A simple calculus for proteins and cells

Cosimo Laneve [a], Fabien Tarissan [b],

[a] *Dipartimento di Scienze dell'Informazione, Università di Bologna*

[b] *Équipe PPS, CNRS & Université Paris VII*

Abstract

The use of process calculi to represent biological systems has led to the design of different calculi such as brane calculi [1] and κ-calculus [3]. Both have proved to be useful to model different types of biological systems.

As an attempt to unify the two directions, we introduce the `bio`κ*-calculus*, a simple calculus for describing proteins and cells, in which bonds are represented by means of shared names and interactions are modelled at the domain level. Protein-protein interactions have to be at most binary and cell interactions have to fit with sort constraints.

We define the semantics of `bio`κ-calculus, analyse its properties, and discuss its expressiveness by modelling two significant examples: a signalling pathway and a virus infection.

1 Introduction

One problem when dealing with molecular biology is to extract a functional meaning out of the mass of current knowledge. This problem has pleaded for the development of specific tools that describe biology in a faithful way. Among these tools, process algebras have been proved powerful enough to formalise the interactions, to render in a natural way the massive parallelism and concurrency of interactions, and to analyse the overall behaviour.

Two different process algebraic developments in particular brought some interest and various results. A first approach based on the π-calculus [7] and following principles proposed in [8] uses shared names to represent bonds between proteins. One of those calculi – the κ-calculus [3] – showed to be very convenient for representing mechanisms such as signalling pathways or regulatory networks. Another family of calculi – the brane calculi – proposed in [1] by relying on Mobile Ambient [2], used action and co-action capabilities located on the surface of cell membranes. Such calculi demonstrated to be suitable for representing molecular transport as well as virus infections.

As these calculi turn out to commit to very different paradigms, it seems compelling to develop a unified framework able to handle the two kinds of

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: `www.elsevier.nl/locate/entcs`

systems. Such a unified framework is the aim of this contribution. In particular, the challenge is to enucleate few basic realistic primitives that permit to describe systems using mechanisms of κ-calculus and brane calculi.

Our new calculus, called bioκ-calculus, uses interactions that are complexations and decomplexations of two proteins. These interactions follow the same pattern of those of κ-calculus. Actually, they are similar, but even simpler, to mκ-calculus interactions [3], a calculus introduced to ease the translation of κ-calculus in π-calculus. The bioκ-calculus also retains compartments denoting cells. We precisely describe protein interactions when one of them belongs to the cell membrane and the other one to the nucleus or to the external solution. In such cases, a side-effect may occur: the interaction may change the capability of the membrane, thus preparing the cell to future fusions, endo- or exocytosis.

Cell fusions open the cell to other cells. In particular, the nucleus, whose interactions with the external solution are mediated by the membrane, may directly interact with another solution – the nucleus of the fusing cell – after the fusion. To model fusions we use a mechanism similar to one defined in the *higher-order π-calculus* [9].

Other relevant cellular interactions are also considered, such as translocations, which transport proteic material inside the cells, and phagocytosis, which allows a cell, such as viruses, to enter other cells. In the process of phagocytosis, the entering cell is enclosed into a membrane that is pull out the host cell membrane. This extraction is particularly difficult to formalise because it amounts to check that the membrane of the host cell has enough material for the new one. The bioκ-calculus formalisation overcome this difficulty by admitting matches of patterns of proteins.

Biological solutions are modelled in bioκ-calculus as labelled transition systems where labels carry information about the reactants and the rule used. It is folklore in process calculi that such transition systems are too *intentional objects* and equivalences are proposed to quotient them. In this paper we consider weak bisimulation [6] that equates two systems if they simulate each other. We demonstrate that weak bisimulation is a congruence in bioκ-calculus: two weakly bisimilar biological systems behave in the same way when put in *every* solution.

Notwithstanding the simplicity of the bioκ-calculus interactions (binary interactions between proteins or between membranes), the calculus is expressive enough. We discuss in full detail two significant biological examples: the RTK-MAPK pathway and a virus infection. However, the purpose of bioκ-calculus is to be a core framework for molecular biology, to be extended suitably for modelling complex systems.

The next section defines the bioκ-calculus where the interaction mechanisms are restricted to be between two proteins. Section 3 extends the basic interaction mechanisms with fusions that make structural modifications in the hierarchical organisation of a system. Section 4 discusses further extensions of

the calculus accounting for translocations and phagocytosis. Section 5 draws few concluding remarks and hints at possible future works. Proofs are omitted for fitting with space limits.

2 The core of the bioκ-calculus

In this section we present a core version of the bioκ-calculus where interactions never change the hierarchical organisation of biological solutions. We define its syntax, its operational semantics, and weak bisimulation. We also analyse its expressiveness by encoding the RTK-MAPK pathway.

Notational preliminaries. Four disjoint countable sets of names will be used: a set of *protein names*, ranged over by A, B, C, $\cdots$; a set of *edge names* E, ranged over by x, y, z, $\cdots$; a set of *membrane names*, ranged over by M, N, $\cdots$. Protein names are sorted according to the number of *sites* they possess. Let $\mathfrak{s}(\cdot)$ be the function yielding the sites of proteins. The sites of a protein are indicated by the natural numbers in the set $\{1, \cdots, \mathfrak{s}(\mathrm{A})\}$.

Sites may be *bound* to other sites, *visible*, i.e. not connected to other sites, or *hidden*, i.e. not connected to other sites but not useful for interactions. The state of sites are defined by maps:

- *interfaces*, ranged over by σ, σ', $\cdots$, are partial functions from naturals to the set $\mathsf{E} \cup \{v, h\}$ (we are assuming that $v, h \notin \mathsf{E}$). For instance, $[1 \mapsto x; 2 \mapsto v; 3 \mapsto h]$ is an interface. In order to simplify the reading, we write this map as $1^x + 2 + \overline{3}$. In the following, when we write $\sigma + \sigma'$ we assume that the domains of σ and σ' are disjoint.

  Interfaces are injective on edges E. Since edges have two endpoints, we are excluding that such endpoints belong to the same protein (*cf.* *self complexation* in [3]).

  The sites of a protein A are completely defined by *total* interfaces on $[1 \,..\, \mathfrak{s}(\mathrm{A})]$
- v-h-maps, ranged over by $\phi, \psi, \cdots$, are interfaces onto $\{v, h\}$. We write $\overline{\phi}$ for the following v-h-map:

$$\overline{\phi}(i) \;=\; \begin{cases} h & \text{if } \phi(i) = v \\ v & \text{if } \phi(i) = h \end{cases}$$

The syntax. The syntax of the bioκ-calculus defines (biological) *solutions* S:

$$\mathsf{S} \;::=\; \underset{\text{(empty)}}{\mathbf{0}} \;\mid\; \underset{\text{(protein)}}{\mathrm{A}(\sigma)} \;\mid\; \underset{\text{(cell)}}{\mathrm{M}(\!|\,\mathsf{S}\,|\!)[\mathsf{S}]} \;\mid\; \underset{\text{(group)}}{\mathsf{S}\,,\,\mathsf{S}}$$

Solutions can be either empty, or a *protein* $\mathrm{A}(\sigma)$ indicating a protein name

and its interface, or a *cell* $\mathrm{M}(\!|\, \mathsf{S} \,|\!)[\mathsf{T}]$, that is a solution T, called *nucleus* [1], surrounded by another S, called *membrane*, or a *group* of solutions S, T. Two auxiliary functions will be applied to solutions and interfaces. The function $\mathtt{en}(\cdot)$ returns the set of *edge names* occurring in the argument; the function $\mathtt{de}(\cdot)$ returns the set of *dangling edge names* of the argument, namely those names that occur exactly once; the function $\mathtt{be}(\cdot)$ returns the set of *bound edge names* of the argument, namely those names that occur exactly twice. Clearly $\mathtt{de}(\mathsf{S}) = \mathtt{en}(\mathsf{S}) \setminus \mathtt{be}(\mathsf{S})$ and similarly for σ. For instance, in $\mathsf{S} = \mathrm{M}(\!|\, \mathrm{C}(1^y+2) \,|\!)[\mathrm{A}(1^x+2+3), \mathrm{B}(1+2^x)]$ the set $\mathtt{en}(\mathsf{S})$ is $\{y, x\}$ and the set $\mathtt{de}(\mathsf{S})$ is $\{y\}$. We abbreviate the group $\mathrm{A}_1(\sigma_1)\ \mathtt{,}\ \cdots\ \mathtt{,}\ \mathrm{A}_n(\sigma_n)$ with $\prod_{i\in 1..n} \mathrm{A}_i(\sigma_i)$.

In the whole paper, we identify solutions that are equal up to a renaming of edge names that are not dangling (called alpha-conversion) and we assume that all solutions meet the following *well-formedness conditions*:

- (*edge-condition)* in every solution, edge names occur at most twice;
- (*membrane-condition)* every membrane is a group of proteins, that is cells do not occur in membranes;
- (*nucleus-condition)* the dangling edges of nuclei of cells are connected to the corresponding membrane, that is, for every $\mathrm{M}(\!|\, \mathsf{S} \,|\!)[\mathsf{T}]$, $\mathtt{de}(\mathsf{T}) \subseteq \mathtt{de}(\mathsf{S})$.

For example $\mathrm{M}(\!|\, \mathrm{C}(1^x+2) \,|\!)[\mathrm{A}(1^x+2+3), \mathrm{B}(1+2^x)]$ does not meet the edge condition because the edge x has three ends (it is a multi-edge). The solution $\mathrm{M}(\!|\, \mathrm{B}(1+2)\ \mathtt{,}\ \mathrm{C}(1+2) \,|\!)[\mathrm{A}(1+2^x+3)]$ does not meet the nucleus condition because the nucleus $\mathrm{A}(1+2^x+3)$ has a dangling edge that is not connected to the membrane.

In the following, solutions that are membranes will be addressed by $\mathsf{M}, \mathsf{N}, \cdots$.

Biological reactions. Biological reactions that we consider in this section are of two types: *complexations*, which create edges between possibly disconnected proteins, and *decomplexations*, which remove edges. For instance a complexation reaction is (we are assuming $\mathfrak{s}(\mathrm{A}) = \mathfrak{s}(\mathrm{B}) = 3$)

$$\mathrm{A}(1^x+2+\overline{3})\ \mathtt{,}\ \mathrm{B}(1+\overline{2}+\overline{3}) \longrightarrow \mathrm{A}(1^x+2^y+3)\ \mathtt{,}\ \mathrm{B}(1^y+2+\overline{3})$$

that creates an edge y connecting the site 2 of A and the site 1 of B. These two sites, in order that this reaction be executed, must be visible. This means that the application of a complexation must check whether the sites being connected are visible or not. For example the above reaction cannot be applied to the group $\mathrm{A}(1^x+\overline{2}+\overline{3})\ \mathtt{,}\ \mathrm{B}(1+\overline{2}+\overline{3})$ because the site 2 of A is hidden. Reactions in $\mathtt{bio}\kappa$-calculus may also change the state of sites that are visible or hidden in the reactants, switching them into hidden and visible, respectively. In the example above, this happens to the sites 3 of A and 2 of B. A concise way for defining the above reaction is the schema

$$\mathtt{r}: \quad ((\mathrm{A}, 2, \overline{3}), (\mathrm{B}, 1, 2))$$

[1] We refer to every material surrounded by a membrane with the generic term "nucleus"; this is also referred as "cytoplasm".

that makes explicit the reactant proteins – the first items of the triples –, the corresponding sites to be complexated – the second items – and the part of the interface whose state must be switched. For example, the rule $\mathtt{r}$ also applies to $\text{A}(1+2+\overline{3})\ ,\ \text{B}(1+\overline{2}+3)$ or $\text{A}(\overline{1}+2+\overline{3})\ ,\ \text{B}(1+\overline{2}+3^x)$ yielding solutions $\text{A}(1+2^y+3)\ ,\ \text{B}(1^y+\overline{2}+3)$ and $\text{A}(\overline{1}+2^y+3)\ ,\ \text{B}(1^y+\overline{2}+3^x)$, respectively. In general, the shape of a reaction schema is

$$\mathtt{r}: \quad ((\text{A}, i, \psi), (\text{B}, j, \phi))$$

that is a reaction name $\mathtt{r}$ and two triples containing a protein name, a site and a v-h-map. A generic *application* of the schema $\mathtt{r}$ may be written as

$$\text{A}(i+\psi+\sigma)\ ,\ \text{B}(j+\phi+\sigma') \longrightarrow \text{A}(i^x+\overline{\psi}+\sigma)\ ,\ \text{B}(j^x+\overline{\phi}+\sigma')$$

where x is a fresh edge name and $i+\psi+\sigma$ and $j+\phi+\sigma'$ are total on $[1\,..\,\mathfrak{s}(\text{A})]$ and $[1\,..\,\mathfrak{s}(\text{B})]$, respectively. It is worth to observe that the interfaces σ and σ' are not changed by $\mathtt{r}$, for this reason they are not mentioned in the schema.

Decomplexations are complexations in the other way round. For instance a decomplexation reaction is

$$\text{A}(1^x+2^y+3)\ ,\ \text{B}(1^y+2+\overline{3}) \longrightarrow \text{A}(1^x+2+\overline{3})\ ,\ \text{B}(1+\overline{2}+\overline{3})$$

that removes the edge y. The schema describing decomplexations is similar to that of complexations:

$$\mathtt{r}': \quad ((\text{A}, i, \psi'), (\text{B}, j, \phi'))$$

The application of the decomplexation rule is different from complexation: in this case the two reactants must be connected by an edge between the site i of A and the site j of B. So, for example, a generic application of $\mathtt{r}'$ is

$$\text{A}(i^x+\psi'+\sigma)\ ,\ \text{B}(j^x+\phi'+\sigma') \longrightarrow \text{A}(i+\overline{\psi'}+\sigma)\ ,\ \text{B}(j+\overline{\phi'}+\sigma')$$

In order to separate complexations from decomplexations we consider two functions, $\mathcal{C}$ for *complexations* and $\mathcal{D}$ for *decomplexations*, from rule names to tuples $((\text{A}, i, \phi), (\text{B}, j, \psi))$. These functions $\mathcal{C}$ and $\mathcal{D}$ are assumed with disjoint domains, therefore a rule name uniquely defines whether it is a complexation or a decomplexation.

Let $\mathcal{R}$ range over $\mathcal{C}$ and $\mathcal{D}$; let also $(\text{A}, i, \phi) \in \mathcal{R}(\mathtt{r})$ if either $\mathcal{R}(\mathtt{r}) = ((\text{A}, i, \phi), (\text{B}, j, \psi))$ or $\mathcal{R}(\mathtt{r}) = ((\text{B}, j, \psi), (\text{A}, i, \phi))$. Finally let μ range over $\text{A}^x_{\mathtt{r}}$ or τ and let $\mathtt{diff}(\mathsf{S}, \mathsf{S}')$ be the set $(\mathtt{en}(\mathsf{S}')\backslash\mathtt{en}(\mathsf{S})) \cup (\mathtt{en}(\mathsf{S})\backslash\mathtt{en}(\mathsf{S}'))$.

Definition 2.1 The *transition relation* $\xrightarrow{\mu}$ is the least relation satisfying the

reductions:

$$\text{(com)}\ \frac{(\mathrm{A}, a, \phi) \in \mathcal{C}(\mathtt{r}) \quad x \notin \mathtt{en}(\sigma)}{\mathrm{A}(a + \phi + \sigma) \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathrm{A}(a^x + \overline{\phi} + \sigma)} \qquad \text{(dec)}\ \frac{(\mathrm{A}, a, \phi) \in \mathcal{D}(\mathtt{r})}{\mathrm{A}(a^x + \phi + \sigma) \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathrm{A}(a + \overline{\phi} + \sigma)}$$

$$\text{(sol)}\ \frac{\mathsf{S} \xrightarrow{\mu} \mathsf{S}' \quad \mathtt{diff}(\mathsf{S}, \mathsf{S}') \cap \mathtt{en}(\mathsf{T}) = \emptyset}{\mathsf{S}\ ,\ \mathsf{T} \xrightarrow{\mu} \mathsf{S}'\ ,\ \mathsf{T}} \qquad \text{(mem)}\ \frac{\mathsf{M} \xrightarrow{\mu} \mathsf{M}' \quad \mathtt{diff}(\mathsf{M}, \mathsf{M}') \cap \mathtt{en}(\mathsf{S}) = \emptyset}{\mathrm{M}\langle\!|\, \mathsf{M} \,|\!\rangle[\mathsf{S}] \xrightarrow{\mu} \mathrm{M}\langle\!|\, \mathsf{M}' \,|\!\rangle[\mathsf{S}]}$$

$$\text{(nucleus)}\ \frac{\mathsf{S} \xrightarrow{\tau} \mathsf{S}' \quad \mathtt{diff}(\mathsf{S}, \mathsf{S}') \cap \mathtt{en}(\mathsf{M}) = \emptyset}{\mathrm{M}\langle\!|\, \mathsf{M} \,|\!\rangle[\mathsf{S}] \xrightarrow{\tau} \mathrm{M}\langle\!|\, \mathsf{M} \,|\!\rangle[\mathsf{S}']}$$

$$\text{(react)}\ \frac{\mathsf{S} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{S}' \quad \mathsf{T} \xrightarrow{\mathrm{B}^x_{\mathtt{r}}} \mathsf{T}' \quad \mathrm{A} \neq \mathrm{B}}{\mathsf{S}\ ,\ \mathsf{T} \xrightarrow{\tau} \mathsf{S}'\ ,\ \mathsf{T}'} \qquad \text{(ms-react)}\ \frac{\mathsf{M} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{M}' \quad \mathsf{S} \xrightarrow{\mathrm{B}^x_{\mathtt{r}}} \mathsf{S}' \quad \mathrm{A} \neq \mathrm{B}}{\mathrm{M}\langle\!|\, \mathsf{M} \,|\!\rangle[\mathsf{S}] \xrightarrow{\tau} \mathrm{M}\langle\!|\, \mathsf{M}' \,|\!\rangle[\mathsf{S}']}$$

and the symmetric rule for *(*sol*)*.

Let $\mathsf{S} \stackrel{\tau}{\Longrightarrow} \mathsf{S}'$ if $\mathsf{S} \xrightarrow{\tau}{}^* \mathsf{S}'$ and $\mathsf{S} \stackrel{\mu}{\Longrightarrow} \mathsf{S}'$, with $\mu \neq \tau$, if $\mathsf{S} \xrightarrow{\tau}{}^* \xrightarrow{\mu} \xrightarrow{\tau}{}^* \mathsf{S}'$.

Rules (com) and (dec) respectively define complexations and decomplexations capabilities of proteins by lifting these information to labels of transitions and, at the same time, updating the proteins. Rules (sol) and (mem) lift transitions to groups and membranes; it is crucial that edge names created or deleted do not occur elsewhere. Rule (nucleus) lift *internal* transitions of nuclei to the whole cell; as before, edge names created or deleted must not occur elsewhere. We observe that (nucleus) bans complexation or decomplexation between nuclei and the solution external to the cells. Rule (react) and (ms-react) define reactions, both complexations and decomplexations, in groups and cells. In particular (react) also accounts for reactions between membranes of different cells. It is worth to notice that the constraint $\mathrm{A} \neq \mathrm{B}$ allows reactions between different proteins only. This is for simplicity sake: in case reactants are proteins with a same name we need to carry more information on the labels to separate them.

By Definition 2.1, the previous notation $\longrightarrow$ must be read as $\xrightarrow{\tau}$. In facts, in [3] the transitions of the $\mathtt{m}\kappa$-calculus were defined by means of an *unlabelled* reduction relation $\longrightarrow$ that corresponds to the foregoing rules (com), (dec), (sol), and (react). In this case we have adhered to a labelled transition for reducing the number of the rules: such number should have been larger than in Definition 2.1 because of the presence of membranes.

The transition relation preserve well-formedness of solutions.

Proposition 2.2 *If* S *is well-formed and* $\mathsf{S} \xrightarrow{\mu} \mathsf{T}$ *then* S *is well-formed as well.*

It is worth to observe that membrane names do not play any role at this stage. They will be relevant in the complete system with complex membrane reactions presented in Section 3.

Example 2.3 The so-called RTK-MAPK pathway are intensely used and studied in many approaches modelling and simulating biological systems [8,3]. We therefore model the first steps of such a pathway in bioκ-calculus, thus providing a touchstone for our calculus.

The signal stimulus stems from the epidermal growth factor EGF whose dimeric form (1) can bind to its associated receptor EGFR (2), a transmembrane protein with an extracellular ligand-binding domain located on the plasmic membrane of some cells. This binding activates EGFR by phosphorylating an internal domain of the protein (3 and 4). This activation leads to multiple interactions with cytoplasmic complexes of proteins by successive binding-phosphorylations, starting with the adapter protein SHC (5). The cascade of interactions ends with the activation of the extracellular signal-regulated kinase ERK. This phosphorylated protein can then translocate into the nucleus and modify the gene expression, stimulating cells to enter mitosis. This causes the cell to divide and proliferate.

After the biological description EGF, RTK, and SHC have respective arities 3, 4, and 2. We give here the formal rendering of the five first steps described above:

$$\begin{aligned}
\mathtt{r}_1 &: && ((\textsc{egf},1,\overline{2}),(\textsc{egf},1,\overline{2})) \in \mathcal{C}\\
\mathtt{r}_2 &: && ((\textsc{egf},2,\emptyset),(\textsc{egfr},1,\overline{4})) \in \mathcal{C}\\
\mathtt{r}_3 &: && ((\textsc{egfr},2,\overline{3}+4),(\textsc{egfr},2,\overline{3}+4)) \in \mathcal{C}\\
\mathtt{r}_4 &: && ((\textsc{egfr},2,\emptyset),(\textsc{egfr},2,\emptyset)) \in \mathcal{D}\\
\mathtt{r}_5 &: && ((\textsc{egfr},3,\emptyset),(\textsc{shc},1,\overline{2})) \in \mathcal{C}
\end{aligned}$$

The simple run below displays that our calculus is expressive enough to define the causality involved in the transduction in a precise yet natural way.

$$\begin{aligned}
&\textsc{egf}(1+\overline{2})\ ,\ \textsc{egf}(1+\overline{2})\ ,\\
&\qquad \textsc{m}\big(\!\big|\ \textsc{egfr}(1+2^{+}\overline{3}+\overline{4})\ ,\ \textsc{egfr}(1+2^{+}\overline{3}+\overline{4})\ ,\ \mathsf{M}\ \big|\!\big)[\textsc{shc}(1+\overline{2})\ ,\ \mathsf{S}]\\
&\xrightarrow{\tau}\ \textsc{egf}(1^{z}+2)\ ,\ \textsc{egf}(1^{z}+2)\ ,\\
&\qquad \textsc{m}\big(\!\big|\ \textsc{egfr}(1+2+\overline{3}+\overline{4})\ ,\ \textsc{egfr}(1+2+\overline{3}+\overline{4})\ ,\ \mathsf{M}\ \big|\!\big)[\textsc{shc}(1+\overline{2})\ ,\ \mathsf{S}] \quad (\mathtt{r}_1)\\
&\xrightarrow{\tau}\ \textsc{egf}(1^{z}+2^{y})\ ,\ \textsc{egf}(1^{z}+2)\ ,\\
&\qquad \textsc{m}\big(\!\big|\ \textsc{egfr}(1^{y}+2+\overline{3}+4)\ ,\ \textsc{egfr}(1+2+\overline{3}+\overline{4})\ ,\ \mathsf{M}\ \big|\!\big)[\textsc{shc}(1+\overline{2})\ ,\ \mathsf{S}] \quad (\mathtt{r}_2)\\
&\xrightarrow{\tau}\ \textsc{egf}(1^{z}+2^{y})\ ,\ \textsc{egf}(1^{z}+2^{u})\ ,\\
&\qquad \textsc{m}\big(\!\big|\ \textsc{egfr}(1^{y}+2+\overline{3}+4)\ ,\ \textsc{egfr}(1^{u}+2+\overline{3}+4)\ ,\ \mathsf{M}\ \big|\!\big)[\textsc{shc}(1+\overline{2})\ ,\ \mathsf{S}] \quad (\mathtt{r}_2)
\end{aligned}$$

$$\xrightarrow{\tau} \text{EGF}(1^z + 2^y)\ ,\ \text{EGF}(1^z + 2^u)\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1^y + 2^x + 3 + \overline{4})\ ,\ \text{EGFR}(1^u + 2^x + 3 + \overline{4})\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1 + \overline{2})\ ,\ \mathsf{S}] \quad (\mathtt{r}_3)$$
$$\xrightarrow{\tau} \text{EGF}(1^z + 2^y)\ ,\ \text{EGF}(1^z + 2^u)\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1^y + 2 + 3 + \overline{4})\ ,\ \text{EGFR}(1^u + 2 + 3 + \overline{4})\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1 + \overline{2})\ ,\ \mathsf{S}] \quad (\mathtt{r}_4)$$
$$\xrightarrow{\tau} \text{EGF}(1^z + 2^y)\ ,\ \text{EGF}(1^z + 2^u)\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1^y + 2 + 3^t + \overline{4})\ ,\ \text{EGFR}(1^u + 2 + 3 + \overline{4})\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1^t + 2)\ ,\ \mathsf{S}] \quad (\mathtt{r}_5)$$

A couple of problems of our notation deserve to be discussed though. Consider the initial solution:

$$\text{EGF}(1 + \overline{2})\ ,\ \text{EGF}(1 + \overline{2})\ ,\ \text{EGF}(1 + \overline{2})\ ,\ \text{EGF}(1 + \overline{2})\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1 + 2^x + \overline{3} + \overline{4})\ ,\ \text{EGFR}(1 + 2^x + \overline{3} + \overline{4})\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1 + \overline{2})\ ,\ \mathsf{S}]$$

After two applications of rule $\mathtt{r}_1$, we obtain the solution

$$\text{EGF}(1^x + 2)\ ,\ \text{EGF}(1^x + 2)\ ,\ \text{EGF}(1^y + 2)\ ,\ \text{EGF}(1^y + 2)\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1 + 2^x + \overline{3} + \overline{4})\ ,\ \text{EGFR}(1 + 2^x + \overline{3} + \overline{4})\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1 + \overline{2})\ ,\ \mathsf{S}]$$

that reduces, after two application of rule $\mathtt{r}_2$ to the wrong solution

$$\text{EGF}(1^x + 2)\ ,\ \text{EGF}(1^x + 2^u)\ ,\ \text{EGF}(1^y + 2^v)\ ,\ \text{EGF}(1^y + 2)\ ,$$
$$\text{M}(\!|\ \text{EGFR}(1^u + 2^x + \overline{3} + 4)\ ,\ \text{EGFR}(1^v + 2^x + \overline{3} + 4)\ ,\ \mathsf{M}\ |\!)[\text{SHC}(1 + \overline{2})\ ,\ \mathsf{S}]$$

where two different dimeric forms of EGF connect to a same pair of EGFR receptors. Our notation is too simple to rule out such configurations. In $\mathtt{m}\kappa$-calculus, this expressiveness issue is solved by the use of reaction IDs and pattern matching over them. Actually, this issue is related to a more general question named self-assembly problem and worth to be studied independently [5].

The second problem is manifested at the end of the RTK-MAPK pathway. The pathway causes a phosphorylation of a particular protein (ERK) that enters in the nucleus, which is represented as cell, as well. At this stage we have no mechanism that make entities enter in the cell. Such mechanisms will be discussed in detail in Section 4.

Extensional semantics: bisimulation. The transition relation of Definition 2.1 associates solutions to graphs where nodes are solutions and μ-labelled edges model the transitions $\mathsf{S} \xrightarrow{\mu} \mathsf{S}'$. The induced equivalence on $\mathtt{bio}\kappa$-calculus solutions is graph isomorphism: two terms are equivalent provided their associated graphs are isomorphic. Graph isomorphism is too strong as a biological semantics because it distinguishes solutions that are equal up-to τ-transitions:

- Let $\mathcal{C}(\mathtt{r}) = ((\text{A}, 1, \emptyset), (\text{B}, 1, \emptyset)) = \mathcal{D}(\mathtt{r}')$, that is $\mathtt{r}$ and $\mathtt{r}'$ are *reversible* reactions. Then the solutions $\text{A}(1^y + \sigma)\ ,\ \text{B}(1^y + \sigma')$ and $\text{A}(1 + \sigma)\ ,\ \text{B}(1 + \sigma')$ have underlying graphs that are not isomorphic. Also in this case there is no

reason to separate the two solutions: they have isomorphic graphs up-to a τ-transition, which is an internal reduction of the solutions and should not be observable.

The following equivalence, adapting weak bisimulation in process calculi [7] to our setting, does not undergo the above criticism.

Definition 2.4 A *(weak) bisimulation* is a symmetric binary relation $\mathfrak{R}$ between solutions such that $\mathsf{S}\ \mathfrak{R}\ \mathsf{T}$ implies:

(i) if $\mathsf{S} \xrightarrow{\tau} \mathsf{S}'$ then $\mathsf{T} \stackrel{\tau}{\Longrightarrow} \mathsf{T}'$ and $\mathsf{S}'\ \mathfrak{R}\ \mathsf{T}'$;

(ii) if $\mathsf{S} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{S}'$ then $\mathsf{T} \stackrel{\mathrm{A}^x_{\mathtt{r}}}{\Longrightarrow} \mathsf{T}'$ and $\mathsf{S}'\ \mathfrak{R}\ \mathsf{T}'$.

S is bisimilar to T, written $\mathsf{S} \approx \mathsf{T}$, if $\mathsf{S}\ \mathfrak{R}\ \mathsf{T}$ for some bisimulation $\mathfrak{R}$.

Proposition 2.5 (i) *" , " is an abelian monoidal operator with identity* $\mathbf{0}$. *Namely* $\mathsf{S}\ ,\ \mathsf{T} \approx \mathsf{T}\ ,\ \mathsf{S}$ *and* $(\mathsf{S}\ ,\ \mathsf{T})\ ,\ \mathsf{R} \approx \mathsf{S}\ ,\ (\mathsf{T}\ ,\ \mathsf{R})$ *and* $\mathsf{S}\ ,\ \mathbf{0} \approx \mathsf{S}$.

(ii) $\approx$ *is preserved by injective renamings that are identities on dangling edge names. Namely, let* ι *be an injective renaming on* $\mathtt{en}(\mathsf{S})$ *such that* ι *is the identity on* $\mathtt{de}(\mathsf{S})$, *then* $\mathsf{S} \approx \iota(\mathsf{S})$.

(iii) $\approx$ *is preserved by reversible rules. Namely, let* $\mathcal{C}(\mathtt{r}) = ((\mathrm{A}, i, \psi), (\mathrm{B}, j, \phi))$ *and* $\mathcal{D}(\mathtt{r}') = ((\mathrm{A}, i, \overline{\psi}), (\mathrm{B}, j, \overline{\phi}))$ *then* $\mathrm{A}(i + \psi + \sigma)\ ,\ \mathrm{B}(j + \phi + \sigma') \approx \mathrm{A}(i^x + \overline{\psi} + \sigma)\ ,\ \mathrm{B}(1^x + \overline{\phi} + \sigma')$.

A relevant property of $\approx$ is that every two bisimilar systems behave in the same way when plugged in a same context.

Theorem 2.6 $\approx$ *is a congruence.*

As far as biology is concerned, the substitution property owned by $\approx$ might be too strong, thus making this equivalence an unsensible semantics. In this context, one usually wants to prove that two parts behave in the same way when plugged *under a certain number of contexts*, rather than every possible one. Therefore, a semantics owning a parametric congruence property might fit better with biology. However what these parameters are and what are the properties owned by a "good" extensional semantics for biology remains unclear to us and is left as an open issue.

Other remarks about $\approx$ are in order.

(i) $\mathsf{S} \approx \mathsf{T}$ does not imply $\mathtt{de}(\mathsf{S}) = \mathtt{de}(\mathsf{T})$. For two reasons: First of all, by bisimulation, $\mathsf{S} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{S}'$ may be matched by $\mathsf{T} \xrightarrow{\mathrm{A}^y_{\mathtt{r}}} \mathsf{T}'$ with $x \neq y$. Second, taking empty biological relations – i.e. $\mathcal{C} = \emptyset$ and $\mathcal{D} = \emptyset$ –, then $\mathrm{A}(1^x) \approx \mathbf{0}$ but their dangling edges are different.

(ii) Nevertheless a relationship on a subset of the dangling edges of two bisimilar solutions may still be established. Let $\mathtt{oe}(\mathsf{S})$, called the *observable edges*, be the set $\{x \mid \mathsf{S} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{S}'$ and $\mathtt{r}$ in domain of $\mathcal{D}\}$. It is easy to prove that if $\mathsf{S} \approx \mathsf{T}$ then there is an injective renaming ι such that $\mathtt{oe}(\mathsf{S}) = \iota(\mathtt{oe}(\mathsf{T}))$.

(iii) Let M be *inert* when $\mathsf{M} \stackrel{\mathrm{A}^x_{\mathtt{r}}}{\not\Longrightarrow}$ for every $\mathrm{A}^x_{\mathtt{r}}$. It is possible to verify that, if M is inert, then $\mathrm{M}(\!|\, \mathsf{M} \,|\!)[\mathsf{S}] \approx \mathbf{0}$.

3 Cell interactions

The calculus of Section 2 is not very different from $\mathtt{m}\kappa$-calculus. Cells, in particulars, do not play any relevant role since their structure is preserved by the transition. In this section we explore an extension with brane primitives, thus being able to model merging and splitting of cells such as the following *endosomes fusion*:

$$\mathrm{ESM}(\!|\, \mathsf{M} \,|\!)[\mathsf{S}] \;,\; \mathrm{ESM}(\!|\, \mathsf{N} \,|\!)[\mathsf{T}] \;\longrightarrow\; \mathrm{ESM}(\!|\, \mathsf{M} \,,\, \mathsf{N} \,|\!)[\mathsf{S} \,,\, \mathsf{T}]$$

The extension of core $\mathtt{bio}\kappa$-calculus we are going to design retains higher-order mechanisms, following *higher-order π-calculus*, a similar extension already studied for π-calculus [9]. We begin by augmenting the syntax with membrane reagents:

$$\mathsf{S} \;::=\; \cdots \;\mid\; \wr \mathsf{M} \,;\, \mathsf{S} \int \cdot \mathsf{S} \qquad \text{(mreagent)}$$

An mreagent is an intermediate (unstable) solution that is used for manifesting the capability to perform a fusion with a cell. Mreagents $\wr \mathsf{M} \,;\, \mathsf{S} \int \cdot \mathsf{T}$ meet the following properties: S and T do not contain other mreagents, M is a multiset of proteins and $\mathtt{de}(\mathsf{S}) \subseteq \mathtt{de}(\mathsf{M})$.

There are two operations involving membranes:

(i) *fusions* – two membranes are close and they are fused becoming a unique membrane. Fusions are formalised by a function $\mathcal{F}$ from rule names to triples $((\mathrm{M}, \mathrm{M}'), \mathrm{N})$. We write $(\mathrm{M} \otimes \mathrm{M}', \mathrm{N}) = \mathcal{F}(\mathtt{r})$ if either $\mathcal{F}(\mathtt{r}) = ((\mathrm{M}, \mathrm{M}'), \mathrm{N})$ or $\mathcal{F}(\mathtt{r}) = ((\mathrm{M}', \mathrm{M}), \mathrm{N})$. We also write $\mathrm{M} \in \mathcal{F}(\mathtt{r})$ if $(\mathrm{M} \otimes \mathrm{M}', \mathrm{N}) = \mathcal{F}(\mathtt{r})$, for some M' and N. We assume that the domains of $\mathcal{F}$, $\mathcal{C}$ and $\mathcal{D}$ are disjoint.

(ii) *side effects of complexations and decomplexations* – a complexation of a protein on the membrane of the cell might activate the membrane and prepare it for possible fusions. Activations of complexations are defined by a function $\mathcal{A}$ that takes pairs $(\mathrm{A}_{\mathtt{r}}, \mathrm{M})$ and returns membrane names.

In the following, with an abuse of notation, we use μ to also range over labels $\mathrm{M}_{\mathtt{r}}$.

Definition 3.1 The *transition relation* $\stackrel{\mu}{\longrightarrow}$ is the least one that includes the rules in Definition 2.1 where (MEM) and (MS-REACT) have also the premise

"$(\mathrm{A}_{\mathtt{r}}, \mathrm{M})$ not in the domain of $\mathcal{A}$" and the following ones

$$\text{(OPEN)}\quad \frac{\mathrm{M} \in \mathcal{F}(\mathtt{r})}{{}_{\mathrm{M}}\langle\!\langle\, \mathsf{M} \,\rangle\!\rangle[\mathsf{S}] \xrightarrow{\mathrm{M}_{\mathtt{r}}} \wr \mathsf{M} \,;\, \mathsf{S} \int \cdot \mathbf{0}} \qquad \text{(GRASP)}\quad \frac{\mathsf{S} \xrightarrow{\mu} \wr \mathsf{M} \,;\, \mathsf{S}'' \int \cdot \mathsf{S}'}{\mathsf{S} \,,\, \mathsf{T} \xrightarrow{\mu} \wr \mathsf{M} \,;\, \mathsf{S}'' \int \cdot (\mathsf{S}' \,,\, \mathsf{T})}$$

$$\text{(FUSE)}\quad \frac{\mathsf{S} \xrightarrow{\mathrm{M}_{\mathtt{r}}} \wr \mathsf{M} \,;\, \mathsf{S}'' \int \cdot \mathsf{S}' \quad \mathsf{T} \xrightarrow{\mathrm{N}_{\mathtt{r}}} \wr \mathsf{N} \,;\, \mathsf{T}'' \int \cdot \mathsf{T}' \quad \mathcal{F}(\mathtt{r}) = (\mathrm{M} \otimes \mathrm{N}, \mathrm{M}')}{\mathsf{S} \,,\, \mathsf{T} \xrightarrow{\tau} \mathsf{S}' \,,\, \mathsf{T}' \,,\, {}_{\mathrm{M}'}\langle\!\langle\, \mathsf{M} \,,\, \mathsf{N} \,\rangle\!\rangle[\mathsf{S}'' \,,\, \mathsf{T}'']}$$

$$\text{(FUSE-I)}\quad \frac{\mathsf{S} \xrightarrow{\mathrm{N}_{\mathtt{r}}} \wr \mathsf{N} \,;\, \mathsf{T} \int \cdot \mathsf{S}' \quad \mathcal{F}(\mathtt{r}) = (\mathrm{M} \otimes \mathrm{N}, \mathrm{M}')}{{}_{\mathrm{M}}\langle\!\langle\, \mathsf{M} \,\rangle\!\rangle[\mathsf{S}] \xrightarrow{\tau} {}_{\mathrm{M}'}\langle\!\langle\, \mathsf{M} \,,\, \mathsf{N} \,\rangle\!\rangle[\mathsf{S}'] \,,\, \mathsf{T}}$$

$$\text{(MEM-A)}\quad \frac{\mathsf{M} \xrightarrow{\mathrm{A}^{x}_{\mathtt{r}}} \mathsf{M}' \quad \mathcal{A}(\mathrm{A}_{\mathtt{r}}, \mathrm{M}) = \mathrm{N} \quad \mathtt{diff}(\mathsf{M}, \mathsf{M}') \cap \mathtt{en}(\mathsf{S}) = \emptyset}{{}_{\mathrm{M}}\langle\!\langle\, \mathsf{M} \,\rangle\!\rangle[\mathsf{S}] \xrightarrow{\mathrm{A}^{x}_{\mathtt{r}}} {}_{\mathrm{N}}\langle\!\langle\, \mathsf{M}' \,\rangle\!\rangle[\mathsf{S}]}$$

$$\text{(MS-AREACT)}\quad \frac{\mathsf{M} \xrightarrow{\mathrm{A}^{x}_{\mathtt{r}}} \mathsf{M}' \quad \mathsf{S} \xrightarrow{\mathrm{B}^{x}_{\mathtt{r}}} \mathsf{S}' \quad \mathrm{A} \neq \mathrm{B} \quad \mathcal{A}(\mathrm{A}_{\mathtt{r}}, \mathrm{M}) = \mathrm{N}}{{}_{\mathrm{M}}\langle\!\langle\, \mathsf{M} \,\rangle\!\rangle[\mathsf{S}] \xrightarrow{\tau} {}_{\mathrm{N}}\langle\!\langle\, \mathsf{M}' \,\rangle\!\rangle[\mathsf{S}']}$$

The notations $\mathsf{S} \stackrel{\tau}{\Longrightarrow} \mathsf{T}$ and $\mathsf{S} \stackrel{\mu}{\Longrightarrow} \mathsf{T}$, $\mu \neq \tau$, are defined in the same way as in Definition 2.1.

Rule (OPEN) prepares a cell to be fused with an enclosing cell or with a peer cell; the precondition guarantees that the cell may participate to a fusion. Rule (GRASP) lifts the fusion capability to groups by freezing them in a mreagent. Rules (FUSE) and (FUSE-I) define fusions between peer and nested cells, respectively. The new cell is created with the membrane name returned by the function $\mathcal{F}$. Rule (MEM-A) is a refinement of (MEM). It models possible side-effects on the membrane name due to interactions between membrane proteins and proteins outside the cell. Such interactions may activate membranes by changing their fusion capability, which is encoded in our formalism by membrane names. In a similar way, rule (MS-AREACT) refines (MS-REACT).

Example 3.2 A virus is an intracellular parasite that uses the infected cell replication machinery in order to duplicate its own genetic material. Usually, a virus consists of a genetic material (DNA or RNA), a capsid – a proteic structure providing protection of the genetic material (we use the term of nucleocapsid to denote both the capsid and the genetic material) –, and a possible envelope (stolen to the infected cell and used later on to infect other cells).

Below we encode in bioκ-calculus an influenza-like virus relying on similar descriptions in [1,4]. We focus on the infection part, as we cannot express any creation of new material. This part consists of the following steps:

(1) a protein-protein interaction between a virus membrane protein – the hemagglutinin HA – and a receptor – a glycoprotein GLY – on the cell's membrane; this activates GLY and prepares the cell to the phagocytosis;

(2) the phagocytosis of the virus occurs thus creating a new vesicle VES en-

gulfing the virus;

(3) a fusion occurs between the new vesicle and an endosomal membrane (EDSM) in the cytoplasm; this makes the virus enter the endosome;

(4) a further fusion occurs between the endosome and the virus that is now part of the nucleus; this leads to an exocytosis that eventually releases the virus nucleocapsid into the cytoplasm.

We consider the following rules:

$$\mathtt{r}_3 \ : ((\text{EDSM}, \text{VES}), \text{EDSM}) \ \in \mathcal{F}$$

$$\mathtt{r}_4 \ : \ ((\text{EDSM}, \text{VS}), \text{EDSM}) \ \in \mathcal{F}$$

The initial solution is $\mathsf{Virus}\ ,\ \mathsf{Cell}$ where the components are as follows:

$$\begin{array}{ll} \mathsf{Virus} & := \text{VS}(\!|\, \text{HA}(1) \,|\!)[\mathsf{Nucaps}] \\ \mathsf{Cell} & := \text{CLL}(\!|\, \text{GLY}(1)\ ,\ \mathsf{M}_c \,|\!)[\mathsf{Endosome}\ ,\ \mathsf{Cytosol}] \\ \mathsf{Endosome} & := \text{EDSM}(\!|\, \mathsf{M}_e \,|\!)[\mathsf{E}_s] \end{array}$$

We describe the last part of the infection pathway, assuming that the virus has already been engulfed in the host cell. We skip the first steps because we cannot express the phagocytosis of the virus. In Section 4 we will analyse the missing part. Therefore, let

$$\text{CLL}(\!|\, \mathsf{M}_c \,|\!)[\mathsf{Endosome}\ ,\ \text{VES}(\!|\, \text{GLY}(1) \,|\!)[\mathsf{Virus}]\ ,\ \mathsf{Cytosol}]$$

be the initial solution. A possible run is:

$$\begin{array}{ll} & \text{CLL}(\!|\, \mathsf{M}_c \,|\!)[\text{EDSM}(\!|\, \mathsf{M}_e \,|\!)[\mathsf{E}_s]\ ,\ \text{VES}(\!|\, \text{GLY}(1) \,|\!)[\mathsf{Virus}]\ ,\ \mathsf{Cytosol}] \\ \xrightarrow{\tau} & \text{CLL}(\!|\, \mathsf{M}_c \,|\!)[\text{EDSM}(\!|\, \mathsf{M}_e\ ,\ \text{GLY}(1) \,|\!)[\mathsf{Virus}\ ,\ \mathsf{E_s}]\ ,\ \mathsf{Cytosol}] & (\mathtt{r}_3) \\ \equiv & \text{CLL}(\!|\, \mathsf{M}_c \,|\!)[\text{EDSM}(\!|\, \mathsf{M}_e\ ,\ \text{GLY}(1) \,|\!)[\text{VS}(\!|\, \text{HA}(1) \,|\!)[\mathsf{Nucaps}]\ ,\ \mathsf{E}_s]\ ,\ \mathsf{Cytosol}] \\ \xrightarrow{\tau} & \text{CLL}(\!|\, \mathsf{M}_c \,|\!)[\text{EDSM}(\!|\, \mathsf{M}_e\ ,\ \text{GLY}(1)\ ,\ \text{HA}(1) \,|\!)[\mathsf{E}_s]\ ,\ \mathsf{Nucaps}\ ,\ \mathsf{Cytosol}] & (\mathtt{r}_4) \end{array}$$

Extensional semantics of cells: context bisimulation. The extensional semantics of Definition 2.4 must be refined in order to account with new transitions and mreagents. This refinement should for instance equate solutions such as $\text{A}(1^x)\ ,\ \text{M}(\!|\, \text{B}(1^x) \,|\!)[\mathsf{S}]$ and $\text{A}(1)\ ,\ \text{N}(\!|\, \text{B}(1) \,|\!)[\mathsf{S}]$ when $\mathcal{C}(\mathtt{r}) = ((\text{A}, 1, \emptyset), (\text{B}, 1, \emptyset))$ $\mathcal{D}(\mathtt{r}') = ((\text{A}, 1, \emptyset), (\text{B}, 1, \emptyset))$ and $\mathcal{A}(\text{B}_{\mathtt{r}}, \text{N}) = \text{M}'$ and $\mathcal{A}(\text{B}_{\mathtt{r}'}, \text{M}') = \text{N}$. This case is very similar to that of reversible reactions discussed in the previous section.

Definition 3.3 A *context bisimulation* is a symmetric binary relation $\mathfrak{R}$ between solutions such that it is a bisimulation and $\mathsf{S}\ \mathfrak{R}\ \mathsf{T}$ implies:

- if $\mathsf{S} \xrightarrow{\text{M}_{\mathtt{r}}} (\!|\, \mathsf{M}\ ;\ \mathsf{S}'' \,|\!) \cdot \mathsf{S}'$ then $\mathsf{T} \xRightarrow{\text{M}_{\mathtt{r}}} (\!|\, \mathsf{M}'\ ;\ \mathsf{T}'' \,|\!) \cdot \mathsf{T}'$ and, for every N, R, and N such that $\mathcal{F}(\mathtt{r}) = (\text{M} \otimes \text{N}, \text{N}')$, both $\Big(\mathsf{S}''\ ,\ \text{N}'(\!|\, \mathsf{M}\ ,\ \mathsf{N} \,|\!)[\mathsf{S}']\Big)\ \mathfrak{R}\ \Big(\mathsf{T}''\ ,\ \text{N}'(\!|\, \mathsf{M}'\ ,\ \mathsf{N} \,|\!)[\mathsf{T}'']\Big)$ and $\Big(\mathsf{S}'\ ,\ \text{N}'(\!|\, \mathsf{M}\ ,\ \mathsf{N} \,|\!)[\mathsf{S}''\ ,\ \mathsf{R}]\Big)\ \mathfrak{R}\ \Big(\mathsf{T}'\ ,\ \text{N}'(\!|\, \mathsf{M}'\ ,\ \mathsf{N} \,|\!)[\mathsf{T}''\ ,\ \mathsf{R}]\Big)$.

S is context bisimilar to T, written $\mathsf{S} \approx_c \mathsf{T}$, if $\mathsf{S}\ \mathfrak{R}\ \mathsf{T}$ for some context bisimulation $\mathfrak{R}$.

Context bisimulation retains the same substitutivity property of $\approx$.

Theorem 3.4 *$\approx_c$ is a congruence.*

Context bisimilarity retains a universal quantification that is hard to check in practice. One might wonder whether it is possible to simplify the definition. For example, instead of quantifying on cells, one may simply require the bisimilarity of components of mreactants, namely $\mathsf{M} \approx_c \mathsf{M}'$, $\mathsf{S}'' \approx_c \mathsf{T}''$, and $\mathsf{S}' \approx_c \mathsf{T}'$. It is easy to demonstrate that the induced equivalence, which we note $\approx_c^+$, is a congruence and $\approx_c^+ \subseteq \approx_c$. At the time we write this note it is not clear to us whether this containment is strict or not. This issue actually requires further investigations.

4 Translocation and phagocytosis

The $\mathtt{bio}\kappa$-calculus presented in Sections 2 and 3 has a limited expressive power: mechanisms such as translocation, where a single protein may enter a cell, or phagocytosis, where a cell may enter another cell cannot be described. For this reason we overlooked the first steps of the virus infection in Example 3.2. The integration of translocation and phagocytosis in $\mathtt{bio}\kappa$-calculus is not simple and admits several design choices. We discuss few possible formalisations below.

Translocation. Translocation is a mechanism enabling the transport of proteins through a membrane. This mechanism is very specific and controlled by particular membrane proteins that are different for each type of membrane.

Translocations usually do not transport the full proteins in one step because they are too big for traversing the membrane. This problem is solved in two ways. One way is that protein codes – the mRNA chains – interact with a ribosome (big proteic complex in charge of translating the code) and the interactions translate part of the code in the cell *and, at the same time,* create the encoded protein. Alternatively, biology uses ad-hoc proteins – the *chaperons* – that unfold the entering protein during the process (this is actually what happen at the end of the RTK-MAPK cascade described in Section 2).

We abstract from this low level mechanisms and assume that proteins may safely traverse the membrane. A first approximative definition of translocation might only check that the entering protein be disconnected and retains a suitable interface:

$$\textsc{a}(\phi+\psi)\ ,\ \textsc{m}(\!|\,\mathsf{M}\,|\!)[\mathsf{S}] \xrightarrow{\tau} \textsc{m}(\!|\,\mathsf{M}\,|\!)[\textsc{a}(\overline{\phi}+\psi)\ ,\ \mathsf{S}']$$

(according to our notation, both ϕ and ψ are v-h-maps, therefore $\mathtt{en}(\phi+\psi) = \emptyset$). This description is not satisfactory for at least two reasons. First, this rule admits a possibly infinite feeding of cells with membrane $\textsc{m}$ by proteins $\textsc{a}$. This is not the case in biology: after a certain number of translocations,

the membrane name changes, disabling further translocations. Second, the above rule makes the membrane play a passive role. This means that it is not possible to model the effects of organelle and chaperon proteins.

A better way is to model translocation as a decomplexation rule between an external protein already connected to the membrane (and nowhere else) and a membrane protein. To avoid the confusion of two different phenomena – simple decomplexations and decomplexations with translocations – we use a further function $\mathcal{T}$ from rule names to tuples $((\text{A}, i, \psi, \psi'), (\text{B}, j, \phi), \text{M}, \text{N})$. As usual we assume that there is no clash between rule names in the domain of $\mathcal{T}$ and the other functions that have been used in the paper. The two rules controlling translocations are:

$$\text{(TRS-P)}\quad \frac{\mathcal{T}(\mathtt{r}) = ((\text{A}, i, \psi, \psi'), (\text{B}, j, \phi), \text{M}, \text{N})}{\text{A}(i^x + \psi + \psi') \xrightarrow{\text{A}^x_{\mathtt{r}}} \mathbf{0}} \qquad \text{(TRS-M)}\quad \frac{\mathsf{M} \xrightarrow{\text{B}^x_{\mathtt{r}}} \mathsf{M}' \quad \mathcal{T}(\mathtt{r}) = ((\text{A}, i, \psi, \psi'), (\text{B}, j, \phi), \text{M}, \text{N})}{\text{M}\langle\!|\,\mathsf{M}\,|\!\rangle[\mathsf{S}] \xrightarrow{\text{B}^x_{\mathtt{r}}} \text{N}\langle\!|\,\mathsf{M}'\,|\!\rangle[\text{A}(i + \overline{\psi} + \psi')\ ,\ \mathsf{S}']}$$

In rule (TRS-P) the interface of A has exactly one site bound because the interfaces ψ and ψ' are v-h-maps according our notation. The interface ψ is being turned into $\overline{\psi}$ during the translocation, ψ' is unchanged. The protein A disappears during (TRS-P). Dually, the protein A appears during (TRS-M). The translocation will be the effect of the synchronisation (REACT).

Phagocytosis. Phagocytosis is a process allowing cells to enter other cells. Phagocytosis of $\text{M}\langle\!|\,\mathsf{M}\,|\!\rangle[\mathsf{S}]$ – usually a small cell – by $\text{N}\langle\!|\,\mathsf{N}\,|\!\rangle[\mathsf{T}]$ – usually a big cell – transports the former into the nucleus of the latter *by surrounding* $\text{M}\langle\!|\,\mathsf{M}\,|\!\rangle[\mathsf{S}]$ *with a new membrane that is part of* N. This surrounding mechanism is the problematic one because it amounts to split the host cell membrane in some "not local" way. For example the transition

$$\text{M}\langle\!|\,\mathsf{M}\,|\!\rangle[\mathsf{S}]\ ,\ \text{N}\langle\!|\,\mathsf{N}\,|\!\rangle[\mathsf{T}] \xrightarrow{\tau} \text{N}\langle\!|\,\mathsf{N}\,|\!\rangle[\text{N}'\langle\!|\,\mathbf{0}\,|\!\rangle[\text{M}\langle\!|\,\mathsf{M}\,|\!\rangle[\mathsf{S}]]\ ,\ \mathsf{T}]$$

is not very appropriate because the new membrane N′ is empty. As we don't have any mechanism for feeding the membrane yet, the solution ban complexations of the new membrane with proteins.

Actually, phagocytosis should be possible provided the membrane of the host cell had enough material for a new membrane. We therefore model phagocytosis as a decomplexation of two proteins in the membranes of the reactant cells with the side effect of splitting the host cell according to some predefined pattern. As for translocations, we use a new functions from rule names to tuples $((\text{A}, i, \psi), (\text{B}, j, \phi), \text{M}, \text{N}, \text{N}', \text{N}'', \mathsf{N}')$, where $\mathtt{de}(\mathsf{N}') = \emptyset$. The meaning of this tuple is the following: (A, i, ψ) and (B, j, ϕ) are the two proteins that decomplexate and are located in two membranes M and N, respectively. The name N′ is the one given at the new membrane surrounding the phagocytosed cell, N' is the membrane material of the new cell.

In the following rules, transition labels are extended with $\text{M}^x_{\mathtt{r}}$ and still

ranged over by μ. Let $\uplus$ denote disjoint union of sets. The rules defining phagocytosis are:

$$\text{(OPEN-P)}\quad \frac{\mathsf{M} \xrightarrow{\mathrm{A}^x_{\mathtt{r}}} \mathsf{M}' \quad \mathcal{P}(\mathtt{r}) = ((\mathrm{A}, i, \psi), (\mathrm{B}, j, \phi), \mathrm{M}, \mathrm{N}, \mathrm{N}', \mathrm{N}'', \mathsf{N}')}{\mathrm{M}(\!|\, \mathsf{M} \,|\!)[\mathsf{S}] \xrightarrow{\mathrm{M}^x_{\mathtt{r}}} (\!|\, \mathsf{M}' \,;\, \mathsf{S} \,|\!) \cdot \mathbf{0}}$$

$$\text{(OPEN-C)}\quad \frac{\mathsf{M} \xrightarrow{\mathrm{B}^x_{\mathtt{r}}} \prod_{k\in I\uplus J} \mathrm{B}_k(\sigma_k) \quad \mathcal{P}(\mathtt{r}) = ((\mathrm{A}, i, \psi), (\mathrm{B}, j, \phi), \mathrm{M}, \mathrm{N}, \mathrm{N}', \mathrm{N}'', \prod_{j\in J} \mathrm{B}_j(\sigma_j))}{\mathrm{N}(\!|\, \mathsf{M} \,|\!)[\mathsf{S}] \xrightarrow{\mathrm{N}^x_{\mathtt{r}}} (\!|\, \prod_{i\in I} \mathrm{B}_i(\sigma_i) \,;\, \mathsf{S} \,|\!) \cdot \mathbf{0}}$$

$$\text{(PHAGO)}\quad \frac{\begin{array}{c}\mathsf{S} \xrightarrow{\mathrm{M}^x_{\mathtt{r}}} (\!|\, \mathsf{M} \,;\, \mathsf{S}'' \,|\!) \cdot \mathsf{S}' \qquad \mathsf{T} \xrightarrow{\mathrm{N}^x_{\mathtt{r}}} (\!|\, \mathsf{N} \,;\, \mathsf{T}'' \,|\!) \cdot \mathsf{T}' \\ \mathcal{P}(\mathtt{r}) = ((\mathrm{A}, i, \psi), (\mathrm{B}, j, \phi), \mathrm{M}, \mathrm{N}, \mathrm{N}', \mathrm{N}'', \mathsf{N}')\end{array}}{\mathsf{S}\,,\,\mathsf{T} \xrightarrow{\tau} \mathsf{S}'\,,\,\mathsf{T}'\,,\,\mathrm{N}'(\!|\, \mathsf{N} \,|\!)[\mathrm{N}''(\!|\, \mathsf{N}' \,|\!)[\mathrm{M}(\!|\, \mathsf{M} \,|\!)[\mathsf{S}'']]\,,\,\mathsf{T}'']}$$

Rule (OPEN-P) defines the transition of the phogocytosed cell. The label $\mathrm{A}^x_{\mathtt{r}}$ of the membrane transition becomes $\mathrm{M}^x_{\mathtt{r}}$ in the cellular transition. This exposes the phagocytosis to the label. Similarly for the rule (OPEN-C). In (OPEN-C) the material needed for the new membrane surrounding the phagocytosed cell is removed from the host cell membrane. This material is restored in the rule (PHAGO).

It is not clear to us how close the above rules are to the biological phagocytosis. It is worth to observe that (OPEN-C) is computationally expensive, at least if compared to the other operations described in the paper. According to (OPEN-C), extracting a pattern of proteins out of a membrane amounts to a long sequel of checks that lock the membrane, thus inhibiting other interactions. It is an open question whether it is possible or not to design simpler and more basic mechanisms for phagocytosis.

Then we could finish the modelling of Example 3.2 by adding

$$\begin{array}{lll} \mathtt{r}_1 : & ((\mathrm{HA}, 1, \emptyset), (\mathrm{GLY}, 1, \emptyset)) & \in \mathcal{C} \\ (1) & (\mathrm{GLY}_{\mathtt{r}_1}, \mathrm{CLL}) \mapsto \mathrm{ACLL} & \in \mathcal{A} \\ \mathtt{r}_2 : & ((\mathrm{HA}, 1, \emptyset), (\mathrm{GLY}, 1, \emptyset)), \mathrm{VS}, \mathrm{ACLL}, \mathrm{CLL}, \mathrm{VES}, (\mathrm{GLY}(1)) & \in \mathcal{P} \end{array}$$

The formal rendering of Example 3.2 can now be stated from the beginning, namely before that the phagocytosis occurs.

$$\begin{array}{ll} & \mathrm{VS}(\!|\, \mathrm{HA}(1) \,|\!)[\mathsf{Nucaps}]\,,\,\mathrm{CLL}(\!|\, \mathrm{GLY}(1)\,,\,\mathsf{M_c} \,|\!)[\mathsf{Endosome}\,,\,\mathsf{Cytosol}] \\ \xrightarrow{\tau} & \mathrm{VS}(\!|\, \mathrm{HA}(1^x) \,|\!)[\mathsf{Nucaps}]\,,\,\mathrm{ACLL}(\!|\, \mathrm{GLY}(1^{\mathrm{x}})\,,\,\mathsf{M_c} \,|\!)[\mathsf{Endosome}\,,\,\mathsf{Cytosol}] \qquad (\mathtt{r}_1 - (1)) \\ \xrightarrow{\tau} & \mathrm{CLL}(\!|\, \mathsf{M}_c \,|\!)[\mathrm{VES}(\!|\, \mathrm{GLY}(1) \,|\!)[\mathrm{VS}(\!|\, \mathrm{HA}(1) \,|\!)[\mathsf{Nucaps}]]\,,\,\mathsf{Endosome}\,,\,\mathsf{Cytosol}] \quad (\mathtt{r}_2) \end{array}$$

5 Conclusions

We have presented a unique framework for modelling proteins and cells interactions – the bioκ-calculus. Protein interactions in bioκ-calculus are of two types: complexations and decomplexations; cell interactions in bioκ-calculus describe fusions. All interactions are "local" in the sense that they always involve two proteins. Fusions have been modelled by using an higher order semantics in the style of [9]. We have studied the operational semantics of bioκ-calculus and an extensional semantics of its – the bisimulation. The expressiveness has been analysed by modelling two significant systems and comparing them with similar ones that have been proposed in other algebraic approaches.

Some extensions of bioκ-calculus rules may be done without difficulties. In this paper we have discussed rules modelling translocations and phagocytosis, even if the latter ones are not very satisfactory. Other extensions have not been discussed, but are simple. For instance complexations and decomplexations might check a part of the interface without modifying it. It suffices to upgrade the functions $\mathcal{C}$ and $\mathcal{D}$ to tuples $((\text{A}, i, \phi, \phi'), (\text{B}, j, \psi, \psi'))$ and change rules (COM) and (DEC) into

$$\text{(COM)}\quad \frac{(\text{A}, a, \phi, \phi') \in \mathcal{C}(\mathtt{r}) \quad x \notin \mathtt{en}(\sigma)}{\text{A}(a + \phi + \phi' + \sigma) \xrightarrow{\text{A}^x_{\mathtt{r}}} \text{A}(a^x + \overline{\phi} + \phi' + \sigma)}$$

and

$$\text{(DEC)}\quad \frac{(\text{A}, a, \psi, \psi') \in \mathcal{D}(\mathtt{r})}{\text{A}(a^x + \psi + \psi' + \sigma) \xrightarrow{\text{A}^x_{\mathtt{r}}} \text{A}(a + \overline{\psi} + \psi' + \sigma)}$$

Other biological reactions have not yet been considered and are left to future work, such as those in [1] or in [4].

Extensional semantics of bioκ-calculus are an intriguing issue we plan to investigate in the future. In particular we are interested in mathematical tools and techniques that help in assessing properties of biological solutions. Such tools might be extensively used to predict outputs of experiments in vitro.

References

[1] Luca Cardelli. Brane calculi. In *CMSB*, pages 257–278, 2004.

[2] Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theoritical Computer Science*, 240(1):177–213, 2000.

[3] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoritical Computer Science*, 325(1):69–110, 2004.

[4] Vincent Danos and Sylvain Pradalier. Projective brane calculus. In *CMSB*, pages 134–148, 2004.

[5] Vincent Danos and Fabien Tarissan. Self-assembling graphs. In José Mira and José R. Álvarez, editors, *Mechanisms, Symbols, and Models Underlying Cognition*, volume 3561 of *Lecture Notes in Computer Science*, pages 498–507. Springer, June 2005.

[6] Robin Milner. *Communication and Concurrency*. International Series on Computer Science. Prentice Hall, 1989.

[7] Robin Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, Cambridge, 1999.

[8] Aviv Regev, William Silverman, and Ehud 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.

[9] Davide Sangiorgi. From π-calculus to Higher-Order π-calculus — and back. In M.-C. Gaudel and J.-P. Jouannaud, editors, *Proc. TAPSOFT'93*, volume 668, pages 151–166, 1993.