Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science 218 (2008) 97–109
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Expressivity in the κ Family

Cosimo Laneve and Antonio Vitale

Dipartimento di Scienze dell'Informazione, Università di Bologna

Abstract

In this paper we study implementation of κ calculus into nanoκ calculus – called *self-assembling* of κ in nk. The former is a model for molecular biology that rewrites graphs of molecules in one step; the latter is a calculus similar to κ that only admits binary interactions. We give a solution of the self-assembling of κ in nanoκ that is divergent and we show the nonexistence of deterministic solutions retaining "reasonable" properties.

Keywords: Calculi for molecular biology, reactions, locality, self-assembling, expressivity

1 Introduction

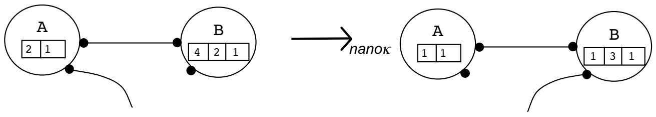
The κ calculus has been introduced in [1] for modelling molecular biology in a formal way. It is a graph rewriting system where nodes represent molecules and edges represent bonds. Nodes retain a finite information, typically about the shape of the molecule or about connected molecules. The semantics allows monotone rewritings of finite graphs whose nodes are in specific states into finite graphs in such a way that changes to a solution are always *localized* to the rewriting part. Monotonicity constraints rewritings to either create or destruct molecules and bonds.

The κ calculus, being as much simple and close to biology as possible, admits rewriting rules where several molecules may interact at a time. The question that was raised already in [1] is whether κ calculus may be implemented in a calculus with binary reactants only or not. This problem, called *self-assembling*, had a positive answer in a variant of κ calculus – the mκ calculus – with binary rewriting rules and *multiedges*. The idea was to use these multiedges as logs of the reactions. The check that reactants are connected, as prescribed in the left-hand side of the reaction, reduced to verify that the connected molecules all share the same log.

Some years later, a new formalism – the nanoκ calculus – similar to κ and mκ, was introduced for modelling nano-technologies [2]. This calculus has binary interactions (as mκ) but no multiedge is admitted (as in κ). Some expressivity is recovered by admitting a new binary rule – the *exchange* – that allows an end of a

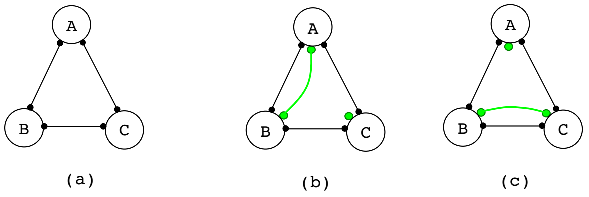
doi:10.1016/j.entcs.2008.10.007

bond to be passed from one molecule to another. This rule, which is illustrated in the following picture

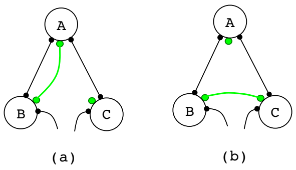


has been motivated by process calculi, where it is customary to receive a channel name and communicating on such a channel in the continuation. However, it was not clear whether nanoκ calculus was expressive enough for implementing the κ calculus.

In this paper we demonstrate that the self-assembling of κ into nanoκ is possible and we define the protocol. The solution is similar to the one from κ to mκ, except that the check verifying the proper connection of reactants is performed by percolating a bond among them. To illustrate the point, consider the structure (a) below, and assume that A wants to verify that the B and C to which it is connected are connected between them.



Molecules A and B create a bond (the one to be exchanged), as illustrated in picture (b). Then the connection between A and C makes this bond to be exchanged as in figure (c). This is a successful state because B and C are connected by two bonds (and the success is eventually reported to A). It is worth to notice that this algorithm fails if B and C are not connected, as illustrated below:



(there is only one bond connecting B and C).

The self-assembling protocols proposed in [1,3] and in this paper avoid deadlocks by admitting divergent computations. A relevant question is whether a deterministic, not-divergent self-assembling encoding κ into a calculus with binary reactants, such as mκ or nanoκ, does exist or not. In this paper we show that, under "reasonable" constraints, such a protocol does not exist. This case is close to the comparison of the expressive power of synchronous and asynchronous π calculi done by Palamidessi [6]. As in that case, we require the self-assembling protocol to be uniform – homomorphic with respect to parallel composition and preserving the connectedness of molecules – and semantically reasonable – preserving termination and the complexes.

However, it turns out that these constraints are inadequate to exclude convergent self-assembling protocols of κ because these protocols must also redefine the reaction rules. That is, since every κ reaction is encoded by a set of low-level ones, we need to regulate this set in order to avoid misbehaviours. For example, a malicious protocol (which is uniform and semantically reasonable) might grab more material then necessary for the reaction (in the worst case, all the material in the solution) and release it after the reaction has been performed, thus being inconsistent with the locality principle of the κ family.

In order to localize the effects of protocols we also add a constraint called *twinning*: some reactions between those implementing a κ reaction $\mathsf{L} \to \mathsf{R}$ have a twin one in the protocol that undoes the effects on bonds. In particular, if $\mathsf{L} \to \mathsf{R}$ is a creation then every destruction in its protocol has a corresponding creation restoring the bonds in the reactants. This means that if the protocol of a creation unbinds a molecule then the released molecule may be rebound, thus yielding either a previous solution – and the computation may diverge – or a previous complex (with a different state). In this latter case, if the complex is too big with respect to the complexes in R then the protocol is not semantically reasonable.

The paper is structured as follows. In Section 2 we define the calculi of the κ family (κ, $\mathtt{m}\kappa$, and $\mathtt{nano}\kappa$ calculus). In Section 3 we discuss the self-assembling problem for κ, recall the protocol in $\mathtt{m}\kappa$ and define the protocol in $\mathtt{nano}\kappa$ calculus. The Section 4 discusses the problem of not-divergent self-assembling protocols.

2 The κ family: syntax and semantics

Two disjoint countable sets of names will be used: a set of *species*, ranged over by A, B, C, $\cdots$; and a totally ordered set of *bonds*, ranged over by x, y, z, $\cdots$. Species are sorted according to the number of *fields* and *sites* they possess. Let $\mathfrak{s}_f(\cdot)$ and $\mathfrak{s}_s(\cdot)$ be two functions returning naturals; the numbers 1, 2, $\cdots$, $\mathfrak{s}_f(A)$ and 1, 2, $\cdots$ $\mathfrak{s}_s(A)$ are respectively the fields and the sites of A. ($\mathfrak{s}_f(A) = 0$ means there is no field; $\mathfrak{s}_s(A) = 0$ means there is no site). In the following, fields are ranged over by h, i, j, $\cdots$; sites are ranged over by a, b, c, $\cdots$.

Sites may be either *bound* to other sites or *unbound*, i.e. not connected to other sites. The state of sites are defined by maps, called *interfaces* and ranged over by σ, ρ, $\cdots$. Given a species A, its interfaces are *partial functions* from $\{1, \cdots, \mathfrak{s}_s(A)\}$ to the set of bonds or a special empty value ε. A site a is bound with bond x in σ if $\sigma(a) = x$; it is unbound if $\sigma(a) = \varepsilon$. For instance, if A is a species with three sites, $(2 \mapsto x, 3 \mapsto \varepsilon)$ is one of its interfaces. In order to ease the reading, we write this map as $2^x + 3$ (the empty value is always omitted). This interface σ does not define the state of the site 1, which may be bound or not. In the following, when we write $\sigma + \sigma'$ we assume that the domains of σ and σ' are disjoint.

Fields represent the internal state of a species. The values of fields are defined by maps, called *evaluations*, and ranged over by u, v, $\cdots$. For instance, if A is a species with three fields, $[1 \mapsto 5, 2 \mapsto 0, 3 \mapsto 4]$ is a possible evaluation. As before, we write this map as $1^5 + 2^0 + 3^4$. We assume there are finitely many internal states, that is every field h is mapped into a *finite set of values.* As for interfaces, we use *partial* evaluations and, when we write the union of evaluations $u+v$, we implicitly assume that the domains of u and v are disjoint.

Definition 2.1 A *molecule* $A[u](\sigma)$ is a term where u and σ are respectively a total evaluation of A and a total interface of A. *Solutions*, ranged over by S, T, $\cdots$, are defined by the following grammar

$$\mathsf{S} \quad ::= \quad A[u](\sigma) \;\mid\; \mathsf{S},\mathsf{S}$$

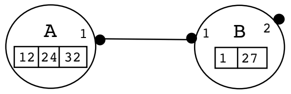
The operator "," is assumed to be associative, so $(\mathsf{S},\mathsf{S}'),\mathsf{S}''$ is equal to $\mathsf{S},(\mathsf{S}',\mathsf{S}'')$ (and we always omit parentheses).

Solutions retain the property that bonds always occur at most twice. A solution is *proper* if bonds occur exactly twice. Let $\mathtt{bonds}(\mathsf{S})$ be the bonds of S.

The calculi in the κ family retain the same terms and differ for the shape of reactions. We define the reactions and the transition system of the κ calculus and, later on, we discuss the reactions of the other calculi in the family. Few preliminary definitions are in order:

- we write $\sigma \leq \sigma'$ if $\mathrm{dom}(\sigma) = \mathrm{dom}(\sigma')$ 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: σ' may map such sites to bonds);
- 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 it retains the property that bonds occur exactly twice.

The κ calculus retains an intelligible graphical notation [5]. For example the solution $A[1^{12}+2^{24}+3^{32}](1^x),B[1^1+2^{27}](1^x+2)$ is represented by the picture



The formal translation from solutions to graphs is given below.

Definition 2.2 Let $\mathsf{graph}(\cdot)$ be a function from solutions to graphs where nodes have sites and an internal state:

(i) $\mathsf{graph}(A[u](\sigma))$ is the graph with a single node labeled A, sites in $\{1,\cdots,\mathfrak{s}_s(A)\}$, and a tuple of values. The site i is labeled with $\sigma(i)$ (i.e. the bond, if any); the j-th element of the tuple has value $u(j)$ (i.e. the j-th field value);

(ii) $\mathsf{graph}(\mathsf{S},\mathsf{S}')$ is the union graph of $\mathsf{graph}(\mathsf{S})$ and $\mathsf{graph}(\mathsf{S}')$ where sites labeled with the same name are connected by an edge, and their common name is erased.

$\mathsf{graph}(\mathsf{S})$ is called the *underlying graph* of S.

Two molecules in a solution S are *connected* if there is a path of bonds in $\mathsf{graph}(\mathsf{S})$ that connects the corresponding nodes.

The definitions of underlying graph and connectedness easily extend to pre-solutions by taking the fields and the sites that are specified. Connectedness allows us to define complexes: a complex is a bunch of connected molecules where bonds occur exactly twice. We will extensively use the graphical notation in the rest of the paper – indeed, it has been already used in the Introduction – sometimes replacing fields with colors. In particular, we will use graphs for describing reactions – see below.

Definition 2.3 Reactions of κ calculus are either *creations* C, or *destructions* D. The format of creations is

$$\begin{aligned} A_1[u_1](\sigma_1), \cdots, A_n[u_n](\sigma_n) \rightarrow\ & A_1[u_1'](\sigma_1'), \cdots, A_n[u_n'](\sigma_n'), \\ & B_1[v_1](\phi_1), \cdots, B_k[v_k](\phi_k) \end{aligned}$$

where, for every i, $\mathrm{dom}(u_i) = \mathrm{dom}(u_i')$ and $\sigma_i \le \sigma_i'$, reactants and products are proper pre-solutions, the products are connected, and v_i and ϕ_i are total. The format of destructions is

$$A_1[u_1](\sigma_1), \cdots, A_n[u_n](\sigma_n) \rightarrow A_{i_1}[u_{i_1}'](\sigma_{i_1}'), \cdots, A_{i_k}[u_{i_k}'](\sigma_{i_k}')$$

where, $i_1, \cdots, i_k$ is an ordered sequence in $[1..n]$, for every i_j, $\mathrm{dom}(u_{i_j}) = \mathrm{dom}(u_{i_j}')$ and $\sigma_{i_j} \ge \sigma_{i_j}'$, reactants and products are proper pre-solutions, the reactants are connected, and, for every $j \notin \{i_1, \cdots, i_k\}$, u_j, σ_j are total.

Creations produce new bonds between two unbound sites and/or synthesize new molecules (that must be connected to the molecules in the left-hand side). Destructions behave in the other way around [1]. We assume that reactants and products always have at least one term.

Example 2.4 We illustrate few κ calculus reactions that corresponds to biochemical reactions. We only discuss creations.

(i) The hydrogen gas is the combination of two hydrogen atoms:

$$H[1], H[1] \rightarrow\ H[1^x], H[1^x]$$

(ii) The homeotrimerization is a combination of three monomers of the same species:

$$A[1+2], A[1+2], A[1+2] \rightarrow\ A[1^x + 2^y], A[1^y + 2^z], A[1^z + 2^x]$$

(iii) As an example of synthesis, we consider Escherichia Coli that has to synthesize galactosidase (Gal) and permease (Per) when the repressor is absent (field rep of RNAp equal to 0):

$$\begin{aligned} RNAp[rep^0](s_{Gal} + s_{Per}) \rightarrow\ & RNAp[rep^1](s_{Gal}^y + s_{Per}^z), \\ & Gal[loaded^0](lac + s^y), Per(lac + s^z) \end{aligned}$$

(With an abuse of notation, here and below, identifiers are used instead of numbers for addressing fields and sites.)

[1] The terms creation and destruction have been preferred to *complexation* and *decomplexation* used in [1,4] because they have a more neutral chemical meaning.

The operational semantics of κ calculus is a *reduction semantics*, which requires a couple of standard definitions.

- The structural equivalence between solutions, noted $\equiv$, is the least equivalence satisfying the following two rules (we remind that solutions are already quotiented by associativity of "$\mathbf{,}$"):
  1. $\mathsf{S}\mathbf{,}\mathsf{T} \equiv \mathsf{T}\mathbf{,}\mathsf{S}$;
  2. $\mathsf{S} \equiv \mathsf{T}$ if there exists an injective renaming $\imath$ on bonds such that $\mathsf{S} = \imath(\mathsf{T})$.

- Let $\mathsf{S} = A_1[u_1](\sigma_1)\mathbf{,}\cdots\mathbf{,}A_n[u_n](\sigma_n)$ be a pre-solution. We say that $\mathsf{T} = A_1[u_1 + u'_1](\sigma_1 \circ \imath + \sigma'_1)\mathbf{,}\cdots\mathbf{,}A_n[u_n + u'_n](\sigma_n \circ \imath + \sigma'_n)$ is an $(\imath, u'_1, \sigma'_1, \cdots, u'_n, \sigma'_n)$-*instance* of S if $\imath$ is an injective renaming on bonds and the maps $u_i + u'_i$ and $\sigma_i \circ \imath + \sigma'_i$ are *total* with respect to the species A_i.

Using structural equivalence it is possible to identify solutions that should not be kept distinct, such as $H(b^x)\mathbf{,}H(b^x)\mathbf{,}H(b^z)\mathbf{,}H(b^z) \equiv H(b^y)\mathbf{,}H(b^k)\mathbf{,}H(b^k)\mathbf{,}H(b^y)$. We also notice that an instance may not be necessarily a proper solution. For example $A[u^0](1^y + 2^x)$ is an $([x \mapsto y], [u \mapsto 0], [2 \mapsto x])$-instance of $A(1^x)$, but it is not a proper solution (bonds occur once).

Definition 2.5 The reduction relation of the κ calculus, written $\longrightarrow$, is the least relation satisfying the rules:

(i) let $\mathsf{P} \longrightarrow \mathsf{P}'\mathbf{,}\mathsf{Q}$ be a creation and S is an $(\imath, u'_1, \sigma'_1, \cdots, u'_n, \sigma'_n)$-instance of P, S' is an $(\imath, u'_1, \sigma'_1, \cdots, u'_n, \sigma'_n)$-instance of P' and T is an $(\imath, \emptyset, \emptyset, \cdots, \emptyset, \emptyset)$-instance of Q. Then $\mathsf{S} \longrightarrow \mathsf{S}'\mathbf{,}\mathsf{T}$;

(ii) let $\mathsf{P} \longrightarrow \mathsf{Q}$ be a destruction and S is an $(\imath, u'_1, \sigma'_1, \cdots, u'_n, \sigma'_n)$-instance of P and T is an $(\imath, u'_{i_1}, \sigma'_{i_1}, \cdots, u'_{i_k}, \sigma'_{i_k})$-instance of Q. Then $\mathsf{S} \longrightarrow \mathsf{T}$;

(iii) let $\mathsf{S} \longrightarrow \mathsf{S}'$ and $(\mathtt{bonds}(\mathsf{S}') \setminus \mathtt{bonds}(\mathsf{S})) \cap \mathtt{bonds}(\mathsf{T}) = \emptyset$; then $\mathsf{S}\mathbf{,}\mathsf{T} \longrightarrow \mathsf{S}'\mathbf{,}\mathsf{T}$;

(iv) let $\mathsf{S} \equiv \mathsf{S}'$, $\mathsf{S}' \longrightarrow \mathsf{T}'$, and $\mathsf{T}' \equiv \mathsf{T}$; then $\mathsf{S} \longrightarrow \mathsf{T}$.

The definition of reduction regards reactions as *schemas*. Namely, a reaction such as $Na[e^0](1)\mathbf{,}Cl[e^0]1 \rightarrow Na[e^1](1^x)\mathbf{,}Cl[e^{-1}](1^x)$ only addresses the fields and the the sites of the reactants that are useful for the reaction. For example, it may be the case that Na retains a site to be used for other complexes (the *sodium peroxide*, for example). In this case, the rule may be applied either to $Na[e^0](1+2)$, where the site is unbound, or to $Na[e^0](1+2^z)$. In this latter case, the reaction is instantiated as the reduction:

$$Na[e^0](1+2^z)\mathbf{,}Cl[e^0](1) \longrightarrow Na[e^1](1^x+2^z)\mathbf{,}Cl[e^{-1}](1^x) \tag{1}$$

Items 3 and 4 of Definition 2.5 allow one to derive the reductions of bigger solutions, such as

$$Na[e^0](1+2^z)\mathbf{,}Cl[e^0](1)\mathbf{,}Na[e^1](1^x+2^z)\mathbf{,}Cl[e^{-1}](1^x) \tag{2}$$

Reduction (1) may be used for deriving a reduction of the first two terms of (2), however it cannot be lifted to the whole solution because the bond created in (1) clashes with a bond already present in the solution. In this case, one derives a reduction for the structural equivalent solution $Na[e^0](1+2^z)\mathbf{,}Cl[e^0](1)\mathbf{,}Na[e^1](1^y+$

$2^z)$, $Cl[e^{-1}](1^y)$ and then a reduction of (2) is got by applying the last rule of Definition 2.5. It is straightforward to verify that the check of bond-clashes and the properness of reactants and products imply that proper solutions always reduce to proper solutions.

A basic property of κ calculus (and the other calculi of the family) is *locality*: if a sub-solution reduces then the reduction may be lifted to the whole solution without any effect on the remaining part – a direct consequence of Definition 2.5. In other words, the effects of a reduction are *localized* to the parts of molecules specified in the reaction rules.

Two other calculi, similar to κ calculus, have also been studied: the $\mathtt{m}\kappa$ calculus and the $\mathtt{nano}\kappa$ calculus [2].

Definition 2.6 The $\mathtt{m}\kappa$ calculus has species and solutions as the κ calculus but

(i) bonds may occur more than twice in a solution (multi-edges are admitted: these are called *group-names* in [1]);

(ii) reactants consist of at most two terms and, as well as products, may be not proper.

The $\mathtt{nano}\kappa$ calculus has species and solutions as the κ calculus but

(i) reactants consist of at most two terms;

(ii) there is a third type of reactions, the *exchanges* E, whose format is

$$A[u](\rho)\,,B[v](\psi) \longrightarrow A[u'](\rho')\,,B[v'](\psi')$$

with either $\rho = \rho'$ and $\psi = \psi'$ or $\rho = a^x + c^y + \rho''$, $\rho' = a + c^y + \rho''$ and $\psi = b + d^y + \psi''$, $\psi' = b^x + d^y + \psi''$.

Reactions of $\mathtt{m}\kappa$ and $\mathtt{nano}\kappa$ retain a process-calculus flavour since they amount to interactions between at most two terms. However, in order to recover (at least, to some extent) the expressivity of κ, reactions are extended with two different features:

- in $\mathtt{m}\kappa$ one may write $A[1^0](1^x)\,,B[1^0](2^x) \longrightarrow A[1^1](1^x)\,,B[1^1](2^x)\,,C[0^1+1^1](1^x+2+3)$ meaning that C is created and complexed both with A and with B: the multi-edge x represents the skeleton of the complex;
- in $\mathtt{nano}\kappa$ one may write $A[1^0](1^x+2^y)\,,B[1^0](1^y+2) \longrightarrow A[1^1](1+2^y)\,,B[1^1](1^y+2^x)$ meaning that the edge x *migrates* from A to B: the other end of the edge remains untouched. We notice that exchanges never modify the connectedness of a solution.

3 The self-assembling problem

The κ calculus allows for several many molecules to react in a reaction. The *self-assembling problem* questions whether it is possible to obtain *the same behaviour*

[2] The following definitions of $\mathtt{m}\kappa$ calculus and $\mathtt{nano}\kappa$ calculus are a bit different from those in [1,2]. In particular we admit creations of several terms at once, while this was not admitted in [1] and was not considered in [2]. However, these differences are not meaningful in the rest of the paper.

with "elementary" reactions involving at most two molecules. This problem got a positive answer when the elementary reactions were those of mκ [1,3]. κ-reactions were decomposed in sequences of mκ reactions by using the following protocol:

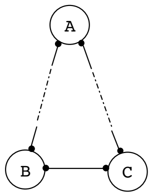
(i) *Recruitment step*: every κ reaction has a unique *spanning tree* covering its reactants; in this step all the reactants are recruited (by using ad-hoc sites in the encoded molecules). At the end of the step, all the molecules in the spanning tree share a common multi-edge.

(ii) *Later contacts step*: the spanning tree is inadequate when the left-hand sides of κ reactions are not trees, such as $A(1^x + 2^y)$, $B(1^x + 2^z)$, $C(1^y + 2^z)$. In this case, letting A be the root of the spanning tree, the protocol has to verify that B and C are connected by means of the two sites 2 *and share the same multi-edge*.

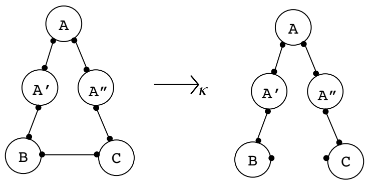
(iii) *Phase shift step*: when the left-hand side of the reaction has been completely checked, the (κ) reaction may occur and the product is generated. The κ reaction is implemented as a sequence of mκ reactions.

It is clear that every mκ reaction of steps (1) and (2) may fail; for this reason such reactions are *reversible* and the protocol has been proved to be correct with respect to *weak coupled simulation* in [1] and *weak bisimulation* in [3] (which are both unsensible to divergence).

The protocol used for self-assembling κ in mκ may be adapted to nanoκ. In particular, the steps are the same as those described in [1,3], except for the later contacts, where the spanning tree must be checked for the presence of additional bonds between the molecules therein. This case is illustrated in the following picture – called *the triangular trade* –, where A is a common parent of B and C in the spanning tree (one may take A as the unique parent at highest depth) and the bond between B and C must be verified (they have already been recruited and the protocol may fail because of the absence of such a bond).

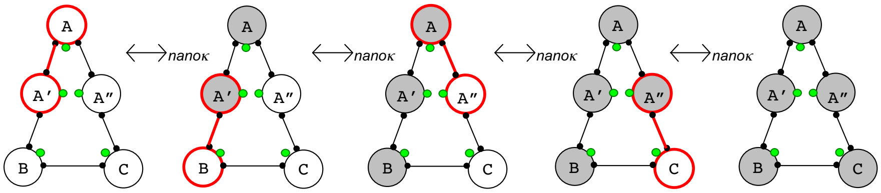


Without loss of generality, we illustrate the protocol for a κ reaction rewriting a triangular trade:

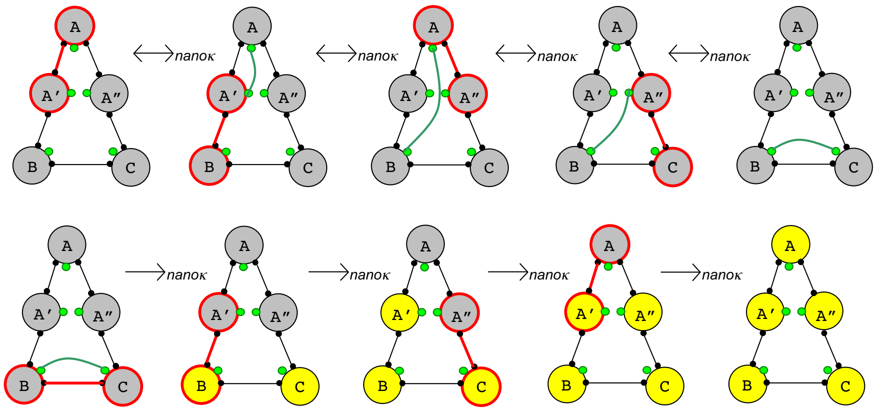


(the arrow is indexed by κ in order to avoid ambiguities). To ease the understanding, the description is pictorial. In this reaction, fields have been omitted for simplicity.

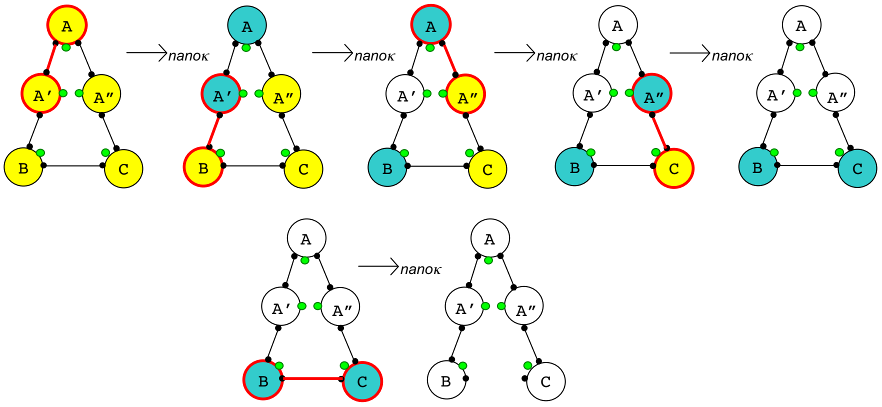
Molecules are encoded into lower-level ones having an additional site (for a bond to be exchanged) and an additional field that, in the following protocol, will store the color. Colors are used to mark the step of the protocol in the molecules. The recruitment step is the following sequence of *reversible* nanoκ reactions, where the spanning tree is assumed to be the right-hand side of the κ reaction. At the end of the step, every molecule is gray.



The later contacts step checks that B and C are actually bound each other (this may be not the case, in general, because B may be bound to a C different from the recruited one).



In order to verify that B and C are bound as required, a new edge is created by the root A and it is percolated among the nodes by means of exchange reactions till reaching the configuration depicted in the rightmost complex of the first line. Up-to now, every nanoκ reaction is reversible because the protocol may fail. On the contrary, once the double connection between B and C is verified – leftmost reaction in the second line –, the protocol cannot fail anymore and the reactions are unidirectional. The success is reported to the root A of the spanning tree by coloring the molecules in yellow. (Actually, every reaction has to be reversible when there are several triangular trades in the reactants of κ.) At this stage the phase shift step may begin and the effects of the κ reaction may be implemented. This is described by the following reactions.



Following the same pattern of [1,3], it is possible to demonstrate the correctness of the self-assembling protocol in nanoκ calculus. The formal proof is omitted.

4 Divergence and determinism

The self-assembling protocols proposed in [1,3] and in the previous section are divergent: the protocols backtrack in case of failures that may happen in the recruitment or the later contacts steps. The combination of forward and backward computations produce (infinite) loops. The questionable issue is whether a deterministic, not-divergent protocol encoding κ into a calculus with binary interactions does exist or not.

We remark that, the self-assembling protocol $[\![\cdot]\!]$ must encode both a solution – the initial one – and a set of κ reactions. Following Palamidessi [6], let $[\![\cdot]\!]$ be *uniform* if

- it is homomorphic with respect to "**,**", namely $[\![\mathsf{S},\mathsf{T}]\!] = [\![\mathsf{S}]\!],[\![\mathsf{T}]\!]$;
- it is renaming preserving, namely for every injective renaming $\imath$ on bonds of S there exists an injective renaming $\jmath$ such that $[\![\imath(\mathsf{S})]\!] = \jmath([\![\mathsf{S}]\!])$.

and be *semantically reasonable* if

- it preserves the relevant observables and the termination properties.

Uniformity guarantees that the degree of distribution of the solution is maintained by the encoding, i.e. no coordinator is added, and that the encoding does not depend on bonds. It is worth to notice that, in our case, $[\![\cdot]\!]$ might introduce new fields and new sites in the nanoκ molecules (called low-level fields and sites in the following). In addition, $[\![\cdot]\!]$ must redefine κ reactions in order to fit with the new schemas of nanoκ. We therefore extend Palamidessi's notion of uniformity of $[\![\cdot]\!]$ with the following requirements:

– for every κ reaction $\mathsf{L} \to \mathsf{R}$, $[\![\mathsf{L} \to \mathsf{R}]\!] = \{\mathsf{L}_1 \to \mathsf{R}_1, \cdots, \mathsf{L}_m \to \mathsf{R}_m\}$, where $\mathsf{L}_i \to \mathsf{R}_i$ are nanoκ reactions;

– (this is for simplicity) $[\![A[u](\sigma)]\!] = A[[\![u]\!] + v]([\![\sigma]\!] + \rho)$, that is $[\![\cdot]\!]$ preserves the granularity but may augment fields and sites. ($[\![u]\!]$ and $[\![\sigma]\!]$ may also have larger domains than u and σ, respectively.)

As regards the semantics reasonableness, in our setting the "relevant observables" are the complexes. The following equivalence equates two solutions if they possess the same complexes.

Definition 4.1 S and T are *equivalent*, in notation $\mathsf{S} \approx \mathsf{T}$, if there exists a bijection f from nodes of $\mathsf{graph}(\mathsf{S})$ to nodes of $\mathsf{graph}(\mathsf{T})$ that preserves the species and such that $A[u](\sigma)$ and $B[v](\rho)$ are connected if and only if $f(A[u](\sigma))$ and $f(B[v](\rho))$ are connected.

Notwithstanding the above revisions of Palamidessi's requirements, they turn out to be insufficient to exclude odd self-assembling protocols. In fact, our case is different than the one discussed in [6] where the dynamics of the calculi were fixed (those of pi calculus). In particular, the self-assembling protocol might completely redefine the dynamics of the encoded solution by tailoring the low-level reactions to the particular problem one wants to solve. For example, one might encode a κ-reaction by grabbing the reactants into one big molecule – that is, changing the degree of distribution – and then yielding the products – that is, re-establishing the degree of distribution. However, these encodings cannot be considered reasonable as much as maps that do not match Palamidessi's requirement of homomorphism.

Definition 4.2 Let $[\![\cdot]\!]$ be an homomorphic encoding of (pre-)solutions and reactions in κ into (pre-)solutions and reactions in $\mathtt{nano}\kappa$. The encoding $[\![\cdot]\!]$ is *twinned* if, for every $\mathsf{L} \rightarrow \mathsf{R}$

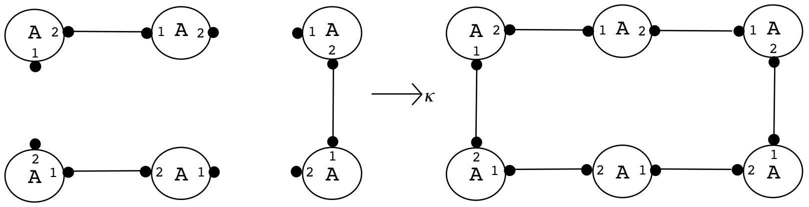
- if it is a creation and $[\![\mathsf{L} \rightarrow \mathsf{R}]\!]$ contains a $\mathtt{nano}\kappa$ destruction $\mathsf{L}' \rightarrow \mathsf{R}'$ then it also contains a *twin* creation $\mathsf{R}' \rightarrow \mathsf{L}''$ such that L' and L'' only differ for the values of fields;
- if it is a destruction and $[\![\mathsf{L} \rightarrow \mathsf{R}]\!]$ contains a $\mathtt{nano}\kappa$ creation $\mathsf{L}' \rightarrow \mathsf{R}'$ then it also contains a *twin* destruction $\mathsf{R}' \rightarrow \mathsf{L}''$ such that L' and L'' only differ for the values of fields.

Twinning guarantees that the self-assembling may undo some previous operation. The circularity may be avoided by yielding pre-solutions with different fields. Twinning also allows to localize the effects of protocols. Let us discuss the case of a κ creation $\mathsf{L} \rightarrow \mathsf{R}$. Then every destruction in $[\![\mathsf{L} \rightarrow \mathsf{R}]\!]$ has a corresponding creation restoring the bonds in the reactants. This means that if a destruction unbinds a molecule then the released molecule may be rebound, thus yielding either the previous complex with the same state – and the computation may diverge – or a previous complex with a different state. In this latter case, if the complex is too big with respect to those in R, then, there is a computation that either diverges or retains the complex (or a larger one). This because twinning will restore the complex if it is broken at some point. Therefore, in any case, the protocol is not semantically reasonable.

Definition 4.3 Let A be a species with no field and two sites 1 and 2. An *homeotrimerization* is a κ-reaction:

$$A(1^x + 2), A(1 + 2^x), A(1^y + 2), A(1 + 2^y), A(1^z + 2), A(1 + 2^z)$$
$$\longrightarrow\ A(1^x + 2^u), A(1^v + 2^x), A(1^y + 2^w), A(1^u + 2^y), A(1^z + 2^v), A(1^w + 2^z)$$

that may be rendered as:



Theorem 4.4 *There exists no self-assembling protocol that is uniform, semantically reasonable and twinned for the homeotrimerization.*

Proof: Let S be a solution consisting of $2^m * 3$ sticks $A(1^x + 2), A(1 + 2^x)$ (we assume $m > 0$). This κ solution yields a stable solution T containing 2^m homeotrimeric complexes. By contradiction, let $[\![\cdot]\!]$ be a self-assembling protocol that satisfies the requirements of the theorem. We show that it is possible to construct an infinite derivation, thus yielding a contradiction because the S always terminates. We notice that the homeotrimerization is a creation; therefore in this case positive-direction set consists of creations and exchanges, while the negative-direction set consists of destructions and exchanges.

We analyze the protocol:

1. Initially every $[\![A(\sigma)]\!]$ may arrange itself in order to participate to the homeotrimerization. Let us call *ready stick* such arranged sticks.
2. Then two ready sticks must be bound, thus making an homeodimeric complex.

   Failure 1: Since the initial solution S consists of an even number of sticks, it is possible to obtain a solution with $2^{m-1} * 3$ homeodimeric complexes.

3. In order to avoid a deadlock, the protocol must admit bonds between two homeodimerics and then discharge one stick.

   Failure 2: The step 3 is not possible because the reaction discharging one stick is a destruction. By the twinning, the protocol must also admit a creation reconnecting the two sticks. Therefore, either one obtains the previous complex, thus yielding a divergent computation, or one obtain a complex with same bonds but different fields. So we are again in case 3. Eventually one gets either a solution with a complex of two homeodimerics and no destruction is possible, therefore it is not equivalent to $[\![\mathsf{T}]\!]$, or a divergent computation.

4. It remains the possibility for an homeodimeric to break the bond created in 2, thus releasing two sticks and breaking the symmetries (changing the fields). That is, while rolling back to 1, it is possible to mark the two sticks in "winner"

and "loser", respectively. It is possible to obtain a solution where half sticks are marked as "winners" and half sticks are marked as "losers". Reactions of losers are frozen (otherwise no symmetry is broken). Then it is possible to build homeodimeric of winners and use losers to build homeotrimerics.

Failure 3: It is possible to obtain a solution where a quarter of initial sticks is frozen (because they are losers). Therefore the protocol, in order to be semantically reasonable, must admit interactions between losers that yield homeodimerics. But this is not possible because they might be also performed before (losers do not know what happens in the context – the locality principle of the κ family), when homeodimerics of winners are built (and obtaining again a solution like 2). □

Our result has rather negative consequences. One for all is the impossibility of implementing a stochastic version of κ in `nano`κ (or pi calculus) by preserving the distribution of rates (see [2]). This means that stochastic simulations must be done directly in κ [7].

References

[1] V. Danos, C. Laneve, Formal molecular biology, Theoretical Computer Science 325 (1) (2004) 69–110.

[2] A. Credi, M. Garavelli, C. Laneve, S. Pradalier, S. Silvi, G. Zavattaro, Modelization and simulation of nano devices in nanok calculus, in: Computational Methods in Systems Biology, Vol. 4695 of Lecture Notes in Computer Science, Springer, 2007, pp. 168–183.

[3] P.-L. Curien, V. Danos, J. Krivine, M. Zhang, Computational self-assembly (2007).

[4] C. Laneve, F. Tarissan, A simple calculus for proteins and cells, Electr. Notes Theor. Comput. Sci. 171 (2) (2007) 139–154.

[5] V. Danos, C. Laneve, Graphs for core molecular biology, in: Computational Methods in Systems Biology, Vol. 2602 of Lecture Notes in Computer Science, Springer, 2003, pp. 34–46.

[6] C. Palamidessi, Comparing the expressive power of the synchronous and asynchronous pi-calculi, Mathematical Structures in Computer Science 13 (5) (2003) 685–719.

[7] V. Danos, J. Feret, W. Fontana, J. Krivine, Scalable simulation of cellular signaling networks, in: Asian Symposium on Programming Languages and Systems, APLAS, Vol. 4807 of Lecture Notes in Computer Science, 2007, pp. 139–157.