A Protein Substructure Based P System for Description and Analysis of Cell Signalling Networks

Thomas Hinze Thorsten Lenser Peter Dittrich

Friedrich Schiller University Jena
Bio Systems Analysis Group
Ernst-Abbe-Platz 1–4, D-07743 Jena, Germany

{hinze,thlenser,dittrich}@minet.uni-jena.de

Abstract. The way how cell signals are generated, encoded, transferred, modified, and utilised is essential for understanding information processing inside living organisms. The tremendously growing biological knowledge about proteins and their interactions draws a more and more detailed image of a complex functional network. Considering signalling networks as computing devices, the detection of structural principles, especially modularisation into subunits and interfaces between them, can help to seize ideas for their description and analysis. Algebraic models like P systems prove to be appropriate to this. We utilise string-objects to carry information about protein binding domains and their ligands. Embedding these string-objects into a deterministic graph structured P system with dynamical behaviour, we introduce a model that can describe cell signalling pathways on a submolecular level. Beyond questions of formal languages, the model facilitates tracing the evolutionary development from single protein components towards functional interacting networks. We exemplify the model by means of the yeast pheromone pathway.

1 Introduction

Protein signalling networks can be viewed as computational devices of the cell, triggering and directing responses to external inputs. Therefore, the utilisation of tools and techniques from computer science to study signalling networks constitutes an almost natural step. In this contribution, we propose a specialised version of the P system framework, which is a term-rewriting mechanism designed with cellular principles in mind [12, 13].

Cell signalling networks (CSNs) represent a class of biochemical reaction networks, set apart from others (such as metabolic networks) by an arrangement of special properties. One of these is the importance of the configuration of individual molecules (proteins) to their function in the network, which is exemplified by the activation of certain kinases via phosphorylation. Therefore, the protein constituents of CSNs should not be viewed as atomic objects, but rather as entities whose configuration can change over the course of time. Another distinguishing property of CSNs is the importance of their temporal behaviour. While the steady-state behaviour might be enough to characterise a metabolic network, the function of a CSN depends heavily on its temporal evolution. Thirdly, partitioning the whole pathway into several connected modules is thought to be a keystone for understanding molecular networks, a concept common to both metabolic and signalling systems. The formalism described here intends to incorporate the module concept by altering the established membrane framework from a tree-based to a graph-based structure. Similar approaches to P system structures are considered in tissue and population P systems [2, 10].

P systems with string-objects have already been considered in [12], while graph-based membrane structures were introduced in [14], and dynamics in [15, 18]. Our approach to temporal dynamics is based on the metabolic algorithm developed in [8], where a kinetic understanding of P system rewriting rules is explained and simulated. While these features have been introduced and investigated previously, the novelty of the approach presented here lies in their combination into one system, in order to define a framework suitable for modelling complex CSNs. A detailed account of the system is provided, and its suitability for modelling CSNs is demonstrated by an extensive example model of the yeast pheromone pathway.

2 Modelling Cell Signalling Networks

Modern molecular biology yields more and more data shedding light on signalling processes in the cell. In order to keep up with these developments, theoretical biologists and computer scientists have to provide modelling formalisms capable of integrating this growing knowledge. This section will briefly review common practices in modelling CSNs and introduce the advantages of using the P systems approach for this task, together with the extensions and novelties that are proposed in this contribution.

The most common class of formal CSN models consists of analytical models, based on differential equations. These systems are relatively straightforward to set up from a reaction network, and a plethora of tools for their numerical evaluation is available. Unfortunately, differential equations do not allow to include much human-readable information into the model, so that large models quickly become incomprehensible. Additionally, due to their continuous nature, these approaches are usually not tractable by tools and theory of computer science.

A classic step from continuous to discrete models leads to stochastic approaches, in which the number of molecules of each molecular species is assumed to be a random variable with temporally varying probability distribution. The conversion of a set of reactions into a stochastic system description has been adressed by a range of publications, mostly building on the fundamental work by Gillespie [5].

Algebraic approaches, drawing heavily on concepts from theoretical computer science, have several advantages to offer. They are well understood, cover a wide field of different modelling aspects, and each one comes with its own set of tools to analyse specific models. Additionally, they enable structuring and classification on several levels of abstraction. A short and by no means complete list of such formalisms would include: state-based systems such as abstract machines and X machines [4], process calculi such as π calculus [11], ambient calculus [3] and Petri nets [16], and term-rewriting systems based on Chomsky-grammars [17]. P systems, which represent an instance of the last category, are especially suited to develop models of cellular computation.

The general P system framework [13] is based on rewriting of multisets of molecule objects, which are contained in different compartments of the cell. Rewriting rules are localised to the compartments, so that object-processing depends on the current localisation. Objects can move between compartments, allowing the flow of a signal through the system. Here, we extend this formalism with a set of concepts that are essential for modelling CSNs.

By considering string-objects, we allow the substructures and properties of individual proteins to carry information. Extending the original concept of membranes to the more abstract view of distinguished modules in the CSN leads to models built out of coherent components, which are easier to create, maintain and re-use. In order to enable detailed studies on the temporal evolution of the system, we replace the maximally parallel rewriting from the original framework with a mechanism that is based on reaction kinetics. For each rewriting rule, the number of applications per turn is given by a kinetic function, depending on the current configuration of the module. This way, a deterministic system evolution is obtained.

3 System Description

We introduce a deterministic rewriting P system based on multisets of string-objects. The system description combines aspects of formal languages with numeric evaluations for handling of object selection and multiplicities. String-objects are composed in a way to encode information about protein substructures and specific protein properties.

Formal Language Prerequisites

We denote the empty word by ε. The concatenation of formal languages L_1 and L_2 over a common alphabet Σ is written as $L_1 \otimes L_2 = \{uv \mid u \in L_1 \wedge v \in L_2\}$. $\mathcal{P}(L)$ denotes the power set of L. Let A be an arbitrary set and $\mathbf{N}$ the set of natural numbers including zero. A multiset over A is a mapping $F : A \longrightarrow \mathbf{N} \cup \{\infty\}$. $F(a)$, also denoted as $[a]_F$, specifies the multiplicity of $a \in A$ in F. Multisets can be written as an elementwise enumeration of the form $\{(a_1, F(a_1)), (a_2, F(a_2)), \ldots\}$ since $\forall (a, b_1), (a, b_2) \in F : b_1 = b_2$. The support $supp(F) \subseteq A$ of F is defined by $supp(F) = \{a \in A \mid F(a) > 0\}$. A multiset F over A is said to be empty iff $\forall a \in A : F(a) = 0$. The cardinality $|F|$ of F over A is $|F| = \sum_{a \in A} F(a)$. Let F_1 and F_2 be multisets over A. F_1 is a subset of F_2, denoted as $F_1 \subseteq F_2$, iff $\forall a \in A : (F_1(a) \leq F_2(a))$. Multisets F_1 and F_2 are equal iff $F_1 \subseteq F_2 \wedge F_2 \subseteq F_1$. The intersection $F_1 \cap F_2 = \{(a, F(a)) \mid a \in A \wedge F(a) = \min(F_1(a), F_2(a))\}$, the multiset sum $F_1 \uplus F_2 = \{(a, F(a)) \mid a \in A \wedge F(a) = F_1(a) + F_2(a)\}$, and the multiset difference $F_1 \ominus F_2 = \{(a, F(a)) \mid a \in A \wedge F(a) = \max(F_1(a) - F_2(a), 0)\}$ form multiset operations. Multiplication of a multiset $F = \{(a, F(a)) \mid a \in A\}$ with a scalar c, denoted $\mathrm{c} \cdot F$, is defined by $\{(a, \mathrm{c} \cdot F(a)) \mid a \in A\}$. The term $\langle A \rangle = \{F : A \longrightarrow \mathbf{N} \cup \{\infty\}\}$ describes the set of all multisets over A.

Definition of the System

Let $\mathbf{N}_+ = \mathbf{N} \setminus \{0\}$ be the set of natural numbers without zero. A P system for describing CSNs of degree $n \in \mathbf{N}_+$ is a construct

$$\Pi_{CSN} = (V, V', E, M, n)$$

where V and V' are two alphabets; without loss of generality $\#, \neg, * \notin V \cup V'$. Furthermore, E and M specify channels and modules. The regular set

$$S = V^+ \otimes \left(\{\#\} \otimes ((V')^+ \cup \{\neg\} \otimes (V')^+ \cup \{*\})\right)^*$$

describes the syntax for string-objects. The leftmost substring from V^+ holds the protein identifier, followed by a finite number of protein property substrings from $(V')^+$ which are separated by #. For example, consider the string-object C:D#p#*#¬q identifying protein (complex) C:D with specified property p, a second arbitrary property (*), and without property q. Each protein property substring expresses a specific additional information about the protein, for instance whether it is activated with respect to a certain function or carries a ligand at a certain binding site. Two kinds of meta symbols are allowed. The symbol ¬ excludes the subsequent property but permits all other properties at this substring position. The placeholder * stands for an arbitrary (also unknown or unspecified) protein property substring. This way, uncertainty about the properties of proteins can be explicitly expressed. String-objects can be processed inside modules and they can move between modules along predefined channels (edges). The finite set of modules

$$M = \{M_1, \ldots, M_n\}$$

defines functional reaction units where multisets of string-objects can be modified by regulated rewriting. A module need not be embedded into a physical membrane, it just represents a space where reactions can occur. Multiple modules are allowed to share the same physical space. Modules are intended to form small units that fulfill well-defined functions. Each module M_i is defined as a tuple:

$M_i = (R_{i1}, \ldots, R_{ir_i}, \mathrm{f}_{i1}, \ldots, \mathrm{f}_{ir_i}, A_i)$ where

$R_{ij} \in \langle S \rangle \times \langle S \rangle$ is a reaction rule composed of two finite multisets

$\mathrm{f}_{ij} : \langle S \rangle \longrightarrow \mathbf{N}$ is a function corresponding to kinetics of reaction R_{ij}

$A_i \in \langle S \rangle$ is a multiset of axioms representing the initial contents of M_i

The set of channels is defined as

$$E \subseteq \{1, \dots, n\} \times \{1, \dots, n\} \times \mathcal{P}(S \times \{\mathrm{g} : \langle S \rangle^2 \longrightarrow \mathbf{N}\}) \times \mathbf{N} \quad \text{where}$$
$$e_{ij} = (i, j, I_{ij}, \mathrm{d}_{ij}) \in E$$

represents a directed channel from module M_i to module M_j. String-objects are allowed to pass the channel if they match the filter interface denoted by the construct

$$I_{ij} \subseteq \{(w, \mathrm{g}_{w,ij}) \mid w \in S \wedge \mathrm{g}_{w,ij} : \langle S \rangle^2 \longrightarrow \mathbf{N}\}.$$

The elements of I_{ij} correspond to the notion of filter patterns (receptors) w and concentration gradients $\mathrm{g}_{w,ij}$ between source module M_i and destination module M_j. Function $\mathrm{g}_{w,ij}$ marks the maximum capacity of the channel for string-objects matching the pattern w, depending on the contents of M_i and M_j. For simplicity, we assume that all filter interface patterns of channels beginning at the same module are pairwise disjoint to each other: $\bigcap_{j \in \{1,\dots,n\}} Match(supp(I_{ij})) = \emptyset \; \forall i \in \{1, \dots, n\}$ where $Match : \mathcal{P}(S) \longrightarrow \mathcal{P}(S)$ is defined in the next subsection. The support of the construct I_{ij} is defined in analogy to multisets. The natural number d_{ij} attached to each channel defines its time delay. Each passing string-object takes this amount of time when moving from module M_i to module M_j.

Matching and Matching Strategies

String-objects may contain excluding symbols $\neg$ and wild-cards $*$ to express partially incomplete knowledge about protein properties. Selecting string-objects for reactions and deciding which string-objects are allowed to pass a channel requires a definition of matching. Matching evaluates whether or not string-objects fit to each other, considering their identifiers and all possible combinations of protein property substrings resulting from their wild-carded patterns. We can distinguish between several matching strategies that differ by their handling of uncertainty. Extreme versions of matching are characterised by a loose and a strict strategy. A prerequisite of matching string-objects is their common number of property substrings.

In the symmetric relation $Match_{\mathrm{loose}}$, two string-objects match iff there is at least one common wild-card free representation. The loose strategy requires a minimum degree of similarity between objects with incomplete information. Uncertainty is interpreted as arbitrary replacements within the search space given by S.

$$\begin{aligned}
&Match_{\mathrm{loose}} \subseteq S \times S \\
&Match_{\mathrm{loose}} = \bigcup_{m \in \mathbf{N}} \{(p\#p_1\#p_2 \dots \#p_m,\ s\#s_1\#s_2 \dots \#s_m) \mid (p = s) \wedge \\
&\qquad\qquad \forall j \in \{1, \dots, m\} : [(p_j = s_j) \vee (p_j = *) \vee (s_j = *) \vee \\
&\qquad\qquad\qquad ((p_j = \neg q) \wedge (s_j \neq q)) \vee ((s_j = \neg q) \wedge (p_j \neq q))]\}
\end{aligned}$$

In contrast, the strict matching strategy follows the opposite intention. The two participating string-objects are interpreted as a pattern and a candidate for matching. Matching only occurs when the candidate $s\#s_1\#s_2\ldots\#s_m$ is a concretion of the pattern $p\#p_1\#p_2\ldots\#p_m$. The strict strategy embodies a matching with maximum degree of similarity between string-objects. Because of the different roles of the matching partners, the strict matching relation is not necessarily symmetric.

$$
\begin{aligned}
&Match_{\text{strict}} \subseteq S \times S\\
&Match_{\text{strict}} = \bigcup_{m\in\mathbf{N}} \{(p\#p_1\#p_2\ldots\#p_m,\ s\#s_1\#s_2\ldots\#s_m) \mid (p=s) \wedge\\
&\qquad\qquad\qquad \forall j \in \{1,\ldots,m\} : [(p_j = s_j) \vee (p_j = *) \vee ((p_j = \neg q) \wedge (s_j \neq q))]\}
\end{aligned}
$$

Let the regular set S be a syntax description for string-objects. Matching of a single string-object $w \in S$ to the search space generated by S is defined by

$$Match(w) = \{s \in S \mid (w,s) \in Match_{\text{X}}\}$$

with x = loose or x = strict. Consequently, we define the matching of a language $L \subseteq S$ by the function $Match : \mathcal{P}(S) \longrightarrow \mathcal{P}(S)$ with

$$Match(L) = \bigcup_{w\in L} Match(w).$$

Definition of System Behaviour

This subsection describes the dynamical behaviour of P systems Π_{CSN}. The multiset $L_i(t)$ denotes the contents of module M_i at time $t \in \mathbf{N}$. $L_i(t)$ is assumed to be empty for $t < 0$. It represents the configuration of the module controlled by a global clock and leads to the definition of the system step:

$$
\begin{aligned}
L_i(0) &= A_i\\
L_i'(t) &= L_i(t) \ominus Educts_i(t) \uplus Products_i(t)\\
L_i(t+1) &= L_i'(t) \ominus Outgoing_i(t) \uplus Incoming_i(t)
\end{aligned}
$$

A system step consists of four stages of modification, each of which is carried out synchronously in all modules. Firstly, the multiset of reaction educts is determined and removed from the module contents $L_i(t)$. Controlled application of local reaction rules transforms these educts into a multiset of products, which is added to the module contents without time delay. A subset of the new module contents can enter outgoing channels to move to (other) modules. Finally, arriving string-objects that have passed channels towards the module complete its contents.

Let $R_{ij} = (F_A, F_B) \in \langle S\rangle \times \langle S\rangle$ be a reaction rule in module M_i with $supp(F_A) = \{a_1,\ldots,a_p\}$ and $supp(F_B) = \{b_1,\ldots,b_q\}$. In terms of a chemical denotation, the rule R_{ij} can be written as

$$F_A(a_1) \cdot a_1 + \ldots + F_A(a_p) \cdot a_p \longrightarrow F_B(b_1) \cdot b_1 + \ldots + F_B(b_q) \cdot b_q$$

where $F_A(a_1), \ldots, F_A(a_p)$ encode stoichiometric factors of educts $a_1, \ldots, a_p$, and $F_B(b_1), \ldots, F_B(b_q)$ stoichiometric factors of products $b_1, \ldots, b_q$, respectively. All educt strings that match to the pattern a_k are provided by $Match(a_k)$. A combination of educt strings from $L_i(t)$ matching the left hand side of R_{ij} forms a multiset of string-objects used to apply the reaction once. Since the kinetic law, described by the scalar function f_{ij}, returns the number of applications of reaction rule R_{ij} within one step, the multiset of string-objects extracted from $L_i(t)$ to act as educts for R_{ij} can be written as $Educts_{ij}(t)$:

$$Educts_{ij}(t) = \biguplus_{e_1 \in Match(a_1)} \ldots \biguplus_{e_p \in Match(a_p)} \mathrm{f}_{ij}\big(\{(e_1,\infty),\ldots,(e_p,\infty)\} \cap L_i(t)\big) \cdot \big\{(e_1, F_{A_{ij}}(a_1)),\ldots,(e_p, F_{A_{ij}}(a_p))\big\}$$

Considering educts of all reaction rules $R_{i1}, \ldots, R_{ir_i}$ in module M_i, we achieve

$$Educts_i(t) = \biguplus_{j \in \{1,\ldots,r_i\}} Educts_{ij}(t).$$

Equivalently, the multiset of products obtained from reaction rule R_{ij} is determined by the multiset $Products_{ij}(t)$:

$$Products_{ij}(t) = \biguplus_{e_1 \in Match(a_1)} \ldots \biguplus_{e_p \in Match(a_p)} \mathrm{f}_{ij}\big(\{(e_1,\infty),\ldots,(e_p,\infty)\} \cap L_i(t)\big) \cdot \big\{(b_1, F_{B_{ij}}(b_1)),\ldots,(b_q, F_{B_{ij}}(b_q))\big\}$$

Considering products of all reaction rules $R_{i1}, \ldots, R_{ir_i}$ in module M_i, we achieve

$$Products_i(t) = \biguplus_{j \in \{1,\ldots,r_i\}} Products_{ij}(t).$$

Although the multiset difference always returns non-negative multiplicities, also in case of a lack of educt-objects, the number of product-objects is only determined by B_{ij}. This effect could be compensated by extension of multiset multiplicities to negative integers as well. This way, the requirement of mass-balance could formally be sustained without additional formalism. For sufficiently large numbers of proteins, however, this effect is negligible.

After performing the reactions, the multisets of outgoing and incoming string-objects are specified using $L'_i(t)$ and filter interfaces I_{ij}. Let

$$Outgoing_{ij}(t) = L'_i(t) \cap \big\{(v, \mathrm{g}_{w,ij}(L'_i(t), L'_j(t))\,\big|\, v \in S \wedge w \in supp(I_{ij}) \wedge v \in Match(w)\big\}$$

the multiset of transferred string-objects along the channel from M_i to M_j. We define:

$$\begin{aligned} Outgoing_i(t) &= \biguplus_{j \in \{1,\ldots,n\}} Outgoing_{ij}(t) \\ Incoming_i(t) &= \biguplus_{k \in \{1,\ldots,n\}} Outgoing_{ki}(t - \mathrm{d}_{ki}) \end{aligned}$$

Generated Language

This subsection specifies the configuration of system Π_{CSN} at time t and finally the generated formal language $L(\Pi_{CSN})$. The contents $L_i(t)$ of all modules M_i form the essential part of the system configuration. Since string-objects can take several time steps to pass channels, the system configuration at time t also subsumes all multisets $Outgoing_{ij}(\tau)$ with $\tau = 0, \ldots, t-1$. The configuration of module M_i at time t is a construct

$$C_i(t) = \left(L_i(t), \left(Outgoing_{ij}(\tau) \right)_{\substack{j=1,\ldots,n \\ \tau=0,\ldots,t-1}} \right).$$

Furthermore,

$$C_{\Pi_{CSN}}(t) = (C_1(t), \ldots, C_n(t)).$$

Π_{CSN} generates the language

$$L(\Pi_{CSN}) = supp \left(\biguplus_{t=0}^{\infty} \left(\biguplus_{i=1}^{n} L_i(t) \right) \right),$$

the set of string-objects that occur in any module during infinite execution of the system.

4 System Properties

P systems of the framework Π_{CSN} feature a combination of properties which are relevant to describe and analyse CSNs. Bringing together notions of substructured string-objects, configurable modules interconnected by channels, and a formalisation of deterministic dynamical system behaviour, the proposed approach lends itself to applications beyond classification of computational power. Further studies are focused on the evolutionary development from small low-structured subunits towards much more complex networks with shared resources. Establishing correlations between physical structures and biological functions is a key issue here. In preparation of these objectives, we have designed Π_{CSN} with the following properties.

Modularity: System composition of a finite number of interacting modules follows the idea of defined functional subunits. Modularisation can be seen as a powerful tool to represent the inherent structure of a complex system, its organisation, and its basic principles. Each module performs a specific set of reactions in an autonomous manner. Communication between modules is separated from reaction processes.

Static System Topology: Established CSNs own a static topology based on modules and directed channels resulting in a graph structure. Each of the channels acts as a filter with regard to both qualitative and quantitative aspects. Configurable patterns represent receptors to accept or reject proteins with specific properties. Maximum capacity as well as time delay reflect physical restrictions. In CSNs, channels often form cascades consisting of several stages with different protein ligands, complexes or activation state.

Ability to Identify Objects / Substructures: Each single (protein) molecule handled within the system is treated as an individual object. It identifies the underlaying protein and provides information about additional specific properties. Since reactions within CSNs often keep proteins but modify their properties, consideration of substructural information is essential in the model in order to handle combinatorial networks.

Flexibility in Level of Abstraction: The concept of substrings containing information about specific protein properties gives the system a high degree of flexibility in the level of abstraction. Wild-carded and excluding patterns enable coping with uncertainty. The way how incomplete information can be processed by reaction and transduction spans a wide range of detailedness. Consequences of uncertainty to the system behaviour become obvious.

Determinism: The system Π_{CSN} is constructed to work in a deterministic manner. Subsequent execution of system steps leads to a unique path through configurations. In terms of the computational path, determinism implies confluence.

Computational Tractability: Determinism and finite system components facilitate simulations in silico. All aspects of the system description and system behaviour are formalised for Π_{CSN}. Sets, multisets, and functions used within the system are polynomially decidable with regard to the number of objects. Software tools like computer algebra systems can serve for further analysis.

Computational Completeness: The ability to use P systems as models for computation can be seen as a fundamental aspect in the field of membrane computing [9]. Investigations about their (sub)classes of computability depending on certain combinations of system properties and restrictions motivate theoretically inspired contributions to the field. Reaction networks are known to be computational complete, constructively shown in [7]. Each module M_i of Π_{CSN} forms such a reaction network.

5 Example: Signal Transduction in the Yeast Pheromone Pathway

The pheromone response pathway in *Saccharomyces cerevisiae* (yeast hereafter) is among the best understood signalling pathways in eukaryotes. Its constituents (proteins) and their interactions have been subject of a great variety of studies, and the overall picture of how these act together in the pathway is rapidly emerging (see [1] for a review). Yeast cells exist in two mating types, *MAT***a** and $MAT\alpha$, which secrete pheromones to stimulate mating behaviour in the opposite type. Effects of this stimulation are the arrest of the cell cycle, changes in the expression of around 200 genes, and even an elongation of the cell in the direction of its mating partner.

To show the suitability of the P system formalism to model cell signalling networks, we have decided to convert the comprehensive yeast pheromone pathway model by Kofahl and Klipp [6] into our framework. The pathway consists of different modules: the G-protein-coupled receptor (M_1, corresponding to receptor activation and G-protein cycle in [6]), formation of the scaffold protein (M_2), MAPK cascade (M_3), Fus3 phosphorylation cycle (M_4), and responses of the cell to the activation of the pathway (not modelled here).

In the module M_1, the receptor protein Ste2 is activated by α-Factor pheromone. In response to activation of Ste2, the trimeric G-protein breaks into its α and $\beta\gamma$ subunits, of which the latter passes on the signal. The "service module" M_2 binds the three components of the MAPK cascade (Ste11, Ste7 and Fus3) to the scaffold protein Ste5, which is then bound by $G\beta\gamma$. Ste20 can now bind to this complex (creating complex C), where it phosphorylates Ste11 and thus triggers the MAPK cascade (M_3). In this cascade, Ste11 activates Ste7, which in turn activates Fus3. Activated Fus3 is then split off (leaving complex C') and moves into the nucleus. Unphosphorylated Fus3 can again bind to C', creating a cycle which amplifies the response.

$$\varPi_{pheromone} = (V, V', E, M, 4)$$

$$\begin{aligned}
V &= \{\mathrm{S}te2, \alpha, G\beta\gamma, G\alpha, Ste5, Ste11, Ste7, Fus3, Ste20, C, C', :\} \\
V' &= \{\mathrm{a}, GDP, GTP, p\}
\end{aligned}$$

$$\begin{aligned}
M &= \{M_1, M_2, M_3, M_4\} \\
E &= \{(1, 3, I_{13}, \mathrm{d}_{13}), (3, 1, I_{31}, \mathrm{d}_{31}), (2, 3, I_{23}, \mathrm{d}_{23}), \\
&\qquad (3, 2, I_{32}, \mathrm{d}_{32}), (3, 4, I_{34}, \mathrm{d}_{34})\}
\end{aligned}$$

$$\begin{aligned}
I_{13} &= \{(\mathrm{G}\beta\gamma,\ \mathrm{g}_{13}(L_1'(t), L_3'(t)) = \lfloor \mathrm{k}_{g_{13}}[G\beta\gamma]_{L_1'(t)} \rfloor)\} \\
I_{31} &= \{(\mathrm{G}\beta\gamma,\ \mathrm{g}_{31}(L_3'(t), L_1'(t)) = \lfloor \mathrm{k}_{g_{31}}[G\beta\gamma]_{L_3'(t)} \rfloor)\} \\
I_{23} &= \{(\mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3,
\end{aligned}$$

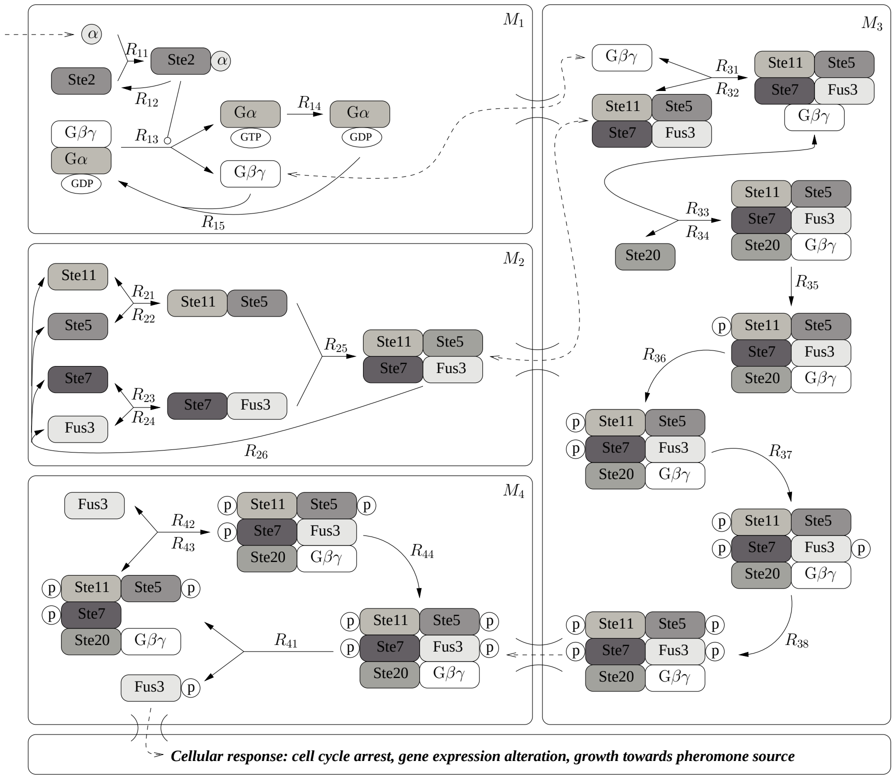

Fig. 1. Module diagram and reaction scheme of the yeast pheromone pathway. At bidirectional arrows, the upper rule corresponds to the rightward direction. Due to the combinatorial complexity, not all sequences of phosphorylation and only one dissociation of complex C are shown in M_3.

$$
\begin{aligned}
&\qquad \mathrm{g}_{23}(L_2'(t), L_3'(t)) = \lfloor \mathrm{k}_{g_{23}} [Ste5{:}Ste11{:}Ste7{:}Fus3]_{L_2'(t)} \rfloor)\} \\
I_{32} &= \{(\mathrm{Ste5}{:}Ste11{:}Ste7{:}Fus3, \\
&\qquad \mathrm{g}_{32}(L_3'(t), L_2'(t)) = \lfloor \mathrm{k}_{g_{32}} [Ste5{:}Ste11{:}Ste7{:}Fus3]_{L_3'(t)} \rfloor)\} \\
I_{34} &= \{(\mathrm{C}\#p\#p\#p\#p,\ \mathrm{g}_{34}(L_3'(t), L_4'(t)) = \lfloor \mathrm{k}_{g_{34}} [C\#p\#p\#p\#p]_{L_3'(t)} \rfloor)\}
\end{aligned}
$$

$$
\begin{aligned}
M_1 &= (R_{11}, R_{12}, R_{13}, R_{14}, R_{15}, \mathrm{f}_{11}, \mathrm{f}_{12}, \mathrm{f}_{13}, \mathrm{f}_{14}, \mathrm{f}_{15}, A_1) \quad with \\
R_{11} &= \mathrm{Ste2}\#\neg a \ + \ \alpha \ \longrightarrow \ \mathrm{Ste2}\#a \\
R_{12} &= \mathrm{Ste2}\#a \ \longrightarrow \ \mathrm{Ste2}\#\neg a \\
R_{13} &= \mathrm{G}\beta\gamma{:}G\alpha\#GDP \ + \ \mathrm{Ste2}\#a \ \longrightarrow \ \mathrm{G}\alpha\#GTP \ + \ \mathrm{G}\beta\gamma \ + \ \mathrm{Ste2}\#a \\
R_{14} &= \mathrm{G}\alpha\#GTP \ \longrightarrow \ \mathrm{G}\alpha\#GDP \\
R_{15} &= \mathrm{G}\beta\gamma \ + \ \mathrm{G}\alpha\#GDP \ \longrightarrow \ \mathrm{G}\beta\gamma{:}G\alpha\#GDP \\
\mathrm{f}_{11}(L_1(t)) &= \left\lfloor \mathrm{k}_{11} [\mathrm{Ste2}\#\neg a]_{L_1(t)} [\alpha]_{L_1(t)} (1/\mathrm{V}_1^2) \right\rfloor
\end{aligned}
$$

$$\begin{aligned}
\mathrm{f}_{12}(L_1(t)) &= \left\lfloor \mathrm{k}_{12}[\mathrm{S}te2\#a]_{L_1(t)}(1/\mathrm{V}_1)\right\rfloor \\
\mathrm{f}_{13}(L_1(t)) &= \left\lfloor \mathrm{k}_{13}[\mathrm{G}\beta\gamma{:}G\alpha\#GDP]_{L_1(t)}[\mathrm{S}te2\#a]_{L_1(t)}(1/\mathrm{V}_1^2)\right\rfloor \\
\mathrm{f}_{14}(L_1(t)) &= \left\lfloor \mathrm{k}_{14}[\mathrm{G}\alpha\#GTP]_{L_1(t)}(1/\mathrm{V}_1)\right\rfloor \\
\mathrm{f}_{15}(L_1(t)) &= \left\lfloor \mathrm{k}_{15}[\mathrm{G}\beta\gamma]_{L_1(t)}[\mathrm{G}\alpha\#GTP]_{L_1(t)}(1/\mathrm{V}_1^2)\right\rfloor \\
A_1 &= \{(\alpha, 6\cdot 10^{17}), (\mathrm{S}te2\#\neg a, 10^{18}), (\mathrm{G}\beta\gamma{:}G\alpha\#GDP, 10^{18})\}
\end{aligned}$$

$$\begin{aligned}
M_2 &= (R_{21}, R_{22}, R_{23}, R_{24}, R_{25}, R_{26}, \mathrm{f}_{21}, \mathrm{f}_{22}, \mathrm{f}_{23}, \mathrm{f}_{24}, \mathrm{f}_{25}, \mathrm{f}_{26}, A_2) \quad \mathrm{w}ith \\
R_{21} &= \mathrm{S}te5 \ + \ \mathrm{S}te11 \ \longrightarrow \ \mathrm{S}te5{:}Ste11 \\
R_{22} &= \mathrm{S}te5{:}Ste11 \ \longrightarrow \ \mathrm{S}te5 \ + \ \mathrm{S}te11 \\
R_{23} &= \mathrm{S}te7 \ + \ \mathrm{F}us3 \ \longrightarrow \ \mathrm{S}te7{:}Fus3 \\
R_{24} &= \mathrm{S}te7{:}Fus3 \ \longrightarrow \ \mathrm{S}te7 \ + \ \mathrm{F}us3 \\
R_{25} &= \mathrm{S}te5{:}Ste11 \ + \ \mathrm{S}te7{:}Fus3 \ \longrightarrow \ \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3 \\
R_{26} &= \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3 \ \longrightarrow \ \mathrm{S}te5 \ + \ \mathrm{S}te11 \ + \ \mathrm{S}te7 \ + \ \mathrm{F}us3 \\
\mathrm{f}_{21}(L_2(t)) &= \left\lfloor \mathrm{k}_{21}[\mathrm{S}te5]_{L_2(t)}[\mathrm{S}te11]_{L_2(t)}(1/\mathrm{V}_2^2)\right\rfloor \\
\mathrm{f}_{22}(L_2(t)) &= \left\lfloor \mathrm{k}_{22}[\mathrm{S}te5{:}Ste11]_{L_2(t)}(1/\mathrm{V}_2)\right\rfloor \\
\mathrm{f}_{23}(L_2(t)) &= \left\lfloor \mathrm{k}_{23}[\mathrm{S}te7]_{L_2(t)}[\mathrm{F}us3]_{L_2(t)}(1/\mathrm{V}_2^2)\right\rfloor \\
\mathrm{f}_{24}(L_2(t)) &= \left\lfloor \mathrm{k}_{24}[\mathrm{S}te7{:}Fus3]_{L_2(t)}(1/\mathrm{V}_2)\right\rfloor \\
\mathrm{f}_{25}(L_2(t)) &= \left\lfloor \mathrm{k}_{25}[\mathrm{S}te5{:}Ste11]_{L_2(t)}[\mathrm{S}te7{:}Fus3]_{L_2(t)}(1/\mathrm{V}_2^2)\right\rfloor \\
\mathrm{f}_{26}(L_2(t)) &= \left\lfloor \mathrm{k}_{26}[\mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3]_{L_2(t)}(1/\mathrm{V}_2)\right\rfloor \\
A_2 &= \{(\mathrm{S}te5, 9.5\cdot 10^{16}), (\mathrm{S}te11, 9.5\cdot 10^{16}), (\mathrm{S}te7, 2\cdot 10^{16}), (\mathrm{F}us3, 2\cdot 10^{16})\}
\end{aligned}$$

$$\begin{aligned}
M_3 &= (R_{31}, R_{32}, R_{33}, R_{34}, R_{35}, R_{36}, R_{37}, R_{38}, \\
&\qquad \mathrm{f}_{31}, \mathrm{f}_{32}, \mathrm{f}_{33}, \mathrm{f}_{34}, \mathrm{f}_{35}, \mathrm{f}_{36}, \mathrm{f}_{37}, \mathrm{f}_{38}, A_3) \quad \mathrm{w}ith \\
R_{31} &= \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3 \ + \ \mathrm{G}\beta\gamma \ \longrightarrow \ \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma \\
R_{32} &= \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma \ \longrightarrow \ \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3 \ + \ \mathrm{G}\beta\gamma \\
R_{33} &= \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma \ + \ \mathrm{S}te20 \ \longrightarrow \ \mathrm{C}\#\neg p\#\neg p\#\neg p\#\neg p \\
R_{34} &= \mathrm{C}\#\ast\#\ast\#\ast\#\ast \ \longrightarrow \ \mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma \ + \ \mathrm{S}te20 \\
R_{35} &= \mathrm{C}\#\neg p\#\neg p\#\ast\#\ast \ \longrightarrow \ C\#\neg p\#p\#\ast\#\ast \\
R_{36} &= \mathrm{C}\#\neg p\#p\#\neg p\#\ast \ \longrightarrow \ C\#\neg p\#p\#p\#\ast \\
R_{37} &= \mathrm{C}\#\neg p\#p\#p\#\neg p \ \longrightarrow \ C\#\neg p\#p\#p\#p \\
R_{38} &= \mathrm{C}\#\neg p\#p\#p\#p \ \longrightarrow \ C\#p\#p\#p\#p \\
\mathrm{f}_{31}(L_3(t)) &= \left\lfloor \mathrm{k}_{31}[\mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3]_{L_3(t)}[\mathrm{G}\beta\gamma]_{L_3(t)}(1/\mathrm{V}_3^2)\right\rfloor \\
\mathrm{f}_{32}(L_3(t)) &= \left\lfloor \mathrm{k}_{32}[\mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor \\
\mathrm{f}_{33}(L_3(t)) &= \left\lfloor \mathrm{k}_{33}[\mathrm{S}te5{:}Ste11{:}Ste7{:}Fus3{:}G\beta\gamma]_{L_3(t)}[\mathrm{S}te20]_{L_3(t)}(1/\mathrm{V}_3^2)\right\rfloor \\
\mathrm{f}_{34}(L_3(t)) &= \left\lfloor \mathrm{k}_{34}[\mathrm{C}\#\ast\#\ast\#\ast\#\ast]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor
\end{aligned}$$

$$
\begin{aligned}
\mathrm{f}_{35}(L_3(t)) &= \left\lfloor \mathrm{k}_{35}[\mathrm{C}\#\neg p\#\neg p\#*\#*]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor\\
\mathrm{f}_{36}(L_3(t)) &= \left\lfloor \mathrm{k}_{36}[\mathrm{C}\#\neg p\#p\#\neg p\#*]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor\\
\mathrm{f}_{37}(L_3(t)) &= \left\lfloor \mathrm{k}_{37}[\mathrm{C}\#\neg p\#p\#p\#\neg p]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor\\
\mathrm{f}_{38}(L_3(t)) &= \left\lfloor \mathrm{k}_{38}[\mathrm{C}\#\neg p\#p\#p\#p]_{L_3(t)}(1/\mathrm{V}_3)\right\rfloor\\
A_3 &= \{(\mathrm{S}te20, 6\cdot 10^{17})\}
\end{aligned}
$$

$$
\begin{aligned}
M_4 &= (R_{41}, R_{42}, R_{43}, R_{44}, \mathrm{f}_{41}, \mathrm{f}_{42}, \mathrm{f}_{43}, \mathrm{f}_{44}, A_4) \quad \mathrm{w}ith\\
R_{41} &= \mathrm{C}\#p\#p\#p\#p \ \longrightarrow\ C'\#p\#p\#p\ +\ Fus3\#p\\
R_{42} &= \mathrm{C}'\#p\#p\#p\ +\ Fus3\#\neg p \ \longrightarrow\ C\#p\#p\#p\#\neg p\\
R_{43} &= \mathrm{C}\#p\#p\#p\#\neg p \ \longrightarrow\ C'\#p\#p\#p\ +\ Fus3\#\neg p\\
R_{44} &= \mathrm{C}\#p\#p\#p\#\neg p \ \longrightarrow\ C\#p\#p\#p\#p\\
\mathrm{f}_{41}(L_4(t)) &= \left\lfloor \mathrm{k}_{41}[\mathrm{C}\#p\#p\#p\#p]_{L_4(t)}(1/\mathrm{V}_4)\right\rfloor\\
\mathrm{f}_{42}(L_4(t)) &= \left\lfloor \mathrm{k}_{42}[\mathrm{C}'\#p\#p\#p]_{L_4(t)}[\mathrm{F}us3]_{L_4(t)}(1/\mathrm{V}_4^2)\right\rfloor\\
\mathrm{f}_{43}(L_4(t)) &= \left\lfloor \mathrm{k}_{43}[\mathrm{C}\#p\#p\#p\#\neg p]_{L_4(t)}(1/\mathrm{V}_4)\right\rfloor\\
\mathrm{f}_{44}(L_4(t)) &= \left\lfloor \mathrm{k}_{44}[\mathrm{C}\#p\#p\#p\#\neg p]_{L_4(t)}(1/\mathrm{V}_4)\right\rfloor\\
A_4 &= \{(\mathrm{F}us3, 6\cdot 10^{17})\}
\end{aligned}
$$

While most parts of the model given in [6] were directly adapted, a few slight changes had to be introduced. Concentrations, which were given in nM in the original, were converted into molecule numbers. In order to compensate for this effect on the reaction kinetics, these were extended by the volume V_i of each module as a normalisation constant. Values for all parameters k_{ij} appearing in the reaction kinetics are not given here, but can be calculated from the data given in [6]. In accordance with the original model, reaction kinetics consist of mass-action formulations, which have to be rounded in order to yield integer values as results.

6 Discussion

An important aspect of the system is its capability to deal with the combinatorial complexity arising when proteins incorporating multiple sites for modification and binding are involved. The usage of wild-cards and exclusions allows a compact formulation of systems that would be substantially larger if only atomic objects were considered. This major advantage could even be extended by the introduction of generic logical expressions into the protein descriptions.

It is important to mention two potential limitations of the system. On the one hand, the reaction kinetics mechanism might produce imprecise results in case of small object numbers, due to the fact that educts of multiple reactions can sum up to more proteins than currently present. As a possible solution, we propose the extension of the multiset definition to negative multiplicities, so that modules can formally contain negative numbers of objects. In this case, the kinetic formulation would ensure any educt with a negative multiplicity could not participate in a reaction, but rather would have to be refilled again. On the other hand, a potential constraint comes from the way in which the kinetic functions are applied to each combination of proteins matching the right side of a reaction rule. In order to work precisely, this approach requires the kinetic functions to be linear. Therefore, it is advisable not to use wild-carded reaction rules with non-linear kinetics. Both of these points require further research to maximise the system's usability.

The choice of the matching strategy strongly influences the system behaviour. Strict matching implies maximum specificity of the reactions. In contrast, to involve a large pool of protein configurations, a loose matching should be preferred. Special application scenarios can be tackled by additional matching strategies.

7 Conclusions

The introduced model Π_{CSN} intends to combine advantages of P systems with mechanisms observed in cell signalling networks. It is conceived as a description, analysis, and prediction tool for ongoing studies about the evolutionary development of protein structures, their properties, interactions, and resulting network functions. To this end, we have integrated string-objects and a modular architecture into a deterministic framework. In future, simulations as well as theoretical investigations will lead the way towards a deeper understanding of the correlation between CSN structure and function.

Acknowledgements

This work is part of the ESIGNET project (Evolving Cell Signalling Networks in Silico), which has received research funding from the European Community's Sixth Framework Programme (project no. 12789). Further funding from the Federal Ministry of Education and Research (BMBF, grant 0312704A) is acknowledged.

References

1. L. Bardwell. *A walk-through of the yeast mating pheromone response pathway.* Peptides 25:1465-1476, 2004
2. F. Bernardini, M. Gheorghe. *Population P systems.* Journal of Universal Computer Science 10(5):509-539, 2004
3. L. Cardelli, A.D. Gordon. *Mobile Ambients.* LNCS 1378, Springer-Verlag London, 1998
4. S. Eilenberg. *Automata, Languages, and Machines.* Academic Press New York, 1976
5. D.T. Gillespie. *Exact stochastic simulation of coupled chemical reactions.* Journal of Physical Chemistry 81:2340-2361, 1977

6. B. Kofahl, E. Klipp. *Modelling the dynamics of the yeast pheromone pathway.* Yeast 21:831-850, 2004
7. M.O. Magnasco. *Chemical Kinetics is Turing Universal.* Physical Review Letters 78(6):1190-1193, 1997
8. V. Manca, L. Bianco, F. Fontana. *Evolution and oscillation in P systems: Applications to biological phenomena.* LNCS 3365, Springer-Verlag Berlin, 2005
9. C. Martin-Vide, G. Păun. *Computing with Membranes (P Systems): Universality Results.* LNCS 2055, Springer-Verlag London, 2001
10. C. Martin-Vide, G. Păun, J. Pazos, A. Rodriguez-Paton. *Tissue P Systems.* Theoretical Computer Science 296(2):295-326, 2003
11. R. Milner. *Communicating and Mobile Systems: the Pi-Calculus.* Cambridge University Press, 1999
12. G. Păun. *Computing with Membranes.* Journal of Computer and System Sciences 61(1):108-143, 2000
13. G. Păun. *Membrane Computing: An Introduction.* Springer-Verlag Berlin 2002
14. G. Păun, Y. Sakakibara, T. Yokomori. *P Systems on Graphs of Restricted Forms.* Publicationes Mathematicae 60, 2002
15. D. Pescini, D. Besozzi, G. Mauri, C. Zandron. *Dynamical probabilistic P systems.* International Journal of Foundations of Computer Science 17(1):183-195, 2006
16. J.L. Peterson. *Petri Net Theory and the Modelling of Systems.* Prentice Hall, 1961
17. G. Rozenberg, A. Salomaa (Eds.). *Handbook of formal languages 1-3.* Springer-Verlag Berlin, 1999
18. Y. Suzuki, H. Tanaka. *Symbolic chemical system based on abstract rewriting and its behavior pattern.* Artificial Life and Robotics 1:211-219, 1997