Core Formal Molecular Biology

Vincent Danos[1]* and Cosimo Laneve[2]

[1] Équipe PPS, University Paris 7& CNRS
[2] Department of Computer Science, University of Bologna

Abstract. A core modeling language for Molecular Biology is introduced, where two simple forms of interaction are considered, *complexation* and *activation*. This core language is equipped with two sensible bisimulation-based equivalences, and it is shown that interactions involving complex reactants are superfluous up to these notions. Strong compilations in π-calculus are given, following Regev's principle of translating physical connection as private name sharing.

1 Introduction

The potential interest of formal models in molecular biology has been recognized for some time in the community of bioinformatics. Post-genomic biology is producing ever bigger networks of reactions at the molecular level. Soon, exhaustive descriptions of entire simple living organisms will be obtained, down to the minutest molecular cogs and rods. Any assistance in investigating and understanding these is welcome.

The sorts of computational models best suited for this task remain unclear. Cases, sometimes very convincing, have been made for the use of various analytic frameworks, from boolean analysis of ordinary differential equations [18] to model-checking of hybrid systems [8].

We take here a different stance by constructing a calculus *specifically designed for the task* of representing and studying biological networks at the molecular level. While it might sound strange to the ears of a computer scientist that no such calculus was given before, one has to understand that it is quite a surprise for a biologist that there is one at all! It's a long way from one's first readings in molecular biology to the realization that, at the appropriate level of description, it all relies on very few principles of interaction: complexation, activation, synthesis and degradation.

The first contribution of this paper is therefore to provide such a calculus. Of course, there is no such calculus without a demonstration of biological expressivity. One has to make sure the formalism is enough to model a significant part of molecular biology. We propose a few examples in the text, merely to illustrate the syntax. Our calculus was tested against Kohn's compilation of the vast network of molecular reactions controlling the cell cycle [10,11]. A system of about

* This research was partly funded by the INRIA ARC CPBIO project.

P. Degano (Ed.): ESOP 2003, LNCS 2618, pp. 302–318, 2003.

300 reactions —after correcting some notational ambiguities— and representing the 'logical' part of the system was obtained in [2].

With the formalism in place we may take a step beyond pure representational problems. The second contribution of this paper is to construct an encoding showing that *complex reactants are useless* in a suitable mathematical sense. Interestingly, this theorem can be construed as raising a biological question. Any such interpretation will, of course, depend on the extent to which our formal biology is biological and not only formal! But this could be said of any model, and one of the interests of building a calculus at all, as opposed to a mere model, is to have a formal counterpart to the phenomenon that can be exposed to precise criticism.

The third contribution is to provide *compilations of our formalism into π-calculus*. These compilations were used to provide guidance in defining reactions in the calculus and could be useful to enrich the model as well. Regev, with her coauthors, outlined in a series of papers [16,14,15] an alarmingly convincing dictionary between elementary biological events: binding, activation, transcriptional regulation, etc and π-calculus constructions. We were particularly keen here on giving compilations respecting *Regev's principle* of representing physical connectivity, a primitive notion in our formalism, as private name sharing. For want of a formal language to describe what was to be represented, Regev's dictionary had to stay informal and could really be understood only by following examples. Encoding of a particular portion of biological knowledge was not straightforward, could be done in many ways and would generally result in a biologically illegible code. Our language and compilations provide a layer of abstraction obviating this.

There is scope for extending the formalism. It is easy to add synthesis (transcription factors included) and degradation reactions. Decomplexation rules can be added too, but with a purely qualitative non-deterministic semantics, such as the one we have in this paper, this is probably not going to make any difference. On the other hand, if one equips solutions with the structure of a Markovian process following [14], decomplexations will make a difference. We still have to develop more theory to understand whether complex reactants are reducible to simple ones in such a quantitative framework.

Then comes the pressing question of whether model-checking methods would let us query the dynamics of our systems. What kind of biologically relevant questions can one crank out of a CTL or mu-calculus approach [3,4,9], or other symbolic toolsets as advocated in [6], when applied to some model of formal reactions ? Queries one can think of, for the time being, would be reachability, and reachability under constraints. Constraints are particularly interesting here because they might allow the model to represent higher-level knowledge for which no efficient molecular mechanism is known or even conjectured, a sort of epistemological cheat code. Another pressing question is to develop a notion of pseudo-metrics on such probabilistic (or differential, see below) systems, which would give a reasonable formal counterpart to the evasive concept of homology for biological processes. Finally, another breed of operational semantics can

be imposed on the same calculus, namely continuous-time dynamics based on classical systems of multilinear differential equations (as explained for instance in [17]). One ought to understand when and how the stochastic and continuous-time deterministic operational semantics relate. This prompts us to remark that another advantage of having a calculus is to be able to plug in different dynamics calling on different tools and constraints. No matter how modest the contribution presented here, it is a preliminary step.

Related Work. A language, called "Pathway Logic", with quite similar concerns has been recently proposed by Lincoln and collaborators in [6]. Pathway Logic is a rewriting system formalism, where reactants are proteins and reactions are modeled by rewriting rules. As in κ, internal wirings of proteins are not modeled. The implementation of Pathway Logic in the Maude system may give access to useful analytic tools. This implementation is sequential, while our π calculus compilations exhibit at least the concurrency of the original biological systems, and are thus amenable to concurrent implementations.

Another language, the *biocalculus*, has been defined by Nagasaki and his colleagues [13]. The biocalculus is similar to Join [7], and therefore less abstract than the κ calculus —in fact, less intelligible for biologists— and could be used as a target language for compilation, similarly to π (see Section 5).

Acknowledgements. The first author gladly acknowledges numerous discussions about representation problems in systems biology with Marc Chiaverini, Magali Roux-Rouquié, Vincent Schächter, and with all the participants of the INRIA research action CPBIO.

2 The Calculus

Notations. We write $\widetilde{x}$ for a (possibly empty) finite sequence $x_1 \cdots x_n$ of names, and $|\widetilde{x}|$ for its length. Given a set S, $S^!$ will denote the set of multisets over S. We write $U + V$ for the multiset union, $V \subseteq U$ for the multiset inclusion and $V \subset U$ for the *strict* multiset inclusion; when $V \subseteq U$ we also write simply $U - V$ for the multiset $U \setminus V$. The empty multiset is denoted by $\varnothing$. The cardinality of a multiset U is denoted $|U|$. We use pattern matching to express set containment, *i.e.* $X = Y + \alpha$ means that $\alpha \subseteq X$. Variables X, Y range over multisets.

Sites, Proteins, and Signatures. We assume two countable sets of *sites* $\mathcal{N}$, and *protein names* $\mathcal{A}$ and a *signature* function $\mathfrak{s}(.) : \mathcal{A} \to \mathcal{N}^!$ from protein names to multiset of sites, mapping a protein name to its *static interface*. We ask further that $\mathfrak{s}(.)$ be countably onto, *i.e.* any static interface has an infinite number of associated protein names.

Sites and protein names will be ranged over by symbols such as r, s, t, and A, B, C, respectively. Sites will serve as both internal states and interfaces of the protein, through which the interactions between proteins may occur. We sort them accordingly in two separate infinite types, the *value* type $\mathcal{N}_v$ and the *proper* type $\mathcal{N}_p$.

A *formal protein*, or simply a *protein*, is a triple, written $A(\rho, \sigma)$, with $A \in \mathcal{A}$, and ρ, $\sigma \in \mathcal{N}^!$ such that $\rho + \sigma \subseteq \mathfrak{s}(A)$ and all sites in $\mathfrak{s}(A) - (\rho + \sigma)$ are of the proper type $\mathcal{N}_p$. In a given protein $A(\rho, \sigma)$, a site $r \in \mathfrak{s}(A)$ will be said *free* if $r \in \rho + \sigma$, *hidden* if $r \in \rho$, *visible* if $r \in \sigma$ and *bound* if not free.[1]

Take note that a site might be free and not visible, yet. The intended meaning is that, for instance, $A(\{r, s\}, \{t\})$ can interact at site t, and has also two other unoccupied sites r and s which are momentarily hidden; this could mean biologically that the particular folding A is assuming at the moment makes it impossible for another protein to dock on these sites, although they are *not* involved in any actual binding.

Complexes. Proteins may be composed into *protein complexes*, or simply *complexes* ranged over by C, D, ... and we denote the composition by the "$\odot$" operator, which is assumed to be associative and commutative (namely, the order of proteins inside the complexes is irrelevant). For instance, the complex

$$A_1(\rho_1, \sigma_1) \odot A_2(\rho_2, \sigma_2) \odot A_3(\rho_3, \sigma_3)$$

represents a compound made of A_1, A_2, and A_3. This notation is equivalent to $A_2(\rho_2, \sigma_2) \odot A_1(\rho_1, \sigma_1) \odot A_3(\rho_3, \sigma_3)$, as well as to any other permutation of A_1, A_2 and A_3.

Biologically, a complex is a bundle of proteins connected together by low energy bounds. Sometimes biologists are able to describe sub-components of the proteins which are instrumental in the binding. These are commonly called *domains*. Depending on the way proteins are folded in space, these domains can be active or not and the behaviour of the protein will be different. Both activations and complexations can modify the folding and therefore the subsequent behaviour. Sites are abstracting over both domains and folding states.

Well-formedness. A complex $A_1(\rho_1, \sigma_1) \odot \cdots \odot A_n(\rho_n, \sigma_n)$ is said to be *well-formed* if the following two conditions hold:

(a) for all $1 \leq i \leq n$, $\rho_i + \sigma_i \subset \mathfrak{s}(A_i)$, or $n = 1$ and $\rho_1 + \sigma_1 \subseteq \mathfrak{s}(A_1)$
(b) $\sum_{1 \leq i \leq n} |\mathfrak{s}(A_i) - (\rho_i + \sigma_i)| = 2m$, *for some* $m \geq n - 1$.

Both conditions work together to make sure that the complex is connected in the sense that there exists a wiring of the proteins participating in the complex, using all bound sites, that turns it into a connected graph. Condition (a) is *local*, and says that each protein must have at least one bound edge (unless $n = 1$), while condition (b) is *global* and says there is a big enough even number of bound edges, to connect the whole. All proteins and complexes will be taken to be well-formed from now on.

Both conditions are obviously necessary and an easy induction proves that they are also sufficient. So even if our complexation operator won't let us see the internal wiring, we still make sure that there is one.

[1] To be completely accurate, one should talk about an occurrence of $r \in \mathfrak{s}(A)$ since we're dealing with multisets here.

Examples. Consider the following complexes:

$$A(\{r,s\},\{r,t\}) \odot B(\varnothing,\varnothing) \qquad A(\{s\},\{r,t\})$$
$$A(\{s\},\{r,t\}) \odot B(\{t\},\varnothing) \odot B(\varnothing,\{t\})$$
$$B(\varnothing,\{t\}) \odot B(\varnothing,\{t\}) \odot B(\varnothing,\{t\}) \odot B(\varnothing,\{t\})$$

with $\mathfrak{s}(A) = \{2r,s,t\}$ and $\mathfrak{s}(B) = \{s,t\}$. All are ill-formed. Indeed, the first violates (a) at A, the second and the third violate the parity condition in (b), and the last doesn't have enough bound edges. On the contrary the following complexes are well-formed:

$$A(\{r,s\},\{r,t\}) \qquad A(\{r\},\{r,t\}) \odot B(\{s\},\varnothing) \qquad A(\varnothing,\{t\}) \odot B(\varnothing,\varnothing) \odot B(\varnothing,\{t\})$$

Solutions and Reactions. Solutions, ranged over by S, S', ... are multisets of proteins and complexes. *Reactions* are defined by rewriting rules which have the shape $\mathsf{S} \longrightarrow \mathsf{S}'$. Following the chemical metaphor further, we'll call complexes present in the left hand side of a given rule *reactants*, and complexes present in the right hand side *products* of the rule. To define reactions we use metavariables X, Y, Z which range over multisets of sites. There are two kinds of *basic reactions*:

$$\overset{\text{(ACTIVATION)}}{\mathsf{C}_1,\ldots,\mathsf{C}_n \longrightarrow \mathsf{C}'_1,\ldots,\mathsf{C}'_n} \qquad \overset{\text{(COMPLEXATION)}}{\mathsf{C}_1,\ldots,\mathsf{C}_n \longrightarrow \mathsf{C}'_1 \odot \ldots \odot \mathsf{C}'_n}$$

and we ask for the following *conservation principles*, for activations:

$$\begin{array}{l}\text{if } \mathsf{C}_i = \bigodot_{1\le j\le k_i} A_{ij}(X_{ij}+\rho_{ij}, Y_{ij}+\sigma_{ij}), \text{ and}\\ \mathsf{C}'_i = \bigodot_{1\le j\le k_i} A_{ij}(X_{ij}+\rho'_{ij}, Y_{ij}+\sigma'_{ij}),\\ \text{then, for every } j,\ \rho_{ij}+\sigma_{ij} = \rho'_{ij}+\sigma'_{ij},\end{array}$$

and for complexations:

$$\begin{array}{l}\text{if } \mathsf{C}_i = \bigodot_{1\le j\le k_i} A_{ij}(X_{ij}+\rho_{ij}, Y_{ij}+\sigma_{ij}+\tau_{ij}), \text{ and}\\ \mathsf{C}'_i = \bigodot_{1\le j\le k_i} A_{ij}(X_{ij}+\rho'_{ij}, Y_{ij}+\sigma'_{ij}), \text{ and } x_i = \sum_{1\le j\le k_i} |\tau_{ij}|,\\ \text{then, for every } j,\ \rho_{ij}+\sigma_{ij} = \rho'_{ij}+\sigma'_{ij}, \text{ and } x_i > 0, \text{ and}\\ \sum_{1\le i\le n} x_i = 2m \text{ for some } m \ge n-1.\end{array}$$

The condition for activation says that free sites may flip their visible/hidden status, but may not be bound by the reaction. The condition for complexation completely mirrors the well-formedness conditions (a) and (b) above and controls that the final product of the complexation rule:

$$\bigodot_{1\le i\le n, 1\le j\le k_i} A_{ij}(X_{ij}+\rho'_{ij}, Y_{ij}+\sigma'_{ij})$$

stays well-formed, knowing that the reactants were already so. We also ask the newly bound sites, namely the τ_{ij}, to be visible in the left hand side.

We could distinguish the case of self-complexation, where $n = 1$ and allow $\tau_1 = \varnothing$ but then self-complexation becomes exactly self-activation, so it doesn't make any difference and it feels better to have two disjoint sets of rules.

A basic reaction will be said *simple* if for all i, $k_i = 1$, that is to say if all the reactants are proteins; otherwise it will be said *complex*.

$$\text{(CTX-ACT)}\quad \frac{\mathsf{C}_1,\ldots,\mathsf{C}_n \longrightarrow \mathsf{C}'_1,\ldots,\mathsf{C}'_n}{\mathsf{D}_1 \odot \mathsf{C}_{i_1}\cdots\mathsf{C}_{i_{j_1}},\ldots,\mathsf{D}_m \odot \mathsf{C}_{i_m}\cdots\mathsf{C}_{i_{j_m}} \longrightarrow \mathsf{D}_1 \odot \mathsf{C}'_{i_1}\cdots\mathsf{C}'_{i_{j_1}},\ldots,\mathsf{D}_m \odot \mathsf{C}'_{i_m}\cdots\mathsf{C}'_{i_{j_m}}}$$

$$\text{(CTX-CMP)}\quad \frac{\mathsf{C}_1,\ldots,\mathsf{C}_n \longrightarrow \mathsf{C}'_1 \odot\cdots\odot \mathsf{C}'_n}{\mathsf{D}_1 \odot \mathsf{C}_{i_1}\cdots\mathsf{C}_{i_{j_1}},\ldots,\mathsf{D}_m \odot \mathsf{C}_{i_m}\cdots\mathsf{C}_{i_{j_m}} \longrightarrow \mathsf{D}_1 \odot \mathsf{C}'_{i_1}\cdots\mathsf{C}'_{i_{j_1}} \odot\cdots\odot \mathsf{D}_m \odot \mathsf{C}'_{i_m}\cdots\mathsf{C}'_{i_{j_m}}}$$

Fig. 1. Contextual Reactions

Contextual Reactions. Basic transitions are made to act on generic solutions by means of suitable contextual rules. We need two such rules, (CTX-ACT) and (CTX-CMP), to define the action of a basic rule when the reactants are themselves embedded in super-complexes D_1, ..., D_m. In this sense these rules are allowing 'contextual reactions' where the context is precisely the multiset D_1, ..., D_m of super-complexes hosting the reactants of the basic rules. The rules are given in Fig. 1, where $\{i_1,\ldots,i_{j_1}\},\ldots,\{i_m,\ldots,i_{j_m}\}$ stands for any partition of $\{1,\ldots,n\}$. Take note that when one uses a partition which is not discrete, that is when $m < n$, then super-complexes hold more than one reactants. For instance, from the basic $\mathsf{C}_1,\mathsf{C}_2 \longrightarrow \mathsf{C}'_1,\mathsf{C}'_2$ one can derive $\mathsf{C}_1 \cdot \mathsf{C}_2 \longrightarrow \mathsf{C}'_1 \cdot \mathsf{C}'_2$. This is going to be crucial in the sequel. It also shows that our rules are not standard multiset rewriting rules; they're actually mixing two levels of multisets, the solution level associated to the operator "," and the complex level associated to the other operator "$\odot$".

Contextual reactions are readily seen to preserve well-formedness.

Systems. To complete the construction of our κ-calculus, we define a *κ-system* $\mathfrak{S}$ to be a pair $\langle \mathsf{S},\mathsf{R}\rangle$ where S is a solution and R is a finite set of basic reactions. And we define also a transition relation over such κ-systems by $\langle(\mathsf{S}_1,\mathsf{S}),\mathsf{R}\rangle \longrightarrow \langle(\mathsf{S}_2,\mathsf{S}),\mathsf{R}\rangle$, if $\mathsf{S}_1 \longrightarrow \mathsf{S}_2$ according to some reaction, basic or contextual, that can be derived from R.

Most of the time, we will write $\mathsf{S} \longrightarrow_{\mathsf{R}} \mathsf{S}'$ instead of $\langle \mathsf{S},\mathsf{R}\rangle \longrightarrow \langle \mathsf{S}',\mathsf{R}\rangle$, or even better $\mathsf{S} \longrightarrow \mathsf{S}'$, when it's clear from the context which R is meant. We also write $\longrightarrow^*_{\mathsf{R}}$ for the reflexive transitive closure of $\longrightarrow_{\mathsf{R}}$.

Core κ-calculus. An interesting fragment of κ, the *core κ-calculus*, is obtained by restricting to simple systems, *i.e.* systems with no complex reactants in their basic rules. The simplest reaction which is not in the core calculus is:

$$A(\varnothing,\{r_1,r_2\}),B_1(\varnothing,\{r'_1\})\cdot B_2(\varnothing,\{r'_2\}) \longrightarrow A(\varnothing,\varnothing)\cdot B_1(\varnothing,\varnothing)\cdot B_2(\varnothing,\varnothing).$$

We'll see in section 4 that, despite what seems a strong restriction, the core calculus is expressive enough to encode κ; but before, we're going to have a few examples and comments.

3 Discussion

Activation. The reader might ask how activation is implemented biologically. It could be a phosphorylation (or the converse reaction of dephosphorylation), a reaction whereby a protein is given (or taken) a phosphate group. Some change of conformation might then occur in the structure possibly revealing formerly hidden sites and hiding formerly visible ones:

$$\overset{\text{(BIO-ACTIVATION)}}{A(X+\rho, Y+\sigma), B(X', Y') \longrightarrow A(X+\rho', Y+\sigma'), B(X', Y')}$$

usually B is called a kinase (or the converse a phosphatase). Our activations are more general, but there doesn't seem to be any theoretical reward in restricting activations to have a closer fit.

Wiring doesn't matter? By hiding the complexes internal wirings, we seem to be assuming that the potential interactions of a given complex are not depending on the underlying graph, which happens to have been used to construct the complex. This is not true however. In case connections of a complex are making a difference regarding interaction, one may account for these differences using some sites as internal states. That said, dealing with explicit graphs and graph-rewriting reactions is an interesting option which we investigated in [5].

To illustrate further choices built in the formalism, we go now through a few small molecular stories (all true, although as usual the names have been changed).

Commuting and Competing behaviours. Suppose B is able to bind A, revealing a site r in A, and, independently, to bind C, then we write:

$$\begin{aligned}\mathfrak{r}_1 &= A(X+r, Y+s), B(X', Y'+s') \longrightarrow A(X, Y+r) \odot B(X', Y')\\ \mathfrak{r}_2 &= B(X, Y+t), C(X', Y'+t') \longrightarrow B(X, Y) \odot C(X', Y')\end{aligned}$$

From an initial solution $\mathsf{S} = A(\{r\}, \{s\}), B(\varnothing, \{s', t\}), C(\varnothing, \{t'\})$ one gets two paths to the same final solution:

$$\begin{aligned}\mathsf{S} &\longrightarrow A(\varnothing, \{r\}) \odot B(\varnothing, \{t\}), C(\varnothing, \{t'\}) \quad \text{by } \mathfrak{r}_1\\ &\longrightarrow A(\varnothing, \{r\}) \odot B(\varnothing, \varnothing) \odot C(\varnothing, \varnothing) \quad \text{by } \mathfrak{r}_2 \text{ and (CTX-CMP)}\end{aligned}$$

$$\begin{aligned}\mathsf{S} &\longrightarrow A(\{r\}, \{s\}), B(\varnothing, \{s'\}) \odot C(\varnothing, \varnothing) \text{ by } \mathfrak{r}_2\\ &\longrightarrow A(\varnothing, \{r\}) \odot B(\varnothing, \varnothing) \odot C(\varnothing, \varnothing) \quad \text{by } \mathfrak{r}_1 \text{ and (CTX-CMP)}\end{aligned}$$

If, on the contrary, bindings to B are competing for the same binding site s' in B, we modify our second basic rule:

$$\mathfrak{r}_2' = B(X, Y+s'), C(X', Y'+t') \longrightarrow B(X, Y) \odot C(X', Y')$$

and starting from the same S, we get a 'deadlocked' solution:

$$\mathsf{S} \longrightarrow A(\varnothing, \{r\}) \cdot B(\varnothing, \{t\}), C(\varnothing, \{t'\})$$

because B has spent s' by binding with A. So both commuting and competing behaviours are expressible. In the latter case, when $A \odot B$, $A \odot C$ both exist, but $A \odot B \odot C$ doesn't, other mechanisms can be invoked and expressed. Instead of having A and C competing for the same docking site on B, A could be obstructing the site targeted by C, by moving that site into the hidden part of B's interface.

Synchronizing behaviours. So far, only simple reactants were used, but the real interest of complex formation is that some further event can be conditioned on two sites to be present on the same complex. For instance, some C' could bind only to A and B at the same time, and it easy to add a complex rule:

$$\begin{aligned}\mathfrak{r} = {} & A(X, Y + r) \cdot B(X', Y' + t), C'(X'', Y'' + \{r', t'\}) \\ & \longrightarrow A(X, Y) \cdot B(X', Y') \cdot C'(X'', Y'')\end{aligned}$$

and get:

$$\begin{aligned}A(\{r\}, \{s\}), B(\varnothing, \{s', t\}), C'(\varnothing, \{r', t'\}) & \xrightarrow{\mathfrak{r}_1} A(\varnothing, \{r\}) \odot B(\varnothing, \{t\}), C'(\varnothing, \{r', t'\}) \\ & \xrightarrow{\mathfrak{r}} A(\varnothing, \varnothing) \odot B(\varnothing, \varnothing) \odot C(\varnothing, \varnothing)\end{aligned}$$

So we can express competition, commutation and even some form of synchronization, and it seems natural to bring process algebra concepts in the picture.

4 Core κ-Calculus Is Enough

We've just said, in the example above, that complexes were useful. One may wonder whether they really are, formally speaking. In the following we show the existence of a non-uniform encoding $[\![\cdot]\!]$ of a complex biological system into a simple one. The encoding is non-uniform, namely $[\![\mathsf{S}, \mathsf{S}']\!] \neq [\![\mathsf{S}]\!], [\![\mathsf{S}']\!]$, because it depends on the initial solution. We suspect that uniform encodings of complex systems might not always exist.

To begin with, we introduce an extensional semantics, called *barbed bisimulation* [12], which equates systems if their transitions match and they are indistinguishable under global observations. The key of barbed bisimulation is the notion of observation. We provide two such notions, which define two different, uncomparable bisimulations.

Let a, b, c, $\cdots$ be a countable set of names—the *observations*—, and f be an injective mapping from protein names $\mathcal{A}$ to names. Let's remind that sites $\mathcal{N}$ are partitioned into *proper* sites $\mathcal{N}_p$ and value sites $\mathcal{N}_v$.

Definition 1. *The* complex barb $\downarrow_c$ *is the least relation satisfying the rules below.*

$$\begin{array}{ll} A(\rho, \sigma) \downarrow_c a & \text{if } f(A) = a \\ \mathsf{A}_1 \cdots \mathsf{A}_n \downarrow_c a_1 \cdots a_n & \text{if } f(A_i) = a_i \text{ for every } i \\ \mathsf{S}, \mathsf{S}' \downarrow_c \widetilde{a} & \text{if } \mathsf{S} \downarrow_c \widetilde{a} \text{ or } \mathsf{S}' \downarrow_c \widetilde{a} \end{array}$$

The simple barb $\downarrow_s$ *is the least relation satisfying the rules below.*

$$\begin{array}{ll} A(\rho,\sigma) \downarrow_s a & \text{if } \sigma \setminus \mathcal{N}_v \neq \varnothing \text{ and } f(A) = a \\ \mathsf{A}_1 \cdots \mathsf{A}_n \downarrow_s a & \text{if there is an } i \text{ such that } \mathsf{A}_i \downarrow_s a \\ \mathsf{S}, \mathsf{S}' \downarrow_s a & \text{if } \mathsf{S} \downarrow_s a \text{ or } \mathsf{S}' \downarrow_s a \end{array}$$

We write $\mathsf{S} \Downarrow_c \widetilde{a}$ *if* $\mathsf{S} \longrightarrow^* \mathsf{T}$ *and* $\mathsf{T} \downarrow_c \widetilde{a}$*, and* $\mathsf{S} \Downarrow_s a$ *if* $\mathsf{S} \longrightarrow^* \mathsf{T}$ *and* $\mathsf{T} \downarrow_s a$*.*

Definition 2. *A* (weak) barbed bisimulation *is a symmetric binary relation* $\mathcal{R}$ *on* κ*-systems such that if* $\langle \mathsf{S}, \mathsf{R} \rangle \ \mathcal{R} \ \langle \mathsf{S}', \mathsf{R}' \rangle$ *then:*

1. *If* $\mathsf{S} \longrightarrow_{\mathsf{R}} \mathsf{T}$ *then, for some* T'*,* $\mathsf{S}' \longrightarrow^*_{\mathsf{R}'} \mathsf{T}'$ *and* $\langle \mathsf{T}, \mathsf{R} \rangle \ \mathcal{R} \ \langle \mathsf{T}', \mathsf{R}' \rangle$*.*
2. *If* $\mathsf{S} \downarrow a$*, then* $\mathsf{S}' \Downarrow a$*.*

Depending on whether $\downarrow$ *and* $\Downarrow$ *are* $\downarrow_c$*,* $\Downarrow_c$*, or* $\downarrow_s$*,* $\Downarrow_s$*, two* κ*-systems* $\mathfrak{S}$ *and* $\mathfrak{T}$ *will be said* complex (resp. simple) barbed bisimilar*, written* $\mathfrak{S} \dot{\approx}_c \mathfrak{T}$ *(resp.* $\mathfrak{S} \dot{\approx}_s \mathfrak{T}$*), if there exists a barbed bisimulation relating* $\mathfrak{S}$ *and* $\mathfrak{T}$*.*

We are ready to state our reducibility theorem:

Theorem 1. *Every* κ*-system is barbed bisimilar—both complex and simple—to a core* κ*-system.*

We explain the idea informally. Reactions with a complex reactant, say C, have to check whether each participant of C is present in a same supercomplex, and each with the appropriate internal state that would trigger the reaction. We have to achieve this synchronisation effect with 'bare hands' if complex reactants are forbidden. The gist of the proof is to use sites of value type to make elementary components 'aware' of the states of the partners they are in physical contact with, at any given moment. Both bisimulations will be blind to such manoeuvring since none is observing internal states—the value sites $\mathcal{N}_v$—and activation rules will provide the means to keep this local information consistent.

5 Compilations into Asynchronous π-Calculus

In this section we detail the compilation of core κ into asynchronous π-calculus [1]. We'll introduce an intermediate equivalent calculus κ_b with *explicit wirings*, that makes it easier to pin down the formal relationship with π-calculus.

We begin with a brief introduction to asynchronous π-calculus, then we introduce κ_b and prove it is equivalent to core κ, and from there we proceed to the compilations of κ_b into asynchronous π-calculus. We present two such compilations: the first fits with the standard pattern of encoding of chemical machines in process calculi, as in Join-calculus [7]; the second one follows Regev's idea that protein behaviours are described within the proteins [16].

5.1 The Asynchronous π-Calculus

The asynchronous π-calculus relies on three countable sets: 1) names, ranged over by a, b, c, x, y, z, ... 2) agent names, ranged over by $\mathtt{A}$, $\mathtt{B}$, ... and 3) variables X, Y, Z, ...; we use u, v to range over values, which may be names, natural numbers, tuples, written $\widetilde{u}$, or finite sets of names. Variables, which include names, represent formal parameters. Even if no explicit type system is given, our programs will be typable. Two syntactic categories are defined, *processes* written P and *boolean expressions* written M:

$$P \stackrel{def}{=} \mathbf{0} \mid \overline{x}\langle\widetilde{u}\rangle \mid \textstyle\sum_{i\in I} x_i(\widetilde{Z_i}).P_i \mid P \mid P \mid (x)P \mid M\,P\,;\,P \mid \mathtt{A}(\widetilde{u})$$
$$M \stackrel{def}{=} [u = v] \mid [u \subseteq v] \mid MM$$

We see that a process P can be the inert process $\mathbf{0}$; a message $\overline{x}\langle\widetilde{u}\rangle$; a nondeterministic input $\sum_{i\in I} x_i(\widetilde{Z_i}).P_i$, where we assume I finite; a parallel composition of processes; a restricted process of the form $(x)P$ where (x) is the 'new' operator that limits the scope of x to P; a conditional $M\,P\,;\,Q$; or a defined agent $\mathtt{A}(\widetilde{u})$, and we ask for a unique defining equation $\mathtt{A}(\widetilde{X}) \stackrel{def}{=} P$ for each agent identifier.

Both restriction and input are binders: $(x)P$ binds name x in P and $x(\widetilde{Z}).P$ binds the variables $\widetilde{Z}$ in P. Names of P which are not bound are called *free* and we write $\mathrm{fn}(P)$ for the set of such names.

The conditional $M\,P\,;\,Q$ evaluates to P or Q depending on whether M is true or not. In our programs, M will be either $[u = v]$ or $[u \subseteq v]$ or sequences of such tests. We assume the existence of an evaluation function for terms in M which gives booleans, which we leave unspecified. Table 1 collects the semantics of π-calculus: there are three basic reductions which are then lifted to parallel contexts and scope contexts, up to the structural congruence defined above.

We consider a standard extensional semantics of π-calculus: (weak) barbed equivalence, written $\dot{\approx}$. The definition parallels definition 2, with processes instead of solutions, and we only need to redefine the notion of observation.

Definition 3. *The* observation relation *on π-calculus processes is the least relation satisfying the rules below.*

$$\overline{a}\langle\widetilde{u}\rangle \downarrow a$$
$$P \mid Q \downarrow a \qquad \text{if } P \downarrow a \text{ or } Q \downarrow a$$
$$(x)P \downarrow a \qquad \text{if } x \neq a \text{ and } P \downarrow a$$
$$\mathtt{A}(\widetilde{u}) \downarrow a \qquad \text{if } \mathtt{A}(\widetilde{X}) \stackrel{def}{=} P \text{ and } P\{\widetilde{u}/\widetilde{X}\} \downarrow a$$

As usual, we write $P \Downarrow a$ if $P \longrightarrow^* Q$ and $Q \downarrow a$.

5.2 The κ_b-Calculus

Proteins and complexes. Besides *sites* $\mathcal{N}$, and *protein names* $\mathcal{A}$, we need a further disjoint set of names $\mathcal{V}$, called *connections*, and ranged over by x, y, z, ..., which are sorted by sites so as to have countably many connections for any given site.

Table 1. Operational semantics of the asynchronous π-calculus.

Structural congruence:

$$P \mid Q \equiv Q \mid P, \quad (P \mid Q) \mid R \equiv P \mid (Q \mid R), \quad P \mid \mathbf{0} \equiv P,$$
$$(x)\mathbf{0} \equiv \mathbf{0}, \quad (x)(y)P \equiv (y)(x)P,$$
$$(x)MP; Q \equiv M(x)P; (x)Q \quad \textit{if } x \notin \mathrm{fn}(M)$$
$$P \mid (z)Q \equiv (z)(P \mid Q), \quad \textit{if } z \notin \mathrm{fn}(P)$$
$$\mathtt{A}(\widetilde{u}) \equiv P\{\widetilde{u}/\widetilde{X}\} \qquad \textit{if } \mathtt{A}(\widetilde{X}) \stackrel{def}{=} P$$

Basic reductions:

$$\overline{x}\,\langle\widetilde{u}\rangle \mid \textstyle\sum_{i\in I} x_i\,(\widetilde{Y_i}).P_i \longrightarrow P_j\{\widetilde{u}/\widetilde{Y_j}\} \qquad (\textit{if } x = x_j)$$
$$M\,P;\,Q \longrightarrow P \quad (\textit{if } M \textit{ is true}) \qquad M\,P;\,Q \longrightarrow Q \quad (\textit{if } M \textit{ is false})$$

Contextual reductions:

$$\frac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q} \qquad \frac{P \longrightarrow P'}{(x)P \longrightarrow (x)P'} \qquad \frac{P \equiv P' \quad P' \longrightarrow Q' \quad Q' \equiv Q}{P \longrightarrow Q}$$

We need to adapt the definitions of Section 2. Protein names now carry three arguments: two multisets of sites ρ and σ, as before in κ, and an additional set ξ of connections. The connection set ξ represents the bound sites of a protein: every site in it is represented by a different connection. Let $\mathfrak{s}(\xi)$ be the multiset of sorts of the connections in ξ. We demand that a protein $A(\rho, \sigma, \xi)$ is such that $\mathfrak{s}(A) - (\rho + \sigma) = \mathfrak{s}(\xi)$ and also that for every complex $A_1(\rho_1, \sigma_1, \xi_1) \odot \cdots \odot A_n(\rho_n, \sigma_n, \xi_n)$:

1. for all $1 \leq i \leq n$, $\rho_i + \sigma_i \subset \mathfrak{s}(A_i)$, or $n = 1$ and $\rho_1 + \sigma_1 \subseteq \mathfrak{s}(A_1)$;
2. connections occur exactly twice in $\xi_1 + \ldots + \xi_n$.

To express the fact that names in the ξ arguments are local to the complex, we borrow the 'new' operator from π-calculus. So the general shape of a complex is $(\xi)A_1(\rho_1, \sigma_1, \xi_1) \odot \cdots \odot A_n(\rho_n, \sigma_n, \xi_n)$.

Solutions and Reactions. Solutions in κ_b must satisfy the additional constraint that *complexes have pairwise disjoint connection sets.*

We have now to adapt our basic reactions to handle correctly the connection sets. Activations don't modify connections:

(ACTIVATION)

$$\begin{array}{c} A_1(X_1 + \rho_1, Y_1 + \sigma_1, Z_1) \\ , \ldots , \\ A_n(X_n + \rho_n, Y_n + \sigma_n, Z_n) \end{array} \longrightarrow \begin{array}{c} A_1(X_1 + \rho'_1, Y_1 + \sigma'_1, Z_1) \\ , \ldots , \\ A_n(X_n + \rho'_n, Y_n + \sigma'_n, Z_n) \end{array}$$

But complexations, by definition, need to create new connections, and these have to be *fresh names* in the ambient solution:

(COMPLEXATION)

$$\begin{array}{c} A_1(X_1+\rho_1, Y_1+\sigma_1+\tau_1, Z_1) \\ ,\ldots, \\ A_n(X_n+\rho_n, Y_n+\sigma_n+\tau_n, Z_n) \end{array} \longrightarrow \begin{array}{c} (\xi_1 \ldots \xi_n) \\ A_1(X_1+\rho'_1, Y_1+\sigma'_1, Z_1+\xi_1) \\ \odot \cdots \odot \\ A_n(X_n+\rho'_n, Y_n+\sigma'_n, Z_n+\xi_n) \end{array}$$

Both kind of rules are subject to the side-conditions that $\rho_i+\sigma_i = \rho'_i+\sigma'_i$, and for complexations, we also demand that $\tau_i \neq \varnothing$, $\mathfrak{s}(\xi_i) = \tau_i$. Since the product of the reaction has to be a correct complex, new connections $\bigcup_{1\leq i\leq n} \xi_i$ do not clash with existing connections in $\bigcup_{1\leq i\leq n} Z_i$, and we observe that the 'big enough even number' side-condition is also taken care of automatically.

To move the new operator upwards, once it is created, we add the structural congruence: $\mathsf{S}, (\xi)\mathsf{S}' \equiv (\xi)(\mathsf{S}, \mathsf{S}')$, when ξ is not free in S. Additionally, we adapt rule (CTX-CMP) in Figure 1 such that names created in the subcomplex are also new in the host supercomplex. With these adaptations done, transitions between κ_b-systems are defined up to structural congruence and with the help of the following additional form of contextual reaction which we need for handling the 'new': $\mathsf{S} \longrightarrow \mathsf{T}$ implies $(\xi)\mathsf{S} \longrightarrow (\xi)\mathsf{T}$.

Equivalence with core κ. The notions of complex and simple barbs of Definition 1 can be easily extended to κ_b by discarding connections ξ. There is an observation preserving function U which maps a κ_b-solution to a κ-solution just by dropping the connections. This 'forgetful' map U is left inverse to a family of converse maps which make explicit the connections of the underlying graphs of complexes. Of course, there isn't a unique right inverse to U because many choices of internal wirings are possible for each complex.

Since reactions are not 'seeing' the internal wiring, any choices will end up to be complex and simple bisimilar. One easily sees that U commutes to transitions, and so does any of the converse maps. Therefore (with the obvious adaptation of weak barbed bisimilarity to include κ_b systems):

Proposition 1. *Every κ_b-system is barbed bisimilar—both complex and simple—to a core κ-system, and conversely.*

5.3 First Compilation: Proteins Encode States, Rules Are Outside

This first compilation is 'reaction-centric', proteins play a passive role and just encode the state of the solution. Contextual reactions are taken care of by contextual reductions in π, there is no need to translate them.

First, to each protein $A(\rho, \sigma, \xi)$, an agent $\mathtt{A}(\rho, \sigma, \xi)$ is associated with the following behaviour (we use the name $a = f(A)$, as defined in Section 4):

$$\mathtt{A}(\rho, \sigma, \xi) \stackrel{def}{=} [\sigma \setminus \mathcal{N}_v = \varnothing]\mathbf{0}\,;\, (x)(\overline{a}\,\langle \rho, \sigma, x\rangle \mid x\,(\rho', \sigma', \xi').\mathtt{A}(\rho', \sigma', \xi + \xi'))$$

By writing $\sigma \setminus \mathcal{N}_v$ we are cheating a little bit, because there is no such operation in the π-calculus of Section 5.1. But, we could easily implement this operation by partitioning σ in two different arguments of $\mathtt{A}$, and keeping track of this difference in messages which exchange σ.

Second, given an activation reaction $\mathfrak{a}$:

$$A_1(X_1+\rho_1, Y_1+\sigma_1, Z_1), \cdots, A_n(X_n+\rho_n, Y_n+\sigma_n, Z_n) \longrightarrow \\ A_1(X_1+\rho'_1, Y_1+\sigma'_1, Z_1), \cdots, A_n(X_n+\rho'_n, Y_n+\sigma'_n, Z_n)$$

we define the agent $[\![\mathfrak{a}]\!] \stackrel{def}{=} (z)(\mathfrak{A}_1(\mathbf{0},\mathbf{0}) \mid \overline{z}\,\langle\rangle)$ with:

$$\begin{array}{l}\mathfrak{A}_1(P,Q) \stackrel{def}{=} a_1\,(\ell'_1, \ell''_1, x_1).\\ \qquad [\rho_1 \subseteq \ell'_1, \sigma_1 \subseteq \ell''_1]\\ \qquad\quad \mathfrak{A}_2(P \mid \overline{x_1}\,\langle(\ell'_1 \setminus \rho_1)+\rho'_1, (\ell''_1 \setminus \sigma_1)+\sigma'_1, \varnothing\rangle, Q \mid \overline{a_1}\,\langle \ell'_1, \ell''_1, x_1\rangle)\,;\\ \qquad\quad (Q \mid \overline{a_1}\,\langle \ell'_1, \ell''_1, x_1\rangle \mid \mathfrak{A}_1(\mathbf{0},\mathbf{0}))\\ \qquad +z\,().(Q \mid [\![\mathfrak{a}]\!])\end{array}$$

and similar definitions for $\mathfrak{A}_i$, with $i < n$, and:

$$\begin{array}{l}\mathfrak{A}_n(P,Q) \stackrel{def}{=} a_n\,(\ell'_n, \ell''_n, x_n).\\ \qquad\quad ([\rho_n \subseteq \ell'_n, \sigma_n \subseteq \ell''_n](P \mid \overline{x_n}\,\langle \ell'_n \setminus \rho_n)+\rho'_n, (\ell''_n \setminus \sigma_n)+\sigma'_n, \varnothing\rangle)\,;\\ \qquad\quad (Q \mid \overline{a_n}\,\langle \ell'_n, \ell''_n, x_n\rangle) \mid \mathfrak{A}_1(\mathbf{0},\mathbf{0}))\\ \qquad +z\,().(Q \mid [\![\mathfrak{a}]\!])\end{array}$$

Nondeterminism in the $\mathfrak{A}_i$s is introduced to avoid deadlocks which could occur when not all of $\mathfrak{a}$'s reactants are present and free to use.

Last, given a complexation reaction $\mathfrak{c}$:

$$A_1(X_1+\rho_1, Y_1+\sigma_1+\tau_1, Z_1), \cdots, A_n(X_n+\rho_n, Y_n+\sigma_n+\tau_n, Z_n) \longrightarrow \\ (\xi_1 \ldots \xi_n)\, A_1(X_1+\rho'_1, Y_1+\sigma'_1, Z_1+\xi_1) \odot \cdots \odot A_n(X_n+\rho'_n, Y_n+\sigma'_n, Z_n+\xi_n)$$

we define the agent $[\![\mathfrak{c}]\!] \stackrel{def}{=} (z)(\mathfrak{C}_1(\mathbf{0},\mathbf{0}) \mid \overline{z}\,\langle\rangle)$, with:

$$\begin{array}{l}\mathfrak{C}_1(P,Q) \stackrel{def}{=} (\xi_1)\, a_1\,(\ell'_1, \ell''_1, x_1).\\ \qquad [\rho_1 \subseteq \ell'_1, \sigma_1+\tau_1 \subseteq \ell''_1]\\ \qquad\quad \mathfrak{C}_2(P \mid \overline{x_1}\,\langle(\ell'_1 \setminus \rho_1)+\rho'_1, (\ell''_1 \setminus (\sigma_1+\tau_1))+\sigma'_1, \xi_1\rangle, Q \mid \overline{a_1}\,\langle \ell'_1, \ell''_1, x_1\rangle)\,;\\ \qquad\quad (Q \mid \overline{a_1}\,\langle \ell'_1, \ell''_1, x_1\rangle \mid \mathfrak{C}_1(\mathbf{0},\mathbf{0}))\\ \qquad +z\,().(Q \mid [\![\mathfrak{c}]\!])\end{array}$$

and similar definitions for $\mathfrak{C}_i$, with $i < n$, and:

$$\begin{array}{l}\mathfrak{C}_n(P,Q) \stackrel{def}{=} (\xi_n)\, a_n\,(\ell'_n, \ell''_n, x_n).\\ \qquad\quad ([\rho_n \subseteq \ell'_n, \sigma_n+\tau_n \subseteq \ell''_n]\\ \qquad\quad\quad (P \mid \overline{x_n}\,\langle(\ell'_n \setminus \rho_n)+\rho'_n, (\ell''_n \setminus (\sigma_n+\tau_n))+\sigma'_n, \xi_n\rangle);\\ \qquad\quad\quad (Q \mid \overline{a_n}\,\langle \ell'_n, \ell''_n, x_n\rangle)) \mid \mathfrak{C}_1(\mathbf{0},\mathbf{0})\\ \qquad +z\,().(Q \mid [\![\mathfrak{c}]\!])\end{array}$$

Observe that biological connections are explained in π-calculus in terms of bound names. In particular, two proteins are connected if they share a same bound name. We give now the action of $[\![\cdot]\!]$ on solutions and complexes:

$$\begin{array}{c}[\![\mathsf{S}, \mathsf{S}']\!] = [\![\mathsf{S}]\!] \mid [\![\mathsf{S}']\!]\\ [\![(\xi)\,\mathsf{S}']\!] = (\xi)\,[\![\mathsf{S}]\!]\\ [\![A_1(\rho_1,\sigma_1,\xi_1) \odot \cdots \odot A_n(\rho_n,\sigma_n,\xi_n)]\!] = (\xi_1 \ldots \xi_n)\,(\mathtt{A}_1(\rho_1,\sigma_1,\xi_1) \mid \cdots \mid \mathtt{A}_\mathtt{n}(\rho_n,\sigma_n,\xi_n))\end{array}$$

Lemma 1. *Let* $\langle \mathsf{S}, \mathsf{R} \rangle$ *be a* κ_b*-system and set* $[\![\mathsf{R}]\!] = \prod_{\mathfrak{r} \in \mathsf{R}} [\![\mathfrak{r}]\!]$*, then:*

1. *(simulation) If* $\mathsf{S} \longrightarrow_{\mathsf{R}} \mathsf{S}'$ *then* $[\![\mathsf{S}]\!] \mid [\![\mathsf{R}]\!] \longrightarrow^* [\![\mathsf{S}']\!] \mid [\![\mathsf{R}]\!]$.
2. *(catch-up) If* $[\![\mathsf{S}]\!] \mid [\![\mathsf{R}]\!] \longrightarrow^* Q$ *then* $Q \longrightarrow^* [\![\mathsf{S}']\!] \mid [\![\mathsf{R}]\!]$ *with* $\mathsf{S} \longrightarrow^*_{\mathsf{R}} \mathsf{S}'$.

Next observe that, $\mathsf{S} \downarrow_s a$ if and only if $[\![\mathsf{S}]\!] \mid [\![\mathsf{R}]\!] \downarrow a$, so the translation respects simple barbs. From there it is possible to deduce that (S, R) and $[\![\mathsf{S}]\!] \mid [\![\mathsf{R}]\!]$ are weakly bisimilar in their respective transition systems, and by transitivity, this gives a full abstraction result with respect to barbed equivalence:

Theorem 2. *Let* $\langle \mathsf{S}, \mathsf{R} \rangle$ *and* $\langle \mathsf{S}', \mathsf{R}' \rangle$ *be* κ_b*-systems, then* $\langle \mathsf{S}, \mathsf{R} \rangle \mathrel{\dot{\approx}_s} \langle \mathsf{S}', \mathsf{R}' \rangle$ *iff* $[\![\mathsf{S}]\!] \mid [\![\mathsf{R}]\!] \mathrel{\dot{\approx}} [\![\mathsf{S}']\!] \mid [\![\mathsf{R}']\!]$.

5.4 A Second Compilation: Rules Are inside Protein Agents

This second compilation will be protein-centric. In the previous compilation, proteins were just passive solution states and even if this is formally correct, we want to provide a more "natural" compilation, more in line with Regev's representations of various biological pathways [14,16].

Let R be the set of basic rules of a given κ_b-system, $\mathfrak{r} \in \mathsf{R}$ be a rule, and $\ulcorner \mathfrak{r} \urcorner$ be the rule identifier (a number). A temporary network is set up between proteins by means of the agent $\textsc{link}_{\mathsf{R}} \stackrel{def}{=} \prod_{\mathfrak{r} \in \mathsf{R}} \mathsf{link}(\ulcorner \mathfrak{r} \urcorner)$, where $\mathsf{link}(\ulcorner \mathfrak{r} \urcorner)$ is defined below for activations and complexations. Suppose the reactant of $\mathfrak{r}$ are $A_1, \ldots, A_n$, we define:

$$\mathsf{link}(\ulcorner \mathfrak{r} \urcorner) = a_1\,(z_1).(z)(\mathsf{link}(\ulcorner \mathfrak{r} \urcorner, 2, z, z_1, \overline{a_1}\,\langle z_1 \rangle) \mid \overline{z}\,\langle\rangle)$$

$$(2 \leq i < n) \quad \mathsf{link}(\ulcorner \mathfrak{r} \urcorner, i, z, \widetilde{y}, P) = a_i\,(z_i).\mathsf{link}(\ulcorner \mathfrak{r} \urcorner, i+1, z, \widetilde{y} z_i, \overline{a_1}\,\langle z_1 \rangle) \\ + z\,().(P \mid \mathsf{link}(\ulcorner \mathfrak{r} \urcorner))$$

while for $i = n$, there is a slight difference depending on whether $\mathfrak{r}$ is an activation or a complexation. Specifically for activations and complexations respectively, we define:

$$\mathsf{link}(\ulcorner \mathfrak{r} \urcorner, n, z, z_1 \cdots z_{n-1}, P) = \\ a_n\,(z_n).(z'_1, \cdots z'_n)\Big(\overline{z_1}\,\langle \ulcorner \mathfrak{r} \urcorner, z'_n, z'_2 \rangle \mid \cdots \mid \overline{z_n}\,\langle \ulcorner \mathfrak{r} \urcorner, z'_{n-1}, z'_1 \rangle \mid \overline{z'_n}\,\langle 1 \rangle \mid \mathsf{link}(\ulcorner \mathfrak{r} \urcorner)\Big) \\ + z\,().(P \mid \mathsf{link}(\ulcorner \mathfrak{r} \urcorner))$$

$$\mathsf{link}(\ulcorner \mathfrak{r} \urcorner, n, z, z_1 \cdots z_{n-1}, P) = \\ a_n\,(z_n).(z'_1, \cdots z'_n)\Big(\overline{z_1}\,\langle \ulcorner \mathfrak{r} \urcorner, z'_n, z'_2 \rangle \mid \cdots \mid \overline{z_n}\,\langle \ulcorner \mathfrak{r} \urcorner, z'_{n-1}, z'_1 \rangle \mid (\widetilde{\xi})\overline{z'_n}\,\langle 1, \widetilde{\xi} \rangle \mid \mathsf{link}(\ulcorner \mathfrak{r} \urcorner)\Big) \\ + z\,().(P \mid \mathsf{link}(\ulcorner \mathfrak{r} \urcorner))$$

In order to coordinate the reactants, link sets up a ring to support a *token ring protocol* and passes the token to the n^{th} reactant. In addition, for complexations, link also creates the potential new connections. We observe that rules are attempted competitively, which is vaguely resembling the real thing.

The behaviour of a protein A depends on the activation and complexation rules it participates in. Let R_A be the set of rules the protein A may participate in (*i.e.*, is a reactant of). Then:

$$\mathtt{A}(X, Y, Z, \mathsf{R}) \;=\; [Y \setminus \mathcal{N}_v = \varnothing]\,\mathbf{0}\,; \\ (z)(\overline{a}\,\langle z \rangle \mid z\,(v, z', z'').\textstyle\prod_{\mathfrak{r} \in \mathsf{R}_A} [v = \ulcorner \mathfrak{r} \urcorner] \mathfrak{R}_A(X, Y, Z, z', z'', \mathsf{R}); \mathbf{0})$$

(the same remark regarding the expression $Y \setminus \mathcal{N}_v$ applies here as well) and we define the agents $\mathfrak{R}_{A_1}(X, Y, Z, z', z'', \mathsf{R})$ for the activation rule $\mathfrak{a}$ and the complexation rule $\mathfrak{c}$ of Section 5.3. (The agents $\mathfrak{A}_{A_i}$ and $\mathfrak{C}_{A_i}$, for other i, are similar.)

$$
\begin{array}{l}
\mathfrak{A}_{A_1}(X, Y, Z, z', z'', \mathsf{R}) \ = \\
\quad [\rho_1 \subseteq X, \sigma_1 \subseteq Y] \\
\qquad z'\,(v).[v = 1] \\
\qquad\quad (\overline{z''}\,\langle v\rangle \mid z'\,(v').[v' = 1](\mathtt{A}_1((X \setminus \rho_1) + \rho_1', (Y \setminus \sigma_1) + \sigma_1', Z, \mathsf{R}) \mid \overline{z''}\,\langle 1\rangle); \\
\qquad\qquad\qquad\qquad (\mathtt{A}_1(X, Y, Z, \mathsf{R}) \mid \overline{z''}\,\langle 0\rangle))); \\
\qquad\quad (\overline{z''}\,\langle 0\rangle \mid \mathtt{A}_1(X, Y, Z, \mathsf{R})); \\
\qquad z'\,(v).(\overline{z''}\,\langle 0\rangle \mid \mathtt{A}_1(X, Y, Z, \mathsf{R}))
\end{array}
$$

$$
\begin{array}{l}
\mathfrak{C}_{A_1}(X, Y, Z, z', z'', \mathsf{R}) \ = \\
\quad [\rho_1 \subseteq X, \sigma_1 + \tau_1 \subseteq Y] \\
\qquad z'\,(v, \widetilde{\xi}).[v = 1] \\
\qquad\quad (\overline{z''}\,\langle v, \widetilde{\xi}\rangle \mid z'\,(v', \widetilde{\xi'}).[v' = 1](\mathtt{A}_1(X', Y', Z', \mathsf{R}) \mid \overline{z''}\,\langle 1, \widetilde{\xi'}\rangle); \\
\qquad\qquad\qquad\qquad (\mathtt{A}_1(X, Y, Z, \mathsf{R}) \mid \overline{z''}\,\langle 0, \widetilde{\xi'}\rangle))); \\
\qquad\quad (\overline{z''}\,\langle 0, \widetilde{\xi}\rangle \mid \mathtt{A}_1(X, Y, Z, \mathsf{R})); \\
\qquad z'\,(v, \widetilde{\xi}).(\overline{z''}\,\langle 0, \widetilde{\xi}\rangle \mid \mathtt{A}_1(X, Y, Z, \mathsf{R})) \\
\quad \textit{where } X' = (X \setminus \rho_1) + \rho_1' \ , \ Y' = Y \setminus (\sigma_1 + \tau_1)) + \sigma_1' \ , \ Z' = Z + \widetilde{\xi_1}
\end{array}
$$

where $\widetilde{\xi_1}$ is the subsequence of $\widetilde{\xi'}$ containing names representing bound sites of the agent A_1. The token ring protocol used by reacting proteins consists in two steps. In the first, each protein sends 1 or 0 depending on whether its own sites satisfy the constraints of the rule or not. In the second step, 1 or 0 is propagated, depending on whether every protein satisfies the constraints of the rule or not. In case a protein receives 1 in the second step, the changes prescribed by the rule are performed, and the 1 is propagated. Otherwise the reduction is aborted, the initial state is restored, and the 0 is propagated.

The correctness of the compilation is established by means of an encoding $[\![\,\cdot\,]\!]_{\cdot}$ which depends on solutions and basic rules:

$$
\begin{array}{rl}
[\![\mathsf{S}, \mathsf{S}']\!]_{\mathsf{R}} & = [\![\mathsf{S}]\!]_{\mathsf{R}} \mid [\![\mathsf{S}']\!]_{\mathsf{R}} \\
[\![A(\rho, \sigma, \xi)]\!]_{\mathsf{R}} & = \mathtt{A}(\rho, \sigma, \xi, \mathsf{R}) \\
[\![A_1(\rho_1, \sigma_1, \xi_1) \odot \cdots \odot A_n(\rho_n, \sigma_n, \xi_n)]\!]_{\mathsf{R}} & = \\
\multicolumn{2}{r}{(\xi_1, \cdots, \xi_n)(\mathtt{A}_1(\rho_1, \sigma_1, \xi_1, \mathsf{R}) \mid \cdots \mid \mathtt{A}_n(\rho_n, \sigma_n, \xi_n, \mathsf{R}))}
\end{array}
$$

Lemma 2. *Let* (S, R) *be a* κ_b*-system and* $\textsc{link}_{\mathsf{R}}$ *be as above, then*

1. *if* $\mathsf{S} \longrightarrow_{\mathsf{R}} \mathsf{S}'$ *then* $[\![\mathsf{S}]\!]_{\mathsf{R}} \mid \textsc{link}_{\mathsf{R}} \mid \mathfrak{G} \longrightarrow^* [\![\mathsf{S}']\!]_{\mathsf{R}} \mid \textsc{link}_{\mathsf{R}} \mid \mathfrak{G}'$*; where* $\mathfrak{G}$ *and* $\mathfrak{G}'$ *are idle garbage agents;*
2. *if* $[\![\mathsf{S}]\!]_{\mathsf{R}} \mid \textsc{link}_{\mathsf{R}} \mid \mathfrak{G} \longrightarrow^* Q$*, where* $\mathfrak{G}$ *is an idle garbage agent, then* $Q \longrightarrow^* [\![\mathsf{S}']\!]_{\mathsf{R}} \mid \textsc{link}_{\mathsf{R}} \mid \mathfrak{G}'$ *with* $\mathfrak{G}'$ *an idle garbage agent, and* $\mathsf{S} \longrightarrow^*_{\mathsf{R}} \mathsf{S}'$.

Our idle garbage agents are parallel compositions of processes like $(z)\overline{z}\,\langle u, \widetilde{\xi}\rangle$. These agents collect the last messages of our token ring protocol. Of course we could refine the protocol such that no garbage is ever produced.

As before, we observe that $\mathsf{S} \downarrow_s a$ iff $[\![\mathsf{S}]\!]_{\mathsf{R}} \mid \textsc{link}_{\mathsf{R}} \downarrow a$, and from there one can prove the following:

Theorem 3. *Let* $\langle \mathsf{S}, \mathsf{R} \rangle$ *and* $\langle \mathsf{S}', \mathsf{R}' \rangle$ *be* κ_b*-systems, then* $\langle \mathsf{S}, \mathsf{R} \rangle \dot{\approx}_s \langle \mathsf{S}', \mathsf{R}' \rangle$ *iff* $[\![\mathsf{S}]\!]_{\mathsf{R}} \mid \text{LINK}_{\mathsf{R}} \dot{\approx} [\![\mathsf{S}']\!]_{\mathsf{R}'} \mid \text{LINK}_{\mathsf{R}'}$.

We conclude with two remarks. What is relevant in the former protocol—the token ring—is that it does not play a crucial role. Namely, our compilation may be abstracted from the protocol implementing the cooperation, still retaining the correctness. Our choice of the token ring protocol follows by its simplicity. But other reasons could be considered, such as efficiency, *i.e.* the number of channels used. In particular, our protocol doesn't use channels ξ_i which describe the underlying graph of connections of a complex. A clever protocol could make a profit of them, without introducing channels z_i. We keep this issue for future investigations. The second remark: barbed bisimilarity of our two compilations in π-calculus is an immediate consequence of Theorems 2 and 3.

6 Conclusions

We've presented a simple language of complexations and activations targetting core molecular biology. Synthesis and degradation were not considered, neither were decomplexations. No locations were introduced in the language either. We could have introduced all these. Though they certainly play a part in molecular biology, the idea here was to have the simplest meaningful language equipped with a barebone operational semantics and see if anything interesting happens.

What happens is that this language can be compiled in a even simpler fragment up to two different bisimulations. A simpler fragment which in turn can be compiled in two different styles in π-calculus. So the simplicity of the formalism pays off with this structured translation in π. In some sense the conclusion for now is that the formalism itself can be seen as a process algebra of its own. Further investigations will show if anything more practical comes out of it.

References

1. Roberto Amadio, Ilaria Castellani, and Davide Sangiorgi. On bisimulations for the asynchronous π-calculus. *Theoretical Computer Science*, 195(2):291–324, 1998.
2. Marc Chiaverini and Vincent Danos. A formalism for Kohn's molecular map. In *Proceedings of International Workshop on Computational Methods in Systems Biology*, LNCS. Springer-Verlag, 2003.
3. E. M. Clarke, E. Allen Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.
4. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.
5. Vincent Danos and Cosimo Laneve. Graphs for formal molecular biology. In *Proceedings of International Workshop on Computational Methods in Systems Biology*, LNCS. Springer-Verlag, 2003.
6. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, José Meseguer, and Kemal Sonmez. Pathway logic: Symbolic analysis of biological signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, January 2002. To appear.

7. Cédric Fournet and Georges Gonthier. The reflexive chemical abstract machine and the join-calculus. In *23rd ACM Symposium on Principles of Programming Languages (POPL'96)*, 1996.
8. Ronojoy Ghosh and Claire J. Tomlin. Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model? In *Proceedings of HSCC 2001*, volume 2034 of *LNCS*, pages 232–246. Springer, 2001.
9. Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM (JACM)*, 32(1):137–161, 1985.
10. Kurt W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*, n. 10:2703–2734, 1999.
11. Ron Maimon and Sam Browning. Diagrammatic notation and computational structure of gene networks. In *Proceedings of the 2nd International Conference on Systems Biology*, 2001.
12. Robin Milner and Davide Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *Nineteenth Colloquium on Automata, Languages and Programming (ICALP) (Wien, Austria)*, volume 623 of *LNCS*, pages 685–695. Springer, 1992.
13. Masao Nagasaki, Shuichi Onami, Satoru Miyano, and Hiroaki Kitano. Bio-calculus: Its concept and molecular interaction. *Genome Informatics*, 10:133–143, 1999.
14. 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.
15. A. Regev and E. Shapiro. Cells as computation. *Nature*, 419, September 2002.
16. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 6, pages 459–470, Singapore, 2001. World Scientific Press.
17. Birgit Schoeberl, Claudia Eichler-Jonsson, Ernst-Dieter Gilles, and Gertraud Müller. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, 20:370–375, 2002.
18. D. Thieffry and R. Thomas. Qualitative analysis of gene networks. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, volume 3, pages 77–88, Singapore, 1998. World Scientific Press.