

Turing Universality of the Biochemical Ground Form

Luca Cardelli¹ Gianluigi Zavattaro² †

¹*Microsoft Research, Cambridge, UK. luca@microsoft.com*

²*Dip. Scienze dell'Informazione, Università di Bologna, Italy. zavattar@cs.unibo.it*

Received 11 February 2009, revised 10 October 2009

We explore the expressive power of languages that naturally model biochemical interactions with relative to languages that naturally model only basic chemical reactions, identifying molecular association as the basic mechanism that distinguishes the former from the latter. We use a process algebra, the Biochemical Ground Form (BGF), which extends with primitives for molecular association CGF, a process algebra proved to be equivalent to the traditional notations for describing basic chemical reactions. We first observe that, differently from CGF, BGF is Turing universal as it supports a finite precise encoding of Random Access Machines, a well-known Turing powerful formalism. Then we prove that the Turing universality of BGF derives from the interplay between the molecular primitives of *association* and *dissociation*. In fact, the elimination from BGF of the primitives already present in CGF does not reduce the computational strength of the process algebra, while if either association or dissociation is removed then BGF is no longer Turing complete.

1. Introduction

In this paper we investigate the computational strength of a process algebra that aims to capture the essential primitives of biochemistry. Biochemistry is obviously based on chemistry, and in principle one can always express the behavior of a biochemical system by a collection of chemical reactions. But there is a major practical problem with that approach: the collection of reactions for virtually all biochemical systems is an infinite one. For example, just to express the chemical reactions involved in linear polymerization which is common in biochemical systems, we need to have a different chemical species for each length n of polymer P_n , with reactions to grow the polymer: $P_n + M \rightarrow P_{n+1}$. While each polymer is finite, the set of possible polymerization reactions is infinite. Nature adopts a more modular solution: each monomer obeys a *finite* simple set of *molecular association and dissociation* rules that leads to the formation and breakup of polymers

† Contact author. Department of Computer Science, University of Bologna, Mura A. Zamboni 7, 40127, Bologna, Italy. The research of this author has been partially funded by “Progetto Strategico” ComReNDe: Compositional and executable Representations of Nano Devices.

of any length; therefore, it seems that there should be a finite way of describing such systems. One can start by writing pseudo-reactions like $P + M \rightarrow P : M$, where $P : M$ is meant to represent a P (olymer) molecule attached to an extra M (onomer), yielding a longer polymer. However, there are in general many possible ways (that is, many different patches on the surface of a molecule) by which one molecule can exclusively attach to other molecules, and soon one needs to describe the *interface* of each molecule. This situation, while not commonly found in basic chemistry, is particularly acute in biochemistry, where virtually all reactions are governed by enzymes and molecular machines, which are themselves often built by molecular association, and which usually operate by associating with their reactants.

The intuitive idea of a biomolecule as a stateful entity with a connectivity interface is now common. Notations have emerged from biology that use such an idea to describe large biochemical systems (Kohn *et al.* 2006; Kitano *et al.* 2005). Many formalized and computerized approaches are currently being developed, including: practical tools, where molecules are drawn as boxes with connecting lines (Danos *et al.* 2007); graph-rewriting and term-rewriting systems where a molecular complex is represented as a graph or term, and a reaction is a graph or term rewrite (Danos and Laneve 2004; Credi *et al.* 2007); coding techniques in process algebra, where molecular association can be expressed via some advanced features (Priami *et al.* 2001); and finally, specialized process algebras where molecular interfaces and association are taken as primitive (Priami and Quaglia 2004; Cardelli and Pradalier 2005). All these approaches aim to find a descriptive framework that goes beyond simple chemical reactions, and that can be used to represent common biochemical situations finitely and modularly.

The aim of this paper is then to determine the inherent difference in expressive power between notations for basic chemical interactions and notions for biochemical interactions. That is: what is the intrinsic power of molecular association that gives it the ability to represent finitely what would otherwise have an infinite representation? To clarify this issue we study a formal system, the Biochemical Ground Form (BGF), that we propose in this paper as a minimalistic extension of the Chemical Ground Form (Cardelli 2008) (CGF) with association and dissociation. CGF is a process algebra based on the notion of molecules as stateful entities which was proved to be equivalent to other commonly used notations for basic chemical interactions. Namely, given the description of a chemical system given in terms of a *finite stochastic chemical reaction network* (SCRN) (Soloveichik *et al.* 2008) there exists an equivalent system expressed in CGF, and vice versa.

As already mentioned, many richer formalisms can represent molecular association, but they also include mechanisms that have no direct biological implementation. Our proposal is minimalistic in the following sense: it adds only two basic actions called *association* and *dissociation*. Association allows two molecules to form a complex, dissociation allows them to subsequently break such a complex. Between these two events, the two molecules can still freely interact with other molecules or among themselves. In other more expressive formalisms (see for instance the so-called *exchange reactions* in (Credi *et al.* 2007)) it is possible to specify events that change the internal state of one molecule only if it is associated with another one having a particular state.

Forms of organized molecular association and dissociation exist in inorganic chemistry,

but certainly not to the extent in which they are found in biochemistry, where they are the foundation of molecular machines, information processing, metabolism, and membranes, both structurally and functionally. In this respect, the first contribution of this paper is the formalization of the following inherent expressiveness gap between SCRNs and notations for biochemical interactions: the CGF is not Turing complete (this result was proved in (Zavattaro and Cardelli 2008)) while its minimalistic extension BGF is already Turing complete. After proving this result, we look for minimal sets of primitives that make BGF Turing complete. The aim is to have a deeper understanding of the basic mechanisms that distinguish biochemical notations from SCRNs. We prove that all primitives that BGF inherits from CGF can be removed from the model without reducing its computational strength. On the contrary, if either the association primitive or the dissociation primitive is removed then BGF is no longer Turing complete. This allows us to conclude that both the association and the dissociation primitives are sufficient and necessary for making BGF Turing universal.

Structure of the paper

The paper is structured as follows. In Section 2 we give the definition of BGF. In Section 3 we prove that BGF is Turing complete. In Section 4 we prove that the process algebra is still Turing complete even if we remove all primitives but association and dissociation. In Sections 5 and 6 we prove that BGF is no longer Turing complete if we remove either the association or the dissociation primitives, respectively. Finally in Section 7 we give some concluding remarks.

A preliminary version of this paper appeared in (Cardelli and Zavattaro 2008). In the present paper we consider a more general version of BGF (in which also molecules inside complexes can split) and we prove new original results (those reported in the Sections 4, 5, 6).

2. Biochemical Ground Form

In this section we give the definition of the Biochemical Ground Form (BGF). We first informally introduce the notation, then we give the formal syntax and semantics.

As discussed in the Introduction, BGF is the extension of the process algebra CGF (Cardelli 2008) with association and dissociation. In this presentation of BGF we abstract away from the stochastic semantics [†] as it is not relevant for the results about Turing completeness and decidability we are going to discuss in this paper. We start presenting the features of BGF inherited from CGF.

The basic ingredient of the process algebra is the notion of chemical species. Each chemical species has an associated definition indicating the actions that the molecules belonging to the species can perform. There are three possible actions in CGF, internal actions τ , input actions $?a$, and output actions $!a$. Action τ indicates the possibility for

[†] The interested reader can refer to (Cardelli 2008) for the stochastic semantics of CGF and to (Cardelli and Zavattaro 2008) for the stochastic semantics of BGF.

a molecule to be engaged in a unary reaction. For instance, the definition $A = \tau; (B|C)$ is used to specify the possibility for one molecule of species A to be engaged in a unary reaction that produces two molecules, one of species B and one of species C (the operator “|” is borrowed from process algebras such as CCS (Milner 1989), where it represents parallel composition, and corresponds here to the chemical “+”). Binary reactions have two reactants. The two reactants perform two complementary actions: input $?a$ and output $!a$, where a is a name used to identify the reaction. For instance, given the definitions $A = ?a; C$ and $B = !a; D$, we have that two molecules of species A and B can be engaged in a binary reaction that produces two molecules, one of species C and one of species D . If the molecules of one species can be engaged in several reactions, then the corresponding definition admits a choice among several actions. The syntax of choice is as follows: $A = \tau; B \oplus ?a; C$, meaning that molecules of species A can be engaged in either a unary reaction that produces a molecule of species B , or in a binary reaction with another molecule able to execute the complementary action $!a$. In the second case, the molecule of species A contributes to the reaction by producing a new molecule of species C .

Example 1 (Two-stations rotaxane). We consider two-stations rotaxanes (Sauvage and Dietrich-Buchecker 1999) (simply called rotaxanes in the following), which are supramolecular systems composed of an axle surrounded by a ring-type molecule. Bulky chemical moieties (“stoppers”) are placed at the extremities of the axle to prevent the disassembly of the system. In rotaxanes containing two different recognition sites on the axle (“stations”), it is possible to switch the position of the ring between the two stations by an external energy input (called the “stimulus”) as illustrated in Figure 1. Upon stimulation, the energy curve is modified in such a way that the equilibrium for the ring changes.

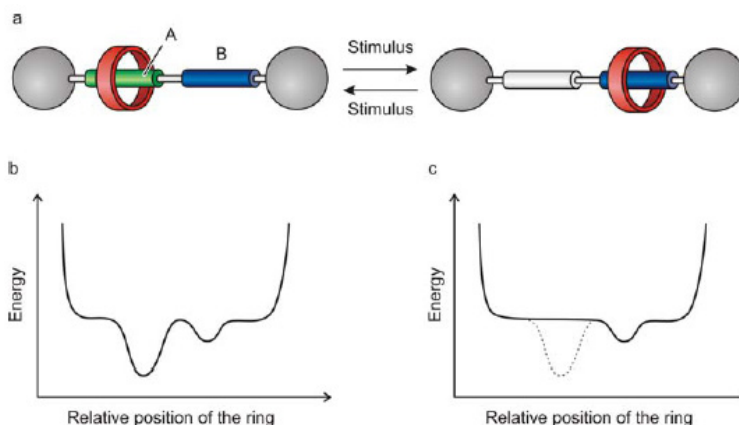


Fig. 1. Representation of a rotaxane with stations A and B (a) and its energy curves before (b) and after (c) the stimulus activating the ring movement from A to B .

According to the approach described above, we can model the behavior of a rotaxane with two stations, namely A and B , which interacts with molecules of species S that

stimulate the movement of the ring. We consider four distinct species: R^A (resp. R^B) representing the *nonstimulated* rotaxane with the ring in position A (resp. B), and R_s^A (resp. R_s^B) representing the *stimulated* rotaxane with the ring in position A (resp. B).

We can consider the following definitions for these species:

$$\begin{aligned} R^A &= ?a_to_b; R_s^A \\ R^B &= ?b_to_a; R_s^B \\ R_s^A &= \tau; R^B \\ R_s^B &= \tau; R^A \\ S &= !a_to_b; \mathbf{0} \oplus !b_to_a; \mathbf{0} \end{aligned}$$

where $\mathbf{0}$ specifies the fact that there is no product (i.e. the stimulus degrades after the reaction with the rotaxane).

We now move to association and dissociation, the new features of BGF with respect to its predecessor CGF, which are modeled by two new pairs of complementary actions: $\&?a, \&!a$ for association and $\%?a, \%!a$ for dissociation. We now introduce them informally by means of examples.

Example 2 (Linearly growing polymer). Each association event involves exactly two partners. We imagine that the partners have two complementary surface patches that can interlock. If c represents a surface shape (say, a paraboloid), then $!c$ indicates one of the two patches (say, the convex one) and $?c$ indicates the complementary patch (the concave one). Then, $\&!c$ is the action that presents the convex patch, and $\&?c$ is the action that presents the concave patch. When two such *association* actions meet, an actual association event can take place, joining the two complementary surfaces.

A linearly growing polymer could be represented as follows, using a seed S and a collection of equal monomers M . The seed starts the chain by presenting a concave patch $?c$: this is our initial, zero-length, polymer. Each monomer presents a convex patch $!c$, which can bind with an existing polymer on the complementary concave patch. After (and only after) such a binding, a bound monomer M' presents another concave patch $?c$, so that the polymer can keep growing. Both the seed and each monomer can have further behavior, S' and M'' .

$$\begin{aligned} S &= \&?c; S' \\ M &= \&!c; M' \\ M' &= \&?c; M'' \end{aligned}$$

Each association event creates a unique bond between exactly the two molecules that are joined to each other. This bond needs to be represented somehow, to make sure that a molecule can bind with only one other molecule at a time on any given patch. We represent such a bond as a unique key k that is shared by the two molecules (think of k as a fresh number, or as a fresh channel in π -calculus (Milner 1989)). Such unique keys, and related information, are collected in the *association history* of each molecule. So, the first interaction of an S with an M , which initially have empty association histories ($\mathbf{0}$), proceeds as follows:

$$S_{\mathbf{0}} \mid M_{\mathbf{0}} \rightarrow S'_{\langle ?c, k1 \rangle} \mid M'_{\langle !c, k1 \rangle}$$

Interaction with a second monomer then introduces a second fresh key in the histories:

$$S_0 \mid M_0 \mid M_0 \rightarrow S'_{\langle ?c, k1 \rangle} \mid M'_{\langle !c, k1 \rangle} \mid M_0 \rightarrow S'_{\langle ?c, k1 \rangle} \mid M''_{\langle ?c, k2 \rangle :: \langle !c, k1 \rangle} \mid M'_{\langle !c, k2 \rangle}$$

This mechanism of creation of fresh association keys is repeated every time a new association is created between a monomer and the subsequent one.

It is worth observing that, in any reachable configuration, we can reconstruct from the association histories who is bound to whom, and on what surface the bond was formed. Note that the description of the system is finite (3 reagents, S , M , M'), but that polymers of any length can be assembled (assuming the initial availability of a corresponding amount of monomers). Moreover, as associations cannot occur on surfaces that are already occupied, we have that association histories are bounded; namely, the maximal length of an association history is given by the statically defined number of surfaces of the molecule which the association history belongs to.

Example 3 (Branching polymer). After association, a molecule is still free to perform additional associations or other interactions. That is, association places no restrictions on the behavior of the original molecules, except for the fact that new associations cannot occur on surfaces that are already occupied, and that dissociations must happen consistently with prior associations (as we discuss shortly). To illustrate this freedom, let us modify the previous example and allow each bound monomer to offer a seed for growing a new polymer branch:

$$\begin{aligned} S &= \&?c; S' \\ M &= \&!c; M' \\ M' &= \&?c; M'' \\ M'' &= \&?d; M''' \\ N &= \&!d; N' \\ N' &= \&?c; N'' \end{aligned}$$

Where an M'' can bind through the interface d to an adaptor molecule N , which then offers another c surface for branching.

Example 4 (Actin-like polymer). *Dissociation* is the inverse of association, that is, two formerly joined molecules can separate. We indicate by $\%!c$ the attempt to dissociate from the convex side, and $\%?c$ the attempt to dissociate from the concave side. When two associated molecules attempt complementary dissociations, an actual dissociation event can take place. To illustrate this situation, we describe a different kind of linear polymer: one that can grow only at one end, and can shrink only at the other end. There are four molecular states for each monomer: M^f (free monomer), M^l (monomer bound on the left), M^r (monomer bound on the right), and M^b (monomer bound on both sides). Each monomer has a left convex surface and a complementary right concave surface. A polymer should associate (grow) only on the right and should dissociate (shrink) only on

the left.

$$\begin{aligned}
 M^f &= \&!c; M^l \oplus \&?c; M^r \\
 M^l &= \%!c; M^f \oplus \&?c; M^b \\
 M^r &= \%?c; M^f \\
 M^b &= \%!c; M^r
 \end{aligned}$$

A free monomer M^f can either associate on the left convex surface and become bound on the left, or associate on the right concave surface and become bound on the right. A monomer M^l bound only on the left can either dissociate on the left (if allowed by its partner, which must in fact be an M^r in this case) and return free, or associate on the right (with an M^f) and become bound on both sides. A monomer M^r bound only on the right can only dissociate on the right: that is, a polymer cannot grow on the left. A monomer M^b bound on both sides can only dissociate on the left (with an M^r): that is, a polymer cannot shrink on the right or break in the middle. These rules cover also the base cases when a polymer of length 2 initially forms or finally dissolves.

A dissociation should succeed only between a pair of molecules that were actually associated in their past history, and this can be checked by inspecting the unique keys introduced during association. For example let us consider two M^f molecules that associate and then immediately dissociate:

$$M_0^f | M_0^f \rightarrow M_{\langle !c, k \rangle}^l | M_{\langle ?c, k \rangle}^r \rightarrow M_0^f | M_0^f$$

The second transition is allowed to happen because M^l offers $\&!c$, M^r offers the complementary $\&?c$, and the same key k appears in both association histories on the c interface (and with the correct convexity). As a consequence of dissociation, the keys are removed from the histories.

Example 5 (Unbounded linearly growing and shrinking polymer). Recursive definitions of the species behavior allows us to specify systems in which an unbounded number of monomers can be created. We use this ability to specify a linearly growing polymer started by a seed, that can also shrink removing the last associated monomer, and for which there is no fixed maximal length. In order to produce an unbounded number of monomers we consider a *factory* species able to continuously produce monomers:

$$\begin{aligned}
 Fact &= \tau; (M^f | Fact) \\
 S &= \&?c; S' \\
 S' &= \%?c; S \\
 M^f &= \&!c; M^l \\
 M^l &= \%!c; M^f \oplus \&?c; M^b \\
 M^b &= \%?c; M^l
 \end{aligned}$$

It is easy to see that each seed molecule of species S has the ability to start the creation of a polymer that can grow and shrink along one direction without any fixed bound to its maximal length. We will exploit this technique in the proof of Turing completeness of the biochemical ground form in order to model registers, i.e., data structures on which increment, decrement and test for zero operations can be executed. The intuition is that increments are modeled by means of the creation and association of a new monomer,

decrements by means of the elimination of the last associated monomer, and test for zero simply by checking the availability of a molecule of species S (the seed becomes of species S' when associated to a monomer).

Almost all new ingredients of the BGF have been presented in the examples above. The unique additional aspect that requires discussion deals with *molecule splitting*, that is the possibility for one reactant to produce more than one molecule. Differently from a previous presentation of the process algebra (Cardelli and Zavattaro 2008), we admit the splitting of molecules even when they are associated with other molecules. To this aim we need to extend the language to allow for the specification of the distribution of the associations among the produced molecules. This is obtained by attaching *association markers* to the produced molecules. These specify which associations in the association history of the splitting molecule are inherited by each newly produced molecule.

Example 6 (Breaking polymer). To illustrate association in combination with molecule splitting, we describe a linearly growing polymer similar to the actin-like polymer of the Example 4 in which each monomer, once bound on both sides, is free to split into two new monomers each one inheriting one of the two bonds. The definition is as follows:

$$\begin{aligned} M^f &= \&!c; M^l \oplus \&?c; M^r \\ M^l &= \%!c; M^f \oplus \&?c; M^b \\ M^r &= \%?c; M^f \\ M^b &= \%!c; M^r \oplus \tau; (M_{!c}^l | M_{?c}^r) \end{aligned}$$

It is worth observing that in case of splitting of the molecules of species M^b , it is necessary to indicate also how to split the associations among the two produced molecules of species M^l and M^r , respectively. This is obtained by adding the association marker $!c$ to M^l and $?c$ to M^r corresponding to the bonds to be split.

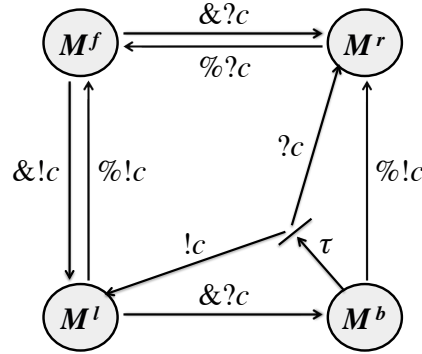


Fig. 2. Graphical representation of the breaking polymer.

In this last example we also introduce a graphical notation that allows us to depict the behavior of the species using an automata like representation. The idea is to represent each species with a state of the automaton. A transition from the species X to the species

Y labeled with the action π means that the definition of X includes the summand $\pi; Y$. In case the species X can split, i.e. there is a summand $\pi; (Y_{h_1}^1 | \dots | Y_{h_n}^n)$ with $n > 1$ in its definition, we consider a transition labeled with π from the state X to an intermediary state represented by a bar, and one transition for each $Y_{h_i}^i$ with the bar as source, the state Y^i as target, and h_i as label. The graphical representation of the breaking polymer is in Figure 2.

We are now ready to present the definition of the syntax of BGF.

Definition 1 (Biochemical Ground Form (BGF)). Consider the following denumerable sets: *Species* ranged over by X, Y, X^1, X^2, \dots , *Channels* ranged over by a, b, \dots , and the totally ordered set of *Associations* ranged over by k, k', \dots .

The syntax of BGF is as follows:

$E ::= \mathbf{0} \mid X = M, E$	Reagents
$M ::= \mathbf{0} \mid \pi; P \oplus M$	Molecule
$\pi ::= \tau \mid ?a \mid !a$	Internal, Input, Output prefix
$\quad \mid \&?a \mid \&!a$	Association prefixes
$\quad \mid \%?a \mid \%!a$	Dissociation prefixes
$P ::= \mathbf{0} \mid X_h P$	Product
$h ::= \mathbf{0} \mid ?a :: h \mid !a :: h$	Association markers
$S ::= \mathbf{0} \mid X_H S$	Solution
$H ::= \mathbf{0} \mid \langle ?a, k \rangle :: H \mid \langle !a, k \rangle :: H$	Association history
$BGF ::= (E, S)$	Biochemical Ground Form

Given a BGF (E, S) , we assume that all variables occurring in S occur also in E . Moreover, for every variable X occurring in E , there is exactly one definition $X = M$ in E .

In the following, trailing $\mathbf{0}$ are usually omitted: for instance, we denote $\langle ?a, k \rangle :: \mathbf{0}$ simply with $\langle ?a, k \rangle$. Moreover, we consider $|$ and $::$ also as associative and commutative operators over the syntax: for instance, if H and H' are $\mathbf{0}$ -terminated association histories, according to the syntax above, then $H :: H'$ means appending the two lists into a single $\mathbf{0}$ -terminated list. Therefore, $\mathbf{0} :: H$, $H :: \mathbf{0}$, and H are syntactically equal. Finally, by commutativity of $::$, we also consider $H :: H'$ and $H' :: H$ syntactically equal. The same holds also for $|$, that is, given the species X and Y , the solutions $X_H | Y_{H'} | \mathbf{0}$, $Y_{H'} | X_H | \mathbf{0}$, $X_H | Y_{H'}$, and $Y_{H'} | X_H$ are syntactically equal.

Every BGF (E, S) must satisfy the following two constraints:

- 1 given a molecule X_H in S , each symbol $?a$ or $!a$ occurs in H at most once;
- 2 given $\langle ?a, k \rangle$ (resp. $\langle !a, k \rangle$) occurring in the association history H of one molecule X_H in S , there exists exactly another one occurrence of k in S , which appears in a complementary $\langle !a, k \rangle$ (resp. $\langle ?a, k \rangle$) stored in the association history H' of a distinct molecule $X'_{H'}$;
- 3 if a molecule X_H occurs in S , and in the definition of X in E there is a summand $\pi; (X_{h_1}^1 | \dots | X_{h_n}^n)$, the association markers h_1, \dots, h_n should contain, without repetition, at least all the symbols $?a$ and $!a$ occurring in H .

The first and the second constraints guarantee that one surface of one molecule, denoted with $?a$ (resp. $!a$), is either not associated or associated to exactly another one complementary surface $!a$ (resp. $?a$) of a distinct molecule.[‡] The third constraint, on the other hand, guarantees that the association markers that are attached to the molecules in a product, correctly specify the redistribution of the associations from the reactants to the products.

In order to guarantee that all the solutions in the process algebra actually satisfy the above constraints, we restrict only to *well formed* BGF. By definition, a well formed BGF (E, S) satisfies the three constraints. Moreover, we will prove (see Proposition 2) that well formedness is preserved by the transitions in the transition graph of a BGF. Thus, given a solution T reachable from a well formed BGF (E, S) , we have that also (E, T) is well formed, thus it satisfies the three constraints.

Before presenting the definition of well formed BGF, it is necessary to observe that, as the BGF is Turing complete (the proof is in Section 3), it is in general impossible to compute the exact set of associations of the molecules belonging to a given species. This information is necessary in order to check the third constraint according to which, when a molecule splits, the association markers should consider all the associations of the splitting molecule. In the light of this impossibility result, in the definition of well formed BGF, we consider for each species X an over-approximation of the set of its possible associations, that we denote with $\Gamma(X)$.

Definition 2 (Well formed BGF). Given the association history H (resp. the association marker h) we denote with $\mathbf{set}(H)$ (resp. $\mathbf{set}(h)$) the set of symbols $?a$ and $!a$ occurring in H (resp. h).

Let us consider a biochemical ground form (E, S) . We say that a function Γ , mapping each species X occurring in E to symbols in $\{?a, !a \mid a \text{ occurs in } E \text{ or } S\}$ is an *association over-approximation* for (E, S) , written $\Gamma \vdash (E, S)$, if it is possible to derive $\Gamma \vdash (E, S)$ in the following proof system:

— $\Gamma \vdash (\mathbf{0}, \mathbf{0})$

— if $\Gamma \vdash (\mathbf{0}, S)$ and $\mathbf{set}(H) \subseteq \Gamma(X)$ then

$$\Gamma \vdash (\mathbf{0}, S | X_H)$$

— if $\Gamma \vdash (E, S)$ and X is not already defined in E then

$$\Gamma \vdash (X = \mathbf{0}, E, S)$$

— if $\Gamma \vdash (X = M, E, S)$, $\sigma \in \{?a, !a \mid a \in \mathit{Channels}\} \cup \{\tau\}$, $\Gamma(X) \subseteq \Gamma(X^1) \cup \dots \cup \Gamma(X^n)$ and $\mathbf{set}(h_i) \cap \Gamma(X) \subseteq \Gamma(X^i)$ for each $0 \leq i \leq n$ then

$$\Gamma \vdash (X = \sigma; (X_{h_1}^1 | \dots | X_{h_n}^n) \oplus M, E, S)$$

— if $\Gamma \vdash (X = M, E, S)$, $\sigma \in \{?a, !a \mid a \in \mathit{Channels}\}$, $\Gamma(X) \cup \{\sigma\} \subseteq \Gamma(X^1) \cup \dots \cup \Gamma(X^n)$

[‡] In BGF, we do not admit self-complexation, i.e., the possibility for one molecule to associate with itself.

and $\mathbf{set}(h_i) \cap (\Gamma(X) \cup \{\sigma\}) \subseteq \Gamma(X^i)$ for each $0 \leq i \leq n$ then

$$\Gamma \vdash (X = \&\sigma; (X_{h_1}^1 | \dots | X_{h_n}^n) \oplus M, E, S)$$

— if $\Gamma \vdash (X = M, E, S)$, $\sigma \in \{?a, !a \mid a \in \mathit{Channels}\}$, $\Gamma(X) \setminus \{\sigma\} \subseteq \Gamma(X^1) \cup \dots \cup \Gamma(X^n)$ and $\mathbf{set}(h_i) \cap (\Gamma(X) \setminus \{\sigma\}) \subseteq \Gamma(X^i)$ or each $0 \leq i \leq n$ then

$$\Gamma \vdash (X = \%_0\sigma; (X_{h_1}^1 | \dots | X_{h_n}^n) \oplus M, E, S)$$

A BGF (E, S) is well formed if

- given X_H in S , each symbol $?a$ or $!a$ occurs in H at most once and
- an association k either does not occur in S or it occurs in exactly two pairs $\langle ?a, k \rangle$ and $\langle !a, k \rangle$ stored in the association history of two distinct molecules and
- there exists an association over-approximation Γ such that $\Gamma \vdash (E, S)$ and for each species X in E the following holds:
 - if $X = \dots \oplus \pi; (X_{h_1}^1 | \dots | X_{h_n}^n) \oplus \dots$ with $n > 1$ then $\mathbf{set}(h_i) \cap \mathbf{set}(h_j) = \emptyset$ for each $1 \leq i < j \leq n$ and $\Gamma(X) \subseteq \mathbf{set}(h_1) \cup \dots \cup \mathbf{set}(h_n)$, moreover if $\pi = \&\sigma$ with $\sigma = ?a$ or $\sigma = !a$ then also $\sigma \in \mathbf{set}(h_1) \cup \dots \cup \mathbf{set}(h_n)$;
 - if $X = \dots \oplus \pi; \mathbf{0} \oplus \dots$ then $\Gamma(X) = \emptyset$ and $\pi \neq \&\sigma$ for any $\sigma = ?a$ or $\sigma = !a$.

It is worth observing that the existence of an association over-approximation Γ satisfying the property reported in the last item of the definition is decidable. In fact, there exist only a finite number of such functions to be checked. This holds because the association over-approximations to be considered are defined on a finite domain (the set of species X occurring in E) and a finite range (the set of symbols $?a$ and $!a$ occurring in E or S).

The operational semantics of a BGF is defined in terms of a Transition Graph (TG), that is a binary relation on solutions denoted with $S \rightarrow T$.

Definition 3 (Semantics of a Biochemical Ground Form). Given the product $P = X_{h_1}^1 | \dots | X_{h_n}^n$ with $n > 1$ and the association history H , with $P \downarrow H$ we denote the solution $X_{H \downarrow h_1}^1 | \dots | X_{H \downarrow h_n}^n$ where $H \downarrow h$ is the projection of the association history H to the associations with a marker in h . On the other hand, if $P = X_h$ then $P \downarrow H$ is X_H while if $P = \mathbf{0}$ then $P \downarrow H$ is $\mathbf{0}$. With $H \setminus \langle ?a, k \rangle$ (resp. $H \setminus \langle !a, k \rangle$) we denote the history achieved by removing from H the association $\langle ?a, k \rangle$ (resp. $\langle !a, k \rangle$).

Given the BGF (E, S) , we define $\mathit{Next}(E, S)$ as the set containing the following kinds of transitions:

Unary: if $S = X_H | S'$ and $X = \dots \oplus \tau; P \oplus \dots$ in E then

$$X_H | S' \rightarrow P \downarrow H | S'$$

Binary: if $S = X_H | X_{H'}' | S'$ and $X = \dots \oplus ?a; P \oplus \dots$ and $X' = \dots \oplus !a; P' \oplus \dots$ in E then

$$X_H | X_{H'}' | S' \rightarrow P \downarrow H | P' \downarrow H' | S'$$

Association: if $S = X_H | X_{H'}' | S'$, $X = \dots \oplus \&?a; P \oplus \dots$ and $X' = \dots \oplus \&!a; P' \oplus \dots$ in E , $?a$ and $!a$ do not appear in the association histories H and H' respectively, and k is the smallest association key among those that do not appear in the association

histories in S then

$$X_H \mid X'_{H'} \mid S' \rightarrow P\downarrow(H::\langle ?a, k \rangle) \mid P'\downarrow(H'::\langle !a, k \rangle) \mid S'$$

Dissociation: if $S = X_H \mid X'_{H'} \mid S'$, $X = \dots \oplus \%?a$; $P \oplus \dots$ and $X' = \dots \oplus \%!a$; $P' \oplus \dots$ in E , and $\langle ?a, k \rangle$ and $\langle !a, k \rangle$ appear in the association histories H and H' respectively, then

$$X_H \mid X'_{H'} \mid S' \rightarrow P\downarrow(H \setminus \langle ?a, k \rangle) \mid P'\downarrow(H' \setminus \langle !a, k \rangle) \mid S'$$

The Transition Graph of (E, S) is defined as follows:

$$TG(E, S) = \bigcup_n \Psi_n$$

where $\Psi_0 = Next(E, S)$ and $\Psi_{n+1} = \bigcup \{Next(E, Q) \mid Q \text{ is a solution of } \Psi_n\}$

In order to prove that all the solutions T reachable from a well formed BGF (E, S) are well formed according to the definitions in E (i.e. (E, T) is well formed) we first need to show that any association over-approximation for S is valid also for T .

Proposition 1. Let (E, S) be a well formed BGF. If $\Gamma \vdash (E, S)$ and $S \rightarrow T$ is a transition in $Next(E, S)$, then also $\Gamma \vdash (E, T)$.

Proof. The proof tree of $\Gamma \vdash (E, S)$ according to the proof system in the Definition 2 is divided in two parts: the first part proves that $\Gamma \vdash (\mathbf{0}, S)$ (and checks that $\mathbf{set}(H) \subseteq \Gamma(X)$ for each X_H in S), while the second part deals with the definitions in E (and can be used to prove that $\Gamma \vdash (E, W)$ for any solution W such that $\Gamma \vdash (\mathbf{0}, W)$). Assuming that $S \rightarrow T$ is in $Next(E, S)$, we show the existence of a proof tree also for $\Gamma \vdash (E, T)$.

The first part of this proof tree, proving that $\Gamma \vdash (\mathbf{0}, T)$, checks that $\mathbf{set}(H) \subseteq \Gamma(X)$ for each molecule X_H occurring in T . For the molecules already in S this condition was already checked (by the proof tree of $\Gamma \vdash (\mathbf{0}, S)$). For the molecules produced by the transition $S \rightarrow T$ we proceed by case analysis considering the four possible kinds of transition. We detail only the analysis for the *unary* case, as the other cases are treated similarly.

Assume that $S \rightarrow T$ is a unary transition that modifies the molecule X_H according to a summand $\tau; (X_{h_1}^1 \mid \dots \mid X_{h_n}^n)$ in the definition of X . In this case the new molecules are $X_{H \downarrow h_1}^1 \mid \dots \mid X_{H \downarrow h_n}^n$. For each of these molecules $X_{H_i}^i$ (where $H_i = H \downarrow h_i$) we have that $\mathbf{set}(H_i) = \mathbf{set}(H) \cap \mathbf{set}(h_i)$, thus we must prove that $\mathbf{set}(H) \cap \mathbf{set}(h_i) \subseteq \Gamma(X^i)$. As observed above, we have that $\mathbf{set}(H) \subseteq \Gamma(X)$, thus also $\mathbf{set}(H) \cap \mathbf{set}(h_i) \subseteq \Gamma(X) \cap \mathbf{set}(h_i)$. Due to the summand $\tau; (X_{h_1}^1 \mid \dots \mid X_{h_n}^n)$ in the definition of X in E , and because $\Gamma \vdash (E, S)$, we have that the fourth rule of the proof system in Definition 2 guarantees that $\Gamma(X) \cap \mathbf{set}(h_i) \subseteq \Gamma(X^i)$. This concludes the analysis of this case.

We have shown that $\Gamma \vdash (\mathbf{0}, T)$. As discussed above, the second part of the proof tree of $\Gamma \vdash (E, S)$ can be used to prove that also $\Gamma \vdash (E, T)$. \square

We can now show that, given a well formed BGF (E, S) and a transition $S \rightarrow T$ in $Next(E, S)$, then also (E, T) is well formed.

Proposition 2. Let (E, S) be a well formed BGF. If $S \rightarrow T$ is a transition in $Next(E, S)$, then also (E, T) is a well formed BGF.

Proof. Consider a well formed BGF (E, S) and a transition $S \rightarrow T$ in $Next(E, S)$. In order to prove that also the BGF (E, T) is well formed, we need to show that the three conditions in the Definition 2 are satisfied by (E, T) .

As far as the first two conditions are concerned, a trivial case analysis on the possible kinds of the transition $S \rightarrow T$ allows us to conclude that, as S satisfies these conditions, the same holds also for T . In particular, the first condition (i.e. no symbol $?a$ or $!a$ appears twice in the same association history) holds as a consequence of the condition “ $?a$ and $!a$ do not appear in the association histories H and H' respectively” considered for *association* transitions. On the other hand, the second condition (i.e. an association key occurs exactly in two complementary associations of two distinct molecules) holds as a consequence of the freshness of k in *association* transitions, by the fact that a new instance of keys is always located in complementary associations, and by the fact that already available instances of keys are always removed in pairs by *dissociation* transitions.

We now consider the third condition. This condition checks the existence of an association over-approximation which has a precise relationship with the definitions in E . As (E, S) is well formed, the existence of at least one of such association over-approximation is guaranteed. Let Γ be such association over-approximation such that $\Gamma \vdash (E, S)$. By Proposition 1 we have that also $\Gamma \vdash (E, T)$, thus we can conclude that also the third condition is satisfied by (E, T) . \square

As a trivial corollary we have that, given a well formed BGF (E, S) and any reachable solution T in $TG(E, S)$, then also (E, T) is well formed.

Corollary 1. Let (E, S) be a well formed BGF. If T is a solution in $TG(E, S)$, then also (E, T) is a well formed BGF.

Proof. Assume that T is a solution in $TG(E, S)$ where (E, S) is a well-formed BGF. By definition of transition graph we have that if there exists a natural number i such that T can be reached after i transitions starting from the initial solution S . The thesis is proved applying i times the Proposition 2. \square

3. Turing completeness of BGF

We prove that the Biochemical Ground Form is Turing complete. In the literature, several notions of Turing completeness for process algebras can be found, see e.g. (Maffeis and Phillips 2005) for a classification of these criteria. We now report three (variants) of the criteria presented in (Maffeis and Phillips 2005). These criteria are based on the three following basic concepts:

- **Computation:** a computation of a process is a maximal sequence of reduction steps, that is an infinite sequence of reductions, or a finite sequence of reductions leading to a state with no outgoing reductions;
- **Computation that signal completion:** a computation signals completion if it passes through a state that contains a given subprocess representing the completion signal.

For instance, in BGF, we could assume that the completion signal is represented by the presence of one molecule of a given species.

We are now ready to present the first criterion for Turing completeness.

Criterion 1: Given a partial recursive function with a given input there is a corresponding process in the process algebra such that:

- if the function is defined then all computations are finite and signal completion;
- if the function is not defined then all computations are infinite and do not signal completion.

The above criterion can be relaxed by requiring that at least one computation is correct (instead of all computations) and considering only one aspect between finiteness of the computation or generation of the completion signal. In this way, we can obtain these two weaker criteria.

Criterion 2: Given a partial recursive function with a given input there is a corresponding process in the process algebra such that:

- if the function is defined then at least one computation is finite;
- if the function is not defined then all computations are infinite.

Criterion 3: Given a partial recursive function with a given input there is a corresponding process in the process algebra such that:

- if the function is defined then at least one computation signals completion;
- if the function is not defined then all computations do not signal completion.

Trivially, we have that Criterion 1 implies both Criteria 2 and 3, i.e. if a process algebra is Turing complete according to the former it is Turing complete also according to the latter. No other direct implications among these criteria exist. For instance, in (Busi *et al.* 2009) it is proved that CCS (Milner 1989) with replication instead of recursion is Turing complete according to Criterion 2 but not according to Criteria 1 or 3.

In this section we prove that BGF is Turing complete according to Criterion 1. On the other hand, when we prove that fragments of BGF are not Turing universal (see Sections 5 and 6) we show that they cannot be Turing complete not only according to the same criterion, but also according to any of the weaker criteria. This strengthens the expressiveness gap that we prove.

In order to prove that BGF is Turing complete, we show how to model Random Access Machines (RAMs) (Minsky 1967), a well known Turing powerful formalism based on registers containing nonnegative natural numbers. The registers are used by a program, that is a set of indexed instructions I_i of two possible kinds:

- $i : Inc(r_j)$ that increments the register r_j and then moves to the execution of the instruction with index $i + 1$ and
- $i : DecJump(r_j, s)$ that attempts to decrement the register r_j ; if the register does not hold 0 then the register is actually decremented and the next instruction is the one with index $i + 1$, otherwise the next instruction is the one with index s .

We assume the existence of a special instruction I_{halt} corresponding to program termination.

In our encoding of RAMs, we use a simplified notation for BGF definitions in which actions can be written in sequence. For instance the definition $A = \pi_1; \pi_2; C$ is a shorthand for the two definitions $A = \pi_1; B$ and $B = \pi_2; C$.

The encoding considers one species I^i for each instruction I_i . The behavior of the molecules of species I^i is to update the registers according to the corresponding instruction I_i , and then produce one molecule of species I^j corresponding to the subsequent instruction to be executed.

Formally, the species corresponding to the instructions are defined as follows:

$$I^i = \begin{cases} !inc_j; ?ack_j; I^{i+1} & \text{if } I_i = i : Inc(r_j) \\ !dec_j; ?ack_j; I^{i+1} \oplus !zero_j; I^s & \text{if } I_i = i : DecJump(r_j, s) \\ \mathbf{0} & \text{if } I_i = I_{halt} \end{cases}$$

In Figure 3 we graphically depict the above definitions using the notation introduced in the Example 6. In case of an increment instruction, a request for increment inc_j is

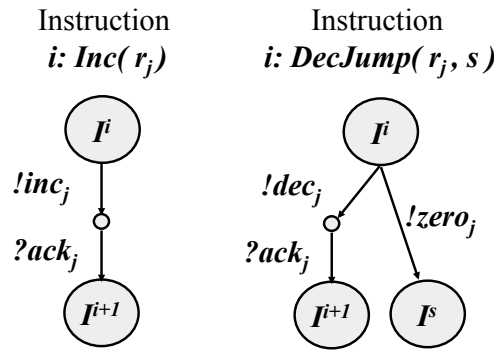


Fig. 3. Encoding of RAM instructions.

considered, then an acknowledgment is required to have confirmation that the increment actually took place, and finally the next instruction is activated. In case of a decrement, either a decrement or a test for emptiness can take place: in the first case an acknowledgment is required before activating the next instruction; in the second case the jump is executed. In case of the terminating instruction, the corresponding molecule simply does nothing.

Each register r_j is modeled by a polymer similar to those described in the Example 5. In this case the seed is of species Z^j and the monomers are of species R^j . The number of monomers in the polymer coincides with the register content, namely, when the register holds the number l the polymer is composed of exactly l monomers. As it is not possible to know a priori the number of monomers necessary during the computation, we consider a factory, that is, a molecule of species RF^j which is responsible for the the generation of the molecules of species R^j whenever they are needed. The last associated molecule in the polymer is the only one able to interact with the instruction molecules: if it is of species Z^j the active action is $?zero_j$, if it is of species R^j the active action is $?dec_j$. The

effect of the execution of $?dec_j$ is the dissociation of the last associated molecule from the polymer.

The formal definition of the species used to model registers is as follows:

$$\begin{aligned} Z^j &= ?zero_j; Z^j \oplus \&?link_j; \%?link_j; Z^j \\ RF^j &= ?inc_j; (RF^j | (\&!link_j; !ack; R^j)) \\ R^j &= (\&?link_j; R^j) \oplus (?dec_j; \%!link_j; !ack; \mathbf{0}) \\ R^j &= \%?link_j; R^j \end{aligned}$$

In Figure 4 we graphically depict the encoding of registers.

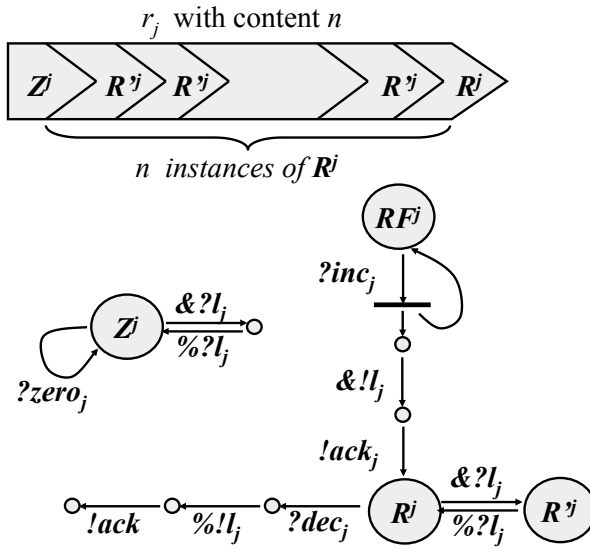


Fig. 4. Encoding of RAM registers.

The remainder of this section is devoted to the formal proof of correctness of this RAM encoding. We use the following notation. Given a RAM with registers r_1, \dots, r_n , we denote with $(I_i, r_1 = l_1, \dots, r_n = l_n) \mapsto (I_j, r_1 = l'_1, \dots, r_n = l'_n)$ its possible steps of computation. Namely, if the RAM is going to execute instruction I_i , and the register contents are l_1, \dots, l_n , respectively, then the next instruction is I_j and the new register contents are l'_1, \dots, l'_n , respectively.

In the following we need to treat as equivalent some syntactically different solutions which represent the same biological system. For instance, the two solutions

$$\begin{aligned} Z^1_{(?link_1,1)} \mid R^1_{(?link_1,4)::(!link_1,1)} \mid R^1_{(!link_1,4)} \\ Z^1_{(?link_1,2)} \mid R^1_{(?link_1,3)::(!link_1,2)} \mid R^1_{(!link_1,3)} \end{aligned}$$

both denote the polymer representing the register r_1 with content 2, even if they differ in their association keys. Formally, we have that two solutions S and T are *equivalent* if

there exists an injective renaming ρ for the association keys in S such that $(S[\rho]) = T$, where $S[\rho]$ denotes the result of the application of the injective renaming to the solution S . In the example above, the injective renaming used to prove that the two solutions are equivalent is $\{1 \mapsto 2, 4 \mapsto 3\}$.

We denote with $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$ the set of equivalent solutions which represent the RAM ready to execute the instruction I_i and in which the registers r_1, \dots, r_m have contents l_1, \dots, l_m , respectively. Formally, $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$ is the set of solutions equivalent to:

$$\begin{aligned} & I^i \mid RF^1 \mid \dots \mid RF^n \mid \\ & Z^1_{\langle ?link_1, k_1^1 \rangle} \mid R^1_{\langle ?link_1, k_1^2 \rangle :: \langle !link_1, k_1^1 \rangle} \mid \dots \mid R^1_{\langle ?link_1, k_1^{l_1} \rangle :: \langle !link_1, k_1^{l_1-1} \rangle} \mid R^1_{\langle !link_1, k_1^{l_1} \rangle} \mid \\ & \dots \\ & Z^m_{\langle ?link_m, k_m^1 \rangle} \mid R^m_{\langle ?link_m, k_m^2 \rangle :: \langle !link_m, k_m^1 \rangle} \mid \dots \mid R^m_{\langle ?link_m, k_m^{l_m} \rangle :: \langle !link_m, k_m^{l_m-1} \rangle} \mid R^m_{\langle !link_m, k_m^{l_m} \rangle} \end{aligned}$$

Given a RAM denoted with \mathcal{R} , having instructions I_1, \dots, I_n and registers r_1, \dots, r_m , we use $E_{\mathcal{R}}$ to denote the definitions of the species $I^1, \dots, I^n, Z^1, \dots, Z_m, RF^1, \dots, RF^m, R^1, \dots, R^m$, and R'^1, \dots, R'^m as defined above. Thus, given one of the possible configurations $(I_i, r_1 = l_1, \dots, r_m = l_m)$ of \mathcal{R} , we model it with the BGF $(E_{\mathcal{R}}, S)$ where S is any of the solutions in $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$.

We are now ready to prove the correctness result.

Theorem 1. Let \mathcal{R} be a RAM. Given one of its possible configurations $(I_i, r_1 = l_1, \dots, r_m = l_m)$ and a solution $S_0 \in \llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$, we have that:

- either $I_i = I_{halt}$ and $Next(E_{\mathcal{R}}, S_0)$ is empty;
- or $(I_i, r_1 = l_1, \dots, r_m = l_m) \mapsto (I_j, r_1 = l'_1, \dots, r_m = l'_m)$ and there exist S_1, \dots, S_z such that for every $0 \leq x < z$ we have that $Next(E_{\mathcal{R}}, S_x)$ contains only one transition which has S_{x+1} as its target solution, and moreover $S_z \in \llbracket (I_j, r_1 = l'_1, \dots, r_m = l'_m) \rrbracket$.

Proof. The proof is by case analysis on the following four possible cases: $I_i = I_{halt}$, $I_i = i : Inc(r_j)$, $I_i = i : DecJump(r_j, s)$ with $l_j > 0$, and $I_i = i : DecJump(r_j, s)$ with $l_j = 0$.

In case $I_i = I_{halt}$ case we have that the molecules in S_0 can perform the following actions: $?zero_j$, $\&?link_j$, inc_j , and $?dec_j$. As there are no complementary actions, or internal τ actions, no transition is admitted thus $Next(E_{\mathcal{R}}, S_0)$ is empty.

Assume that $I_i = i : Inc(r_j)$. In this case $(I_i, r_1 = l_1, \dots, r_m = l_m) \mapsto (I_{i+1}, r_1 = l_1, \dots, r_j = l_m + 1, \dots, r_m = l_m)$. Consider now $S_0 \in \llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$. We have that S_0 has only one possible transition following from a synchronization on channel inc_j . The reached solution, can perform only one transition, that is an association on the surface $link_j$. This last reached solution has only one transition following from a synchronization on the channel ack . The reached solution is in $\llbracket (I_{i+1}, r_1 = l_1, \dots, r_j = l_m + 1, \dots, r_m = l_m) \rrbracket$.

The remaining two cases are treated similarly. □

As a corollary of this Theorem we have that BGF is Turing complete according to Criterion 1.

Corollary 2. The process algebra BGF is Turing complete according to Criterion 1.

Proof. As the RAM computational model is Turing universal, we have that given any partial recursive function with a given input, we can consider a corresponding RAM \mathcal{R} with an initial configuration $(I_i, r_1 = l_1, \dots, r_m = l_m)$ that contains the input in one of its registers.

Assume that the function is defined for the given input. We have that the RAM \mathcal{R} with initial configuration $(I_i, r_1 = l_1, \dots, r_m = l_m)$ terminates its computation. Assume that the computation has length i . Applying i -times Theorem 1 we can prove that the computation of the encoding $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$ in BGF proceeds deterministically, terminates its computation, and produces the molecule I_{halt} that we can consider as the signal of completion.

Assume now that the function is not defined for the given input. In this case the RAM \mathcal{R} with initial configuration $(I_i, r_1 = l_1, \dots, r_m = l_m)$ does not terminate its computation. By contraposition, we can prove that also the encoding $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$ in BGF does not terminate and does not produce the molecule I_{halt} . In fact, if $\llbracket (I_i, r_1 = l_1, \dots, r_m = l_m) \rrbracket$ could terminate or produce the molecule I_{halt} , we can apply Theorem 1 to prove that also the RAM \mathcal{R} with initial configuration $(I_i, r_1 = l_1, \dots, r_m = l_m)$ should terminate. \square

We complete the section discussing the expressiveness gap between BGF and CGF (that corresponds to BGF without the association and dissociation primitives). The computational strength of CGF has been studied in (Zavattaro and Cardelli 2008). In particular, it is shown that in CGF with a nondeterministic semantics (as the one considered in this paper) properties such as universal termination (all computations terminate), existential termination (existence of a terminating computation), and coverability (existence of a computation leading to a solution containing at least some given molecules) are decidable. As a consequence of these results we have that CGF cannot be Turing complete according to any of the three criteria described above. On the contrary, in Corollary 2 we have shown that BGF is Turing complete according to the strongest of the three criteria. In the next three sections we try to shed light on this expressiveness gap between BGF and CGF looking for minimal sets of primitives that make BGF Turing complete. The final result is that all primitives but association and dissociation can be removed from BGF without reducing its computational strength.

4. Turing completeness of BGF with only association and dissociation

In this Section we consider the fragment of BGF obtained by removing the internal, input, and output prefixes. We refer to this fragment with $\text{BGF}_{-\tau?!}$. This simple process algebra, including only association and dissociation actions, is already Turing complete. We prove this result simply observing that the encoding of RAMs presented in the previous section does not contain internal τ actions, and moreover it can be rewritten replacing all input/output actions with an association action followed by the corresponding dissociation. For instance, given two interacting species $X = ?a; \mathbf{0}$ and $Y = !a; \mathbf{0}$, we can

consider the alternative definitions $X = \&?a; \%?a; \mathbf{0}$ and $Y = \&!a; \%!a; \mathbf{0}$ in which we use association and dissociation actions instead of input and output.

We now rephrase the previous encoding following this idea. The species corresponding to the instructions are redefined as follows:

$$I^i = \begin{cases} \&!inc_j; \%!inc_j; \&?ack; \%?ack; I^{i+1} & \text{if } I_i = i : Inc(r_j) \\ \&!dec_j; \%!dec_j; \&?ack; \%?ack; I^{i+1} \oplus \\ \quad \&!zero_j; \%!zero_j; I^s & \text{if } I_i = i : DecJump(r_j, s) \\ \mathbf{0} & \text{if } I_i = I_{halt} \end{cases}$$

The new definition of the species used to model registers is as follows:

$$\begin{aligned} Z^j &= \&?zero_j; \%?zero_j; Z^j \oplus \&?link_j; \%?link_j; Z^j \\ RF^j &= \&?inc_j; \%?inc_j; (RF^j | (\&!link_j; \&!ack; \%!ack; R^j)) \\ R^j &= (\&?link_j; R'^j) \oplus (\&?dec_j; \%?dec_j; \%!link_j; \&!ack; \%!ack; \mathbf{0}) \\ R'^j &= \%?link_j; R^j \end{aligned}$$

Keeping all the other definitions as in the previous section, we have that both Theorem 1 and Corollary 2 hold also for this new encoding of RAM in $BGF_{-\tau?}$.

5. Non Turing completeness of BGF without association

We now move to the fragment of BGF obtained by removing the association prefixes. We refer to this fragment with $BGF_{-\&}$. In $BGF_{-\&}$ one complex, that is a multiset of connected molecules, cannot increment its size. This property of $BGF_{-\&}$ allows us to prove that this fragment of BGF is no longer Turing complete, even if we consider the weaker Criteria 2 and 3 presented in Section 3. As far as Criterion 2 is concerned, we show that the existence of a terminating computation is decidable in $BGF_{-\&}$. This is enough to prove that $BGF_{-\&}$ cannot be Turing complete according to this criterion: in fact, it imposes that the encoding of a given partial function for a given input has a terminating computation if and only if the function is defined on that input. As far as Criterion 3 is concerned, we show that the reachability of a solution containing at least one molecule of a given species. Assuming that this is the unique reasonable way to signal completion in a calculus such as BGF, this is sufficient to prove that $BGF_{-\&}$ cannot be Turing complete neither according to this criterion: in fact, it imposes that the encoding of a given partial function for a given input signals completion if and only if the function is defined on that input.

The decidability results follow by reduction to decidable problems in 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 termination, reachability or coverability (Esparza and Nielsen 1994). We recall here the basic notation, for a full description of this computational model see (Reisig 1985).

Definition 4. A *P/T net* is a tuple $N = (\mathcal{P}, \mathcal{T}, m_0)$, where \mathcal{P} is a finite set of *places*. A finite multiset over the set \mathcal{P} of places is called a *marking*. If a place P occurs n times

in a marking m , we say that n tokens are placed in P . m_0 is the initial marking. \mathcal{T} is the finite set of transitions, that is a finite relation on markings. Given a pair $\langle m', m'' \rangle$ in \mathcal{T} , we call the marking m' the *pre-set* and the marking m'' the *post-set*. 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 $(m \setminus m') \uplus m''$ where \setminus and \uplus are the difference and union operators for multisets, respectively.

In the following we will consider the termination and the coverability problems for P/T nets. Given a P/T net $(\mathcal{P}, \mathcal{T}, m_0)$, the termination problem (resp. coverability problem) consists of checking whether there exists a sequence of markings m_0, \dots, m_n such that, for $0 \leq i < n$, the marking m_i can be modified into m_{i+1} by some transition in \mathcal{T} , and no transitions can fire in m_n (resp. a given marking m is contained in m_n). Both problems turn out to be decidable (Esparza and Nielsen 1994) for P/T nets.

The basic idea of the P/T net semantics we present for $\text{BGF}_{-\&}$, is to represent the evolution of complexes in the solution. A complex is a solution composed of molecules connected by associations. We represent each complex in the solution with a token in a corresponding place of the P/T net. We will show that given a system (E, S) specified in $\text{BGF}_{-\&}$ it is possible to compute an upper-bound to the size of complexes in solutions in $TG(E, S)$, thus only finitely many different complexes can be generated in the given system. This guarantees that the corresponding P/T net is finite.

We first formalize the notion of complex.

Definition 5 (Complex). A *complex* is a nonempty connected solution, that is a solution $X_{H_1}^1 | \dots | X_{H_n}^n$ with $n \geq 1$ such that given any pair of indexes $1 \leq i < j \leq n$ there exists a sequence of indexes l_1, \dots, l_m , with $l_1 = i$ and $l_m = j$, such that for any $1 \leq s < m$ the association histories H_{l_s} and $H_{l_{s+1}}$ share at least one association key.

As done in the proof of correctness of the RAM encoding in Section 3, we do not distinguish between two *equivalent* complexes, that is two complexes S and T for which there exists an injective renaming ρ for the association keys in S such that $(S[\rho]) = T$, where $S[\rho]$ denotes the result of the application of the injective renaming to the solution S . We will use $[S]$ to denote the set of complexes equivalent to S .

Before presenting the P/T net semantics for $\text{BGF}_{-\&}$ we show that it is possible to compute an upper-bound to the size of the complexes that can be generated in a given system specified in $\text{BGF}_{-\&}$. The first observation follows from the fact that the absence of association actions guarantees that the number of different association keys in a system cannot grow. This is formalized in the following proposition.

Proposition 3. Let (E, S) be a specification in $\text{BGF}_{-\&}$ and let T be a solution in $\text{Next}(E, S)$. We have that the number of different association keys in T is not greater than the number of different association keys in S .

Proof. The thesis trivially follows from the fact that all transitions in $\text{Next}(E, S)$ are not *associations*. \square

As a corollary, we have that given (E, S) in $\text{BGF}_{-\&}$, it is not possible to generate in

$TS(E, S)$ a complex with a size greater than the number of different association keys in S plus one.

Corollary 3. Let (E, S) be a specification in $BGF_{-\&}$ and let T be a solution in $TG(E, S)$. We have that all complexes in T have size smaller than the number of different association keys in S plus one.

Proof. The thesis follows from Proposition 3 and by the observation that a complex of size n contains at least $n - 1$ different association keys. \square

We now define the function $\text{dec}(-)$ that decomposes a solution in the set of its complexes.

Definition 6 (Decomposition of a solution). Given a solution S , this can be partitioned in the complexes S_1, \dots, S_n such that $S = S_1 | \dots | S_n$. The *decomposition* of S is formally defined as the multiset $\text{dec}(S) = \{\{[S_1], \dots, [S_n]\}\}$.

We are now ready to define the P/T net semantics for $BGF_{-\&}$.

Definition 7. Consider a system (E, S) specified in $BGF_{-\&}$. The P/T net of (E, S) , denoted with $PT_{-\&}(E, S)$, is the tuple $(\mathcal{P}, \mathcal{T}, m_0)$ where:

— \mathcal{P} is the following set of (equivalence classes of) complexes

$$\{[V] \mid V \text{ is a complex of size smaller than the number of different association keys in } S \text{ plus one and s.t. all species and symbols } ?a \text{ and } !a \text{ occurring in } V \text{ are also in } E \text{ or } S\}$$

— \mathcal{T} is the following set of transitions (i.e. pairs of markings)

$$\{\langle [V], \text{dec}(T) \rangle \mid V \in \mathcal{P} \text{ and } V \rightarrow T \in \text{Next}(E, V)\} \cup \{\langle \{\{[V_1], [V_2]\}, \text{dec}(T) \rangle \mid V_1, V_2 \in \mathcal{P} \text{ and } V_1 | V_2 \rightarrow T \in \text{Next}(E, V_1 | V_2)\}$$

— $m_0 = \text{dec}(S)$

It is worth to observe that, given (E, S) of $BGF_{-\&}$, then $PT_{-\&}(E, S)$ is a well defined P/T net. In fact, we have that the set of places \mathcal{P} is finite. This follows from the upper-bound to the size of the complexes in \mathcal{P} , and the fact that these complexes are composed of molecules belonging to a finite amount of species and with associations on a finite set of symbols $?a$ and $!a$. The finiteness of \mathcal{P} guarantees that also the set of transitions \mathcal{T} is finite.

We now prove the correctness of the P/T net semantics.

Theorem 2. Consider a system (E, S) specified in $BGF_{-\&}$. We have that the following hold:

- if $T_1 \rightarrow T_2$ is in $TG(E, S)$ then there exists a transition in $PT_{-\&}(E, S)$ that modifies the marking $\text{dec}(T_1)$ into $\text{dec}(T_2)$;
- if $\text{dec}(T_1)$ is modified into the marking m by some transition in $PT_{-\&}(E, S)$ and T_1 is a solution in $TG(E, S)$, then there exists a solution T_2 such that $\text{dec}(T_2) = m$ and $T_1 \rightarrow T_2$ is in $TG(E, S)$.

Proof. Assume that $T_1 \rightarrow T_2$ is in $TG(E, S)$. Let $\mathbf{dec}(T_1) = \{\{V_1, V_2, \dots, V_n\}\}$. The transition $T_1 \rightarrow T_2$ follows from the execution of actions performed by (one or two) molecules belonging to either one or two complexes. We consider only the first case, as the second one is treated similarly. It is not restrictive to assume that the complex containing the molecule(s) performing the action(s) is V_1 . In this case we have that $T_2 = V_1'|V_2|\dots|V_n$. The complex V_1 can perform the same transition also in isolation, that is, there exists a solution V_1'' equivalent to V_1' such that $V_1 \rightarrow V_1''$ is in $Next(E, V_1)$. Notice that we had to consider a different solution V_1'' as target for this transition. In fact, if the transition is an *association*, it could be the case that the fresh key (that is the smallest among the unused association keys) adopted for the solution V_1 is different from the one adopted for $T_1 = V_1|V_2|\dots|V_n$. Nevertheless, as V_1' and V_1'' are equivalent, we have that $\mathbf{dec}(V_1') = \mathbf{dec}(V_1'')$. We complete this first part of the proof observing that by definition of \mathcal{T} , the transition $V_1 \rightarrow V_1''$ in $Next(E, V_1)$ ensures the presence of a transition $\langle [V], \mathbf{dec}(V_1'') \rangle$ that can modify the marking $\mathbf{dec}(T_1) = \{\{V_1, V_2, \dots, V_n\}\}$ to the marking $\mathbf{dec}(V_1'') \uplus \{\{V_2, \dots, V_n\}\} = \mathbf{dec}(T_2)$.

Assume now that T_1 is a solution in $TG(E, S)$ such that $\mathbf{dec}(T_1)$ is modified in the marking m by some transition in \mathcal{T} . By definition of \mathcal{T} , this transition can be of two kinds, either $\langle [V], \mathbf{dec}(T) \rangle$ or $\langle \{\{[V_1], [V_2]\}\}, \mathbf{dec}(T) \rangle$. We consider only the first case, as the second one is treated similarly. We have that $\mathbf{dec}(T_1) = \{\{[V]\}\} \uplus \mathbf{dec}(T_1')$ (assuming $T_1 = V|T_1'$), $m = \mathbf{dec}(T) \uplus \mathbf{dec}(T_1')$, and that $V \rightarrow T$ is in $Next(E, V)$. The same kind of transition can be performed also in the (greater) solution T_1 , namely, as $V \rightarrow T$ is in $Next(E, V)$ then there exists a solution T' equivalent to T such that $V|T_1' \rightarrow T'|T_1'$ is in $TG(E, S)$ (as discussed above, we need to consider a solution T' equivalent to T as in case the transition is an *association* two different association keys could be adopted in the solutions V and $V|T_1'$). We conclude the proof observing that the thesis holds if we take $T_2 = T'|T_1'$. In fact, as T' is equivalent to T , then $\mathbf{dec}(T_2) = \mathbf{dec}(T') \uplus \mathbf{dec}(T_1') = \mathbf{dec}(T) \uplus \mathbf{dec}(T_1') = m$. \square

We conclude this section showing that the existence of a terminating computation and the existence of a computation leading to a solution containing at least one molecule of a given species are both decidable properties for $BGF_{-\&}$.

Corollary 4. Given (E, S) of $BGF_{-\&}$, we have that there exists a terminating computation in $TG(E, S)$ (i.e. a sequence of solutions S_0, \dots, S_n such that $S_0 = S$, $S_i \rightarrow S_{i+1} \in TG(E, S)$ for $0 \leq i < n$, and $Next(E, S_n)$ is empty) if and only if there is a terminating computation in $PT_{-\&}(E, S)$.

Proof. If there is a terminating computation in $TG(E, S)$ (resp. in $PT_{-\&}(E, S)$) then, by Theorem 2, we have that there is also a terminating computation in $PT_{-\&}(E, S)$ (resp. in $TG(E, S)$). \square

As the existence of a terminating computation is decidable in P/T nets we have that the same holds also in $BGF_{-\&}$. Thus we can conclude that $BGF_{-\&}$ cannot be Turing complete according to the Criterion 2.

Corollary 5. Given (E, S) of $BGF_{-\&}$, we have that there exists a computation leading

to a solution T that contains at least one molecule of a given species X , if and only if there is a computation in $PT_{-\&}(E, S)$ leading to a marking with a token in a place $[V]$ such that at least one molecule of species X belongs to V .

Proof. If there is a computation in $TG(E, S)$ leading to a solution T that contains at least one molecule of a given species X , by Theorem 2 there exists a computation in $PT_{-\&}(E, S)$ leading to a marking m such that $\text{dec}(T) = m$. Hence, m has at least one token in a place $[V]$ such that V contains one molecule of species X .

On the other hand, if $PT_{-\&}(E, S)$ has a computation leading to a marking m with at least one token in a place $[V]$ such that V contains one molecule of species X , by Theorem 2 there exists a computation in $TG(E, S)$ leading to a solution T such that $\text{dec}(T) = m$. Hence, T contains at least one molecule of species X . \square

As the existence of a computation leading to a marking containing at least one token in a given place is a particular case of the coverability problem which is decidable for P/T nets, we have that the possibility to generate a molecule of a given species is decidable also in $BGF_{-\&}$. Thus we can conclude that $BGF_{-\&}$ cannot be Turing complete neither according to the Criterion 3.

6. Non Turing completeness of BGF without dissociation

We now complete our analysis of the fragments of BGF showing that also the fragment obtained by removing the dissociation actions is no longer Turing complete. We refer to this fragment with $BGF_{-\%}$. Also for $BGF_{-\%}$, as done in the previous Section for $BGF_{-\&}$, we prove the decidability of termination (thus $BGF_{-\%}$ cannot be Turing complete according to the Criterion 2) and of the existence of a computation in which at least one molecule of a given species is produced (thus $BGF_{-\%}$ cannot be Turing complete according to the Criterion 3).

Before discussing the decidability results we observe that, even if there is no dissociation actions, in $BGF_{-\%}$ complexes can reduce their size. This is a consequence of the possibility for complexed molecules to split and distribute their associations among the produced molecules. Consider, for instance, a polymer of length n . If the monomer at the intermediary position i splits in two molecules (as, e.g., discussed in the Example 6), and passes its two associations to the two produced molecules, the initial polymer of length n generates one polymer of length i and one of length $n - i + 1$.

The decidability results are proved also in this section by reduction to the termination and coverability problems for P/T nets. Nevertheless, the proof technique presented in the previous section does not apply to $BGF_{-\%}$ because in this fragment there exists, in general, no upper-bound to the size of the complexes that can be generated from an initial solution.

The P/T net semantics we present in this section is based on the following idea: each molecule X_H in the solution is represented with a token in a corresponding place $X_{\text{set}(H)}$ where $\text{set}(H)$ is the function defined in Definition 2 that associates to an association history the set of symbols $?a, !a$ occurring in it. In this way, the P/T net semantics abstract away from the association keys but, as we will prove, this loss of information is not

problematic in $\text{BGF}_{-\%}$ where associations, once created, cannot be used by subsequent actions.

Before presenting the PT net semantics, we introduce an auxiliary operator $\text{nokeys}(_)$ that removes the association keys from the solutions. Given the solution $S = X_{H_1}^1 | \cdots | X_{H_n}^n$ we define $\text{nokeys}(S) = \uplus_{i \in \{1, \dots, n\}} \{\{X_{\text{set}(H_i)}^i\}\}$ where $\uplus_{i \in I} M_i$ is the multiset union of the multisets M_i with index i taken from the set of indexes I .

We are now ready to define the P/T net semantics for $\text{BGF}_{-\%}$.

Definition 8. Consider a system (E, S) specified in $\text{BGF}_{-\%}$. The P/T net of (E, S) , denoted with $PT_{-\%}(E, S)$, is the tuple $(\mathcal{P}, \mathcal{T}, m_0)$ where:

— \mathcal{P} is the following set of species decorated with a set of symbols $?a, !a$

$$\{X_S \mid X \text{ is a species defined in } E \text{ and } S \text{ is a set of symbols } ?a, !a \text{ with } a \text{ occurring in } E \text{ or } S\}$$

— \mathcal{T} is the following set of pairs of markings

$$\begin{aligned} \langle \{\{X_S\}, \uplus_{i \in \{1, \dots, n\}} \{\{Y_{S_i}^i\}\} \rangle \mid & X_H \rightarrow Y_{H_1}^1 | \cdots | Y_{H_n}^n \in \text{Next}(E, X_H), X_S \in \mathcal{P}, \\ & S = \text{set}(H), \text{ and } S_i = \text{set}(H_i) \text{ for } 1 \leq i \leq n \rangle \cup \\ \langle \{\{X_S, X_{S'}\}, \uplus_{i \in \{1, \dots, n\}} \{\{Y_{S_i}^i\}\} \rangle \mid & X_H | X_{H'} \rightarrow Y_{H_1}^1 | \cdots | Y_{H_n}^n \in \text{Next}(E, X_H | X_{H'}), \\ & X_S \in \mathcal{P}, X_{S'} \in \mathcal{P}, S = \text{set}(H), S' = \text{set}(H'), \\ & \text{ and } S_i = \text{set}(H_i) \text{ for } 1 \leq i \leq n \rangle \end{aligned}$$

— $m_0 = \text{nokeys}(S)$

It is easy to see that $PT_{-\%}(E, S)$ is a finite P/T net. In fact, the places X_S can be defined considering a finite set of species and a finite set of symbols $?a, !a$.

We now formalize the correspondence between the semantics of a system (E, S) specified in $\text{BGF}_{-\%}$ and the corresponding P/T net semantics.

Theorem 3. Consider a system (E, S) specified in $\text{BGF}_{-\%}$. We have that the following hold:

- if $T_1 \rightarrow T_2$ is in $TG(E, S)$ then there exists a transition in $PT_{-\%}(E, S)$ that modifies the marking $\text{nokeys}(T_1)$ into $\text{nokeys}(T_2)$;
- if $\text{nokeys}(T_1)$ is modified to the marking m by some transition in $PT_{-\%}(E, S)$ and T_1 is a solution in $TG(E, S)$, then there exists a solution T_2 such that $\text{nokeys}(T_2) = m$ and $T_1 \rightarrow T_2$ is in $TG(E, S)$.

Proof. Assume that $T_1 \rightarrow T_2$ is in $TG(E, S)$. Let $T_1 = X_{H_1}^1 | \cdots | X_{H_n}^n$. The transition $T_1 \rightarrow T_2$ follows from the execution of actions performed by (one or two) molecules in T_1 . We consider only the first case, as the second one is treated similarly. It is not restrictive to assume that the molecule performing the action is $X_{H_1}^1$. In this case we have that $X_{H_1}^1 \rightarrow Y_{H'_1}^1 | \cdots | Y_{H'_m}^m$ is in $\text{Next}(E, X_{H_1}^1)$ and that $T_2 = Y_{H'_1}^1 | \cdots | Y_{H'_m}^m | X_{H_2}^2 | \cdots | X_{H_n}^n$. By definition of \mathcal{T} , we have the transition $\langle \{\{X_{\text{set}(H_1)}^1\}, \{\{Y_{\text{set}(H'_1)}^1, \dots, Y_{\text{set}(H'_m)}^m\}\}\} \rangle$ in $PT_{-\%}(E, S)$. This transition modifies the marking $\text{nokeys}(T_1) = \{\{X_{\text{set}(H_1)}^1, \dots, X_{\text{set}(H_n)}^n\}\}$ in the marking $\text{nokeys}(T_2) = \{\{Y_{\text{set}(H'_1)}^1, \dots, Y_{\text{set}(H'_m)}^m, X_{\text{set}(H_2)}^2, \dots, X_{\text{set}(H_n)}^n\}\}$.

Assume now that T_1 is a solution in $TG(E, S)$ such that $\text{nokeys}(T_1)$ is modified in

the marking m by some transition in \mathcal{T} . By definition of \mathcal{T} , this transition can be of two kinds, either $\langle \{\{X_S\}, \uplus_{i \in \{1, \dots, n\}} \{\{Y_{S_i}^i\}\} \rangle$ or $\langle \{\{X_S, X_{S'}\}, \uplus_{i \in \{1, \dots, n\}} \{\{Y_{S_i}^i\}\} \rangle$. We consider only the first case, as the second one is treated similarly. We have that $\text{nokeys}(T_1) = \{\{X_{\text{set}(H)}, X_{\text{set}(H_1)}^1, \dots, X_{\text{set}(H_l)}^l\}\}$ (assuming $T_1 = X_H | X_{H_1}^1 | \dots | X_{H_l}^l$) with $\text{set}(H) = S$, $m = \uplus_{i \in \{1, \dots, n\}} \{\{Y_{S_i}^i\}\} \uplus \{\{X_{\text{set}(H_1)}^1, \dots, X_{\text{set}(H_l)}^l\}\}$ and that $X_H \rightarrow Y_{H_1}^1 | \dots | Y_{H_n}^n$ is in $\text{Next}(E, X_H)$ with $S_i = \text{set}(H'_i)$ for $1 \leq i \leq n$. The same kind of transition can be performed also in the (greater) solution T_1 , namely, $T_1 \rightarrow Y_{H_1}^1 | \dots | Y_{H_n}^n | X_{H_1}^1 | \dots | X_{H_l}^l$ is in $TG(E, S)$. We conclude the proof observing that the thesis holds if we take $T_2 = Y_{H_1}^1 | \dots | Y_{H_n}^n | X_{H_1}^1 | \dots | X_{H_l}^l$. In fact, $\text{nokeys}(T_2) = \{\{Y_{\text{set}(H'_1)}^1, \dots, Y_{\text{set}(H'_n)}^n, X_{\text{set}(H_1)}^1, \dots, X_{\text{set}(H_l)}^l\}\} = m$. \square

As a consequence of the above theorem we have that the existence of a terminating computation and the existence of a computation leading to a solution containing at least one molecule of a given species are both decidable properties for $\text{BGF}_{-\%}$.

Corollary 6. Given (E, S) of $\text{BGF}_{-\%}$, we have that there exists a terminating computation in $TG(E, S)$ if and only if there is a terminating computation in $PT_{-\%}(E, S)$.

Proof. As in Corollary 4 considering Theorem 3 instead of Theorem 2. \square

Corollary 7. Given (E, S) of $\text{BGF}_{-\%}$, we have that there exists a computation leading to a solution T that contains at least one molecule of a given species X , if and only if there is a computation in $PT_{-\%}(E, S)$ leading to a marking with a token in a place X_S .

Proof. As in Corollary 5 considering Theorem 3 instead of Theorem 2. \square

As the existence of a terminating computation (resp. the existence of a computation leading to a marking containing at least one token in a given place) is decidable in P/T nets we have that termination (resp. the possibility to generate a molecule of a given species) is decidable also in $\text{BGF}_{-\%}$. Thus we can conclude that $\text{BGF}_{-\%}$ cannot be Turing complete according to the Criterion 2 (resp. Criterion 3).

7. Conclusion

Turing-powerful mechanisms are not a requirement for building sophisticated nanomachines. Yet, the existence of Turing-powerful mechanisms guarantees a certain level of generality and flexibility in constructing machinery of any desired complexity, and provides evolution with adaptable toolkits to build upon. This paper highlights the fact that the elaborate nature of biomolecules increases system complexity in a finitary combinatorial way that qualitatively transcends the expressive power of any notation that models at the level of ordinary chemical reactions between simple species. We have shown that the biochemically-inspired operations of association and dissociation, formalized in a very basic form, are sufficient to raise expressiveness to the level of Turing-completeness, while finite sets of chemical reactions are not sufficient. In other words, finite programming constructs that are Turing powerful are required to model biochemical systems but not to model simple chemical systems.

In our proof of Turing completeness an important role is played by the *association histories* that, attached to each molecule, keep track of the currently active associations. The information stored in the association histories indicates which dissociations can occur, as only previously associated molecules can give rise to a dissociation event. Despite the relevance of association histories in the proof of Turing completeness that we have presented, it is important to notice that Turing universality follows from the intended meaning of the association and dissociation primitives, and does not rely on the notion of association histories. In fact, one could define an alternative semantics for the association and dissociation primitives in which association histories are replaced by, e.g., mechanisms for the creation of fresh channels such as those in the π -calculus (Milner *et al.* 1992). In terms of the π -calculus, the creation of an association between two molecules can be seen as the creation of a new channel, which is kept private between the two associated molecules, and which is used only once on molecule dissociation.

The κ -calculus is another model for biology based on the notion of molecules that can form complexes (Danos and Laneve 2004). The κ -calculus is reaction centric instead of process centric, that is, the evolution of the system is governed by reactions that modify the state and the bonds within a group of reacting molecules. Besides permitting more than two molecules to interact, the rules in κ -calculus also allows for the specification of expressive interactions not allowed in BGF. For instance, as already commented in the Introduction, in κ -calculus reactions can be written that modify the internal state of two complexed molecules without breaking the bond connecting them. If we include a similar mechanism in BGF, for instance introducing a new pair of actions that allow two associated molecules to interact without breaking the association, we claim that the dissociation primitive is no longer necessary for Turing completeness of BGF. In fact, it should be possible to model a register as a monotonically growing polymer composed of active and inactive monomers: an increment adds an active monomer, while a decrement changes the state of one monomer from active to inactive. A similar technique has been used in (Delzanno *et al.* 2009) to prove that a fragment of the κ -calculus in which bonds cannot be destroyed is still Turing complete. Some fragments of the κ -calculus without bond destruction have been considered in (Delzanno *et al.* 2009) and revealed very interesting properties: even if they are Turing complete some reachability properties (i.e. the reachability of a solution that contains complexes with a given structure) turn out to be decidable. These decidability results follow from the monotonic growth of complexes. It is interesting to note that in $\text{BGF}_{\%}$, the fragment without dissociation we consider in Section 6, we cannot adopt the same technique to prove similar decidability results because, due to splitting of molecules, a complex in $\text{BGF}_{\%}$ could also shrink. We leave the decidability of the reachability of a complex with a given structure as an open problem for $\text{BGF}_{\%}$ that we plan to investigate in future research.

It is interesting to note that expressiveness boundaries, similar to those that we have shown between notations for basic chemistry and notations for biochemistry, have been proved also in the context of process calculi based on membrane interactions such as endocytosis, exocytosis, fusion, and fission. In (Busi and Gorrieri 2006), Busi and Gorrieri prove that a basic process calculus including endocytosis and exocytosis is Turing complete, while this is not the case when only fusion and fission are considered. This

is because endocytosis allows for the nesting of membranes with an unbounded depth, while this is not possible when only fission and fusion are considered. In BGF, instead of using membrane nesting, we consider a more basic association mechanism in order to generate structures with unbounded length.

The boundary of Turing-completeness gets even more interesting at the quantitative, approximate, level. For instance, work by Liekens and Fernando (Liekens and Fernando 2007) shows how to approximate in a language of discrete chemistry finite computations of Register Machines with an error probability smaller than any given precision $\delta > 0$. Soloveichik et al. (Soloveichik *et al.* 2008), besides proving that in notations for discrete chemistry it is not possible to precisely model any Turing powerful formalism, show also how to approximate unbounded computations.

A consequence of their results is that it is always decidable *whether* the production of a certain molecule is predicted by a model in a language of basic chemistry, while the question whether the system is *likely* to produce that molecule is in general undecidable.

This opens interesting questions about the set of decidable properties for models in the/a language of discrete chemistry. For instance, in (Zavattaro and Cardelli 2008) we have considered the termination problem. We proved that abstracting away from the stochastic semantics of the/a language of discrete chemistry, *may termination* (i.e. the existence of a sequence of reactions leading to a state in which no reaction can fire) and *must termination* (i.e. all maximal sequences of reactions lead to a terminated state) are both decidable. On the other hand, when also the stochastic rates are considered, we have that the probabilistic version of may termination (i.e. termination with probability > 0) is still decidable while the probabilistic version of must termination (i.e. termination with probability $= 1$) is no longer decidable.

We conclude the paper discussing how the results discussed in the previous paragraph about quantitative models of chemistry, as well as the results proved in this paper, scale to a possible stochastic version of the Biochemical Ground Form. The undecidability results proved in (Soloveichik *et al.* 2008) and (Zavattaro and Cardelli 2008) on finite stochastic chemical reaction networks (SCRNs) apply also to a stochastic version of BGF. In fact, BGF is a calculus that strictly includes SCRNs. On the contrary, the decidability results proved in (Soloveichik *et al.* 2008) and (Zavattaro and Cardelli 2008) do not hold in BGF under a stochastic semantics. In fact, we have proved in this paper that the addition of association and dissociation makes BGF expressive enough to model Random Access Machines deterministically.

The encoding of Random Access Machines that we have presented exploits deterministic processes of BGF and $BGF_{-\tau?}$. This allows us to conclude that the probabilistic versions of may and must termination (i.e. termination with probability > 0 or with probability $= 1$, respectively) both turn out to be undecidable in a stochastic version of these two calculi. In fact, for deterministic processes the two problems coincide, and correspond to the classical halting problem we have proved undecidable for both BGF and $BGF_{-\tau?}$.

The decidability results we have proved for $BGF_{-\&}$ and $BGF_{-\%}$ could be rephrased in a stochastic setting as follows: in stochastic versions of $BGF_{-\&}$ and $BGF_{-\%}$ it is decidable whether a system can terminate with probability > 0 (resp. a molecule of a

given species can be generated with probability > 0). In fact, the decidability result we have proved in this paper deal with the problem of checking the existence of a finite computation leading to a terminated state (resp. a state containing a molecule of a given species). In a stochastic setting this corresponds to checking whether a terminated state (resp. a state containing a molecule of a given species) can be reached with probability > 0 .

References

- N. Busi and R. Gorrieri. On the Computational Power of Brane Calculi. In *Transactions on Computational Systems Biology*, volume 4220 of *LNCS*, pages 16–43. Springer, 2006.
- N. Busi, M. Gabbriellini, and G. Zavattaro. On the Expressive Power of Recursion, Replication, and Iteration in Process Calculi. *To appear in Mathematical Structure in Computer Science*, 2009.
- L. Cardelli. On Process Rate Semantics. *Theoretical Computer Science*, 391(3):190–215, 2008.
- A. Credi, M. Garavelli, C. Laneve, S. Pradalier, S. Silvi, and G. Zavattaro. Modelization and Simulation of Nano Devices in nano-kappa Calculus. In *Proc. of Computational Methods in Systems Biology (CMSB07)*, volume 4695 of *LNCS*, pages 168–183, 2007.
- L. Cardelli and S. Pradalier. Where Membranes Meet Complexes. In *Proc. of Concurrent Models in Molecular Biology (BioConcur05)*, 2005.
- L. Cardelli and G. Zavattaro. On the Computational Power of Biochemistry. In *Proc. of Algebraic Biology (AB08)*, volume 5147 of *LNCS*, pages 65–80, 2008.
- V. Danos, J. Feret, W. Fontana, and J. Krivine. Kappa Factory, 2007. Available at: <http://www.lix.polytechnique.fr/~krivine/kappaFactory.html>.
- G. Delzanno, C. Di Giusto, M. Gabbriellini, C. Laneve, and G. Zavattaro. The κ -Lattice: Decidability Boundaries for Qualitative Analysis in Biological Languages. In *Proc. of Computational Systems in Systems Biology (CMSB09)*, volume 5688 of *LNCS*, pages 158–172, 2009.
- V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.
- J. Esparza and M. Nielsen. Decidability Issues for Petri Nets, 1994. Technical report BRICS RS-94-8.
- K.W. Kohn, M.I. Aladjem, J.N. Weinstein, and Y. Pommier. Molecular interaction maps of bioregulatory networks: a general rubric for systems biology. *Molecular biology of the cell*, 17(1):1–13, 2006.
- H. Kitano, A. Funahashi, Y. Matsuoka, and K. Oda. Using process diagrams for the graphical representation of biological networks. *Nature Biotechnology*, 23:961–966, 2005.
- A.M.L. Liekens and C.T. Fernando. Turing complete catalytic particle computers. In *Proc. of 9th European Conference on Artificial Life (ECAL07)*, volume 4648 of *Lecture Notes in Computer Science*, pages 1202–1211, 2007.
- R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
- R. Milner, J. Parrow, and D. Walker 1992. A calculus of mobile processes. *Journal of Information and Computation*, 100:1–77. Academic Press.
- M. L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Englewood Cliffs, 1967.
- S. Maffei and I. Phillips. On the computational strength of pure ambient calculi. *Theoretical Computer Science*, 330(3):501–551, 2005.
- C. Priami and P. Quaglia. Beta Binders for Biological Interactions. In *Proc. of Computational Methods in Systems Biology (CMSB04)*, volume 3082 of *Lecture Notes in Computer Science*, pages 20–33, 2004.

- C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80:25–31, 2001.
- W. Reisig. *Petri nets: an introduction*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.
- D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. Computation with Finite Stochastic Chemical Reaction Networks. *Natural Computing*, 7(4):615–633, 2008.
- J.-P. Sauvage and C. O. Dietrich-Buchecker (eds.). *Molecular Catenanes, Rotaxanes and Knots*. Wiley-VCH, Weinheim, 1999.
- G. Zavattaro and L. Cardelli. Termination Problems in Chemical Kinetics. In *Proc. of Concurrency Theory (CONCUR08)*, volume 5201 of *LNCS*, pages 477–491, 2008.