling of 2 Counter Machines presented in " the proof of Theorem 13). On the contrary, if we remove the possibility to test the presence of one bond in a reaction, the calculus is no longer Turing universal (see Theorem 10).

Our work can be extended along at least two lines. First,

several other fragments of κ can be considered for a similar investigation. Notably nanoκ that admits at most two reactants. In particular, our encoding of a 2CM into κ^{-d-i} uses ternary (at the left hand side) rules and we conjecture that a 2CM cannot be encoded faithfully into κ^{-d-i} with binary rules only. Second, there are several other interesting properties to investigate, for example a form of coverability where one admits complexes strictly larger than the original ones. In this perspective, we plan to exploit the theory of well structured transition systems as done in [21] to prove decidability of coverability w.r.t. the graph minor relation in classes of graph rewriting systems.

References

[1] M. L. Blinov, J. Yang, J. R. Faeder, and W. S. Hlavacek. Graph theory for rule-based modeling of biochemical networks. In *Transactions on Computational Systems Biology VII*, volume 4230 of *LNCS*, pages 89–106, 2006.

[2] I. Boneva and J.-M. Talbot. When ambients cannot be opened. *Theor. Comput. Sci.*, 333(1-2):127–169, 2005.

[3] N. Busi and G. Zavattaro. Deciding reachability in mobile ambients. In *ESOP*, volume 3444 of *LNCS*, pages 248–262, 2005.

[4] N. Busi and G. Zavattaro. Reachability analysis in boxed ambients. In *ICTCS*, volume 3701 of *LNCS*, pages 143–159, 2005.

[5] L. Cardelli and G. Zavattaro. On the Computational Power of Biochemistry. In *AB'08*, volume 5147 of *LNCS*, pages 65–80. Springer-Verlag, 2008.

[6] J.-L. Coquidé, M. Dauchet, R. Gilleron, and S. Vágvölgyi. Bottom-up tree pushdown automata and rewrite systems. In *RTA*, volume 488 of *LNCS*, pages 287–298, 1991.

[7] A. Credi, M. Garavelli, C. Laneve, S. Pradalier, S. Silvi, and G. Zavattaro. Modelizations and simulations of nano devices in nanok calculus. In *CMSB 2007*, volume 4695 of *LNCS*, pages 168–183, 2007.

[8] V. Danos, J. Feret, W. Fontana, and J. Krivine. Abstract interpretation of cellular signalling networks. In *VMCAI*, volume 4905 of *LNCS*, pages 83–97, 2008.

[9] V. Danos and C. Laneve. Formal molecular biology. *Theor. Comput. Sci.*, 325(1):69–110, 2004.

[10] M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *LICS*, pages 242–248, 1990.

[11] G. Delzanno and R. Montagna. On reachability and spatial reachability in fragments of bioambients. *Electr. Notes Theor. Comput. Sci.*, 171(2):69–79, 2007.

[12] D. L. Dill, M. Knapp, P. Gage, C. L. Talcott, K. Laderoute, and P. Lincoln. The pathalyzer: A tool for analysis of signal transduction pathways. In *SBRG*, volume 4023 of *LNCS*, pages 11–22. Springer, 2005.

[13] F. Drewes, H.-J. Kreowski, and A. Habel. Hyperedge replacement, graph grammars. In *Handbook of Graph Grammars*, pages 95–162. World Scientific, 1997.

[14] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and M. K. Sönmez. Pathway logic: Symbolic analysis of biological signaling. In *Pacific Symposium on Biocomputing*, pages 400–412, 2002.

[15] J. Esparza and M. Nielsen. Decidability Issues for Petri Nets-a Survey. *Bulletin of the European Association for TCS*, 52:245–262, 1994.

[16] F. Fages. Symbolic model-checking for biochemical systems. In *ICLP*, volume 2916 of *LNCS*, page 102. Springer, 2003.

[17] F. Fages and S. Soliman. Formal cell biology in biocham. In *SFM*, volume 5016 of *LNCS*, pages 54–80, 2008.

[18] F. Fages, S. Sollman, and N. Chabrier-Rivier. Modelling and querying interaction networks in the biochemical abstract machine biocham. *Journal of Biological Physics and Chemistry*, 4:64–73, 2004.

[19] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In *SFM*, volume 5016 of *LNCS*, pages 215–264, 2008.

[20] F. Jacquemard. Decidable approximations of term rewriting systems. In *RTA*, volume 1103 of *LNCS*, pages 362–376, 1996.

[21] S. Joshi and B. König. Applying the graph minor theorem to the verification of graph transformation systems. In *CAV*, volume 5123 of *LNCS*, pages 214–226, 2008.

[22] C. Mao, W. Sun, and N. Seeman. Assembly of Borromean rings from DNA. *Nature*, 386(6621):137 –138, 1997.

[23] R. Mayr. Process rewrite systems. *Inf. Comput.*, 156(1-2):264–286, 2000.

[24] R. Mayr and M. Rusinowitch. Reachability is decidable for ground ac rewrite systems. In *Infinity '98*, pages 53–64, 1998.

[25] M. Minsky. *Computation: finite and infinite machines*. Prentice Hall, 1967.

[26] I. Panyutin and P. Hsieh. The kinetics of spontaneous DNA branch migration. *Proc. National Academy of Science USA*, 91(6):2021–2025, 1994.

[27] M. Peleg, I. Yeh, and R. B. Altman. Modelling biological processes using workflow and petri net models. *Bioinformatics*, 18(6):825–837, 2002.

[28] A. Phillips and L. Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In *CMSB*, volume 4695 of *LNCS*, pages 184–199, September 2007.

[29] W. Reisig. *Petri nets: an introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[30] K. Salomaa. Deterministic tree pushdown automata and monadic tree rewriting systems. *J. Comput. Syst. Sci.*, 37(3):367–394, 1988.

[31] C. L. Talcott. Pathway logic. In *SFM*, volume 5016 of *LNCS*, pages 21–53, 2008.

[32] J. Watson and F. Crick. A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738, 1953.

[33] G. Whitesides, J. Mathias, and C. Seto. Molecular self-assembly and nanochemistry – a chemical strategy for the synthesis of nanostructures. *Science*, 254:1312–1319, 1991.

[34] P. Yin, M. Harry, M. Choi, R. Colby, R. Calvert, and N. Pierce. Programming biomolecular self-assembly pathways. *Nature*, 451:318–322, 2008.

[35] G. Zavattaro. Reachability analysis in bioambients. *Electron. Notes Theor. Comput. Sci.*, 227:179–193, 2009.

[36] G. Zavattaro and L. Cardelli. Termination problems in chemical kinetics. In *CONCUR*, volume 5201 of *LNCS*, pages 477–491, 2008.

A   Details of the proof of Theorem 13.

In this appendix we formally define the encoding of 2CMs in κ^{-d-u} informally discussed in the proof of Theorem 13.

The encoding of an increment $[\![j : \mathsf{Succ}(R_i)]\!]_{\kappa^{-d-u}}$ is defined by the three rules

$$
\begin{aligned}
P_j(1), Z_i(1) &\rightarrowtail P_j(1^x), D(1^x), Z_i(1^y), \\
&\qquad R_{i,j}(1^y + 2 + 3 + 4), P_{j+1}(1) \\
P_j(1), R_{i,j'}(2 + 3) &\rightarrowtail P_j(1^x), D(1^x), R_{i,j'}(2^y + 3), \\
&\qquad R_{i,j'}(1^y + 2 + 3 + 4), P_{j+1}(1) \\
P_j(1), NV_{i,j'}(2 + 3) &\rightarrowtail P_j(1^x), D(1^x), NV_{i,j'}(2^y + 3), \\
&\qquad R_{i,j'}(1^y + 2 + 3 + 4), P_{j+1}(1)
\end{aligned}
$$

The encoding of a decrement

$$
[\![j : \mathsf{DecJump}(R_i, l)]\!]_{\kappa^{-d-u}}
$$

is reported in Figure 7.

The encoding of an halt instruction $[\![j : \mathsf{Halt}]\!]_{\kappa^{-d-u}}$ is defined by the rule

$$
P_j(1) \rightarrowtail P_j(1^x), D(1^x), Halt
$$

$$P_j(1), Z_i(1) \rightarrowtail \quad P_j(1^x), D(1^x), Z_i(1^y), P_l(1)$$

$$P_j(1), R_{i,j'}(2+3) \rightarrowtail \quad P_j(1^x), D(1^x), R_{i,j'}(2+3^y),$$
$$NV'_{i,j}(1+2+3+4^y)$$

$$P_j(1), X_{i,j'}(2+3) \rightarrowtail \quad P_j(1^x), D(1^x), X_{i,j'}(2+3^y),$$
$$NV_{i,j}(1+2+3+4^y) \qquad (X \in \{NV, NV'\})$$

$$X_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), W_{i,j'}(2+4^y) \rightarrowtail \quad X_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), W_{i,j'}(2^u+4^y),$$
$$X_{i,j}(1+2^z+3+4^u) \quad (X, W \in \{NV, NV'\}, \ Y \in \{NV, NV'\})$$

$$NV_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2+4^y) \rightarrowtail \quad NV_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2^u+4^y),$$
$$NV'_{i,j}(1+2^z+3+4^u) \qquad (Y \in \{NV, NV'\})$$

$$NV'_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2+4^y) \rightarrowtail \quad NV'_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2^u+4^y),$$
$$R_{i,j}(1+2^z+3+4^u) \qquad (Y \in \{NV, NV'\})$$

$$R_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2+4^y) \rightarrowtail \quad R_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), R_{i,j'}(2^u+4^y),$$
$$R_{i,j}(1+2^z+3+4^u) \qquad (Y \in \{R, NV, NV'\})$$

$$R_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), X_{i,j'}(2+4^y) \rightarrowtail \quad R_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), X_{i,j'}(2^u+4^y),$$
$$NV'_{i,j}(1+2^z+3+4^u) \quad (X \in \{NV, NV'\}, Y \in \{R, NV, NV'\})$$

$$R_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3) \rightarrowtail \quad R_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3^u),$$
$$Z_{i,j}(1^u+2^z+3), P_{j+1}(1) \qquad (Y \in \{R, NV, NV'\})$$

$$NV'_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3) \rightarrowtail \quad NV'_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3^u),$$
$$Z_{i,j}(1^u+2^z+3), P_{j+1}(1) \qquad (Y \in \{R, NV, NV'\})$$

$$NV_{i,j}(1+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3) \rightarrowtail \quad NV_{i,j}(1^z+4^x), Y_{i,j'}(1^y+3^x), Z_{i,j'}(2^y+3^u),$$
$$Z_{i,j}(1^u+2^z+3), P_l(1) \qquad (Y \in \{R, NV, NV'\})$$

Figure 7. Encoding of decrement instructions $[\![j : \mathsf{DecJump}(R_i, l)]\!]_{\kappa^{-d-u}}$ in κ^{-d-u}.