



# Compact Representation of Photosynthesis Dynamics by Rule-based Models<sup>1</sup>

L. Brim, J. Nižnan, D. Šafránek

*Faculty of Informatics, Masaryk University  
Brno, Czech Republic*

---

## Abstract

Traditional mathematical models of photosynthesis are based on mass action kinetics of light reactions. This approach requires the modeller to enumerate all the possible state combinations of the modelled chemical species. This leads to combinatorial explosion in the number of reactions although the structure of the model could be expressed more compactly. We explore the use of rule-based modelling, in particular, a simplified variant of Kappa, to compactly capture and automatically reduce existing mathematical models of photosynthesis. Finally, the reduction procedure is implemented in BioNetGen language and demonstrated on several ODE models of photosynthesis processes.

*Keywords:* Biological models, model annotation, systems biology, simulation, database

---

## 1 Introduction

Photosynthesis is one of the most important biophysical processes driving life on Earth. Most life forms, including humans, depend on photosynthesis that transforms energy of solar radiation into energy-rich organic matter, releases oxygen that we breathe, and removes excess carbon dioxide from the atmosphere that would threaten the Earth's energy balance. Adding to the relevance of photosynthesis, significant expectations emerged lately in connection with potential human interventions in the global carbon cycle – among the considered alternatives are the higher generation biofuels [1] or biomineralization by point-source carbon capture [11].

Current coarse-grained mathematical models of photosynthesis [13] cover the known parts of the entire process. They build up the light reactions dynamics from simplified interactions on and inbetween complicated protein complexes involved in the transfer of the energy from light into the cell. Many different local modifications at these protein structures are traversed after reception of the photon. To capture

---

<sup>1</sup> The work has been supported by the EC OP project No. CZ.1.07/2.3.00/20.0256.

the process mechanistically, many elementary chemical reactions connect together to form the model. Each effective structure combination has to be enumerated in order to assign the appropriate kinetic laws. This inevitable expansion then leads to combinatorial explosion in the number of possible complexes.

In [20] we have developed an online repository for mathematical models of photosynthesis. That effort has opened many questions regarding the differing levels of available models and the problem of their formal representation in a single suitably expressive formalism. After several years of interactions with modellers targeting photosynthesis, we now attempt to move towards practical applications of state-of-the-art formal methods in that field.

Rule-based modelling [8,5,4] is an approach that has been developed to tackle primarily the complexity of cell signalling systems where combinatorial explosion comes from configurations of phosphate bindings to specific sites of a protein. In particular, it allows us to compactly represent complicated models that would be tedious to specify using traditional reaction-based methods [6,12]. The interactions between proteins are represented using rules at the level of functional components. In photosynthesis, there occurs a number of specific protein complex modifications that are in abstract essence similar to phosphorylation though crucially different at the side of physics. The two most well-known representatives of rule-based modelling languages are Kappa [8] and BioNetGen Language [9].

On the theoretical side, we contribute to the increasing set of algebraic-based modelling efforts by employing a simplified version of Kappa calculus to compact a set of domain-specific models coming from biophysics of photosynthesis. We do not construct the models from the scratch but we rather take several existing kinetic models of photosynthesis-related processes and reformulate them in the algebraic framework. At the level of compositional representation, we formulate syntactic reductions of the models that preserve behavioural equivalence.

On the practical side, we employ BioNetGen language (BNGL) and related tools [9,19] to implement the models. We contribute by automatising our reduction procedure in Python. By applying reductions to the considered models we show the power of process-algebraic framework to compactly represent combinatorially exploding systems of light reactions.

The paper shows the importance of process-algebraic description for the domain of photosynthesis. To the best of our knowledge, the application of such techniques in the field of systems biology of photosynthesis is still at its beginning and we believe our contribution is useful to help to establish rule-based modelling in the domain.

### 1.1 Related Work

There are many applications of rule-based modelling available (see [4] for an overview). However, applications to photosynthesis are very rare. In [21], the authors provide a model of chlorophyll *a* fluorescence induction kinetics that is simulated in the rule-based framework by means of Monte Carlo simulation. The work is unique in the sense it pioneers rule-based approach for photosynthesis models. The contri-

bution brings a stochastic simulation algorithm that reflects the differing context where the context-free rules appear. This is important, since the quantitative rates of electron transfers are modulated by aggregation of several modifications of the protein complex, i.e., the photosystem. The considered model is comparable to Lazar model [15] we use for our case study.

Automatic reduction of rule-based model is available in the tool `complx`. It is able to automatically perform so-called *decontextualization* transformation on Kappa models [7]. This automatic reduction works differently than our method and yields different results for the photosynthetic models.

In [3], the problem of combinatorial complexity in models where the quantitative semantics cannot be generated is addressed. Several frameworks for abstracting the models at the level of semantics have been developed for Kappa.

## 2 Background

We define simplified Kappa using a process-like notation as is presented in [7], syntax and the notions of structural equivalence and matching are entirely take from [7]:

expression	$E ::= \emptyset \mid a, E$	site	$s ::= n_i^\lambda$
agent	$a ::= N(\sigma)$	site name	$n ::= x \in \mathcal{S}$
agent name	$N ::= A \in \mathcal{A}$	internal state	$\iota ::= \epsilon \mid m \in \mathbb{V}$
interface	$\sigma ::= \emptyset \mid s, \sigma$	binding state	$\lambda ::= \epsilon \mid i \in \mathbb{N}$

where  $\mathcal{A}$  is a finite set of agent names,  $\mathcal{S}$  is a finite set of site names,  $\mathbb{V}$  is a finite set of values representing modified states of the sites. An agent is denoted by its name and its interface. Interface consists of a sequence of sites.  $x_i^\lambda$  denotes a site  $x$  with internal state  $\iota$  and binding state  $\lambda$ . If the binding state is  $\epsilon$  then the site is free, otherwise it is bound. By convention, when a binding or internal site is not specified,  $\epsilon$  is considered.

Note that full Kappa is richer. It allows a binding state meaning a *free or bound site*, denoted by a question mark. We also omit rates from the rules.

**Definition 2.1** *An expression is well-formed if a site name occurs only once in an interface and if each binding state ( $\neq \epsilon$ ) present in the expression occurs exactly twice. The set of all well-formed expressions is denoted as  $\mathcal{E}$ . The set of all well-formed expressions that can be generated from the literal  $a$  is called the set of all well-formed agents and is denoted as  $\mathcal{E}_a$ . Similarly,  $\mathcal{E}_\sigma$  denotes the set of all well-formed interfaces and  $\mathcal{E}_s$  the set of all well-formed sites.*

Next, we define some notations we use throughout the text.

**Notation 2.2** expressions  $E, E' \in \mathcal{E}$ ; agents  $a, a' \in \mathcal{E}_a$ ; agent name  $A \in \mathcal{A}$ ; interfaces  $\sigma, \sigma' \in \mathcal{E}_\sigma$ ; sites  $s, s' \in \mathcal{E}_s$ ; site name  $x \in \mathcal{S}$ ; internal state  $\iota \in \{\epsilon\} \cup \mathbb{V}$ ; specific internal state  $m \in \mathbb{V}$ ; binding state  $\lambda \in \{\epsilon\} \cup \mathbb{N}$ ;

Next, we provide inductive definitions of some useful mappings.

**Definition 2.3** Agent name is a mapping name :  $\mathcal{E}_a \rightarrow \mathcal{A}$  defined as  $\text{name}(A(\sigma)) = A$ .

We define agent sites as a mapping sites :  $\mathcal{E}_a \rightarrow 2^{\mathcal{S}}$  such that  $\text{sites}(A()) = \emptyset$ ,  $\text{sites}(A(x_i^\lambda)) = \{x\}$ , and  $\text{sites}(A(s, \sigma)) = \text{sites}(A(s)) \cup \text{sites}(A(\sigma))$ .

Agent internal state is a mapping state :  $\mathcal{E}_a \rightarrow (\mathbb{V} \cup \{\epsilon\})^{\mathcal{S}}$  defined as  $\text{state}(A()) = \emptyset$ ,  $\text{state}(A(x_i^\lambda)) = \{(x, \iota)\}$ , and  $\text{state}(A(s, \sigma)) = \text{state}(A(s)) \cup \text{state}(A(\sigma))$ .

**Definition 2.4** Structural equivalence  $\equiv \subseteq \mathcal{E} \times \mathcal{E}$  is defined as a relation satisfying the following properties:

- (i) Reflexivity:  $E \equiv E$
- (ii) The order of sites in interfaces does not matter:  
 $E, A(\sigma, s, s', \sigma'), E' \equiv E, A(\sigma, s', s, \sigma'), E'$
- (iii) The order of agents in an expression does not matter:  
 $E, a, a', E' \equiv E, a', a, E'$
- (iv) Binding states can be injectively renamed:  
 $E[i/j] \equiv E$   
 where  $i, j \in \mathbb{N}$  and  $i$  does not occur in  $E$ .

Solution  $[E] \in 2^{\mathcal{E}}$  denotes the equivalence class of  $E$  in  $\equiv$ .  $\mathcal{L}$  is a set of all solutions.

**Definition 2.5** A rule is a pair of expressions  $E_l, E_r$  (usually written as  $E_l \rightarrow E_r$ ). The set of all rules is denoted as  $\mathcal{R}$ .

The left hand side  $E_l$  of the rule describes the solution taking part in the reaction and the right hand side  $E_r$  describes the effects of the rule. The rule can be either a binding rule or a modification rule. A binding (unbinding) rule binds two free sites together (or unbinds two bound sites). A modification rule modifies some internal state [7].

**Definition 2.6** Matching is a relation denoted as  $\models \subseteq \mathcal{E} \times \mathcal{E}$  and defined inductively in the left column below. Replacement is a function  $\mathcal{E} \times \mathcal{E} \rightarrow \mathcal{E}$  defined in the right column.

$$\begin{array}{ll}
 n_i^\lambda \models n_i^\lambda & n_i^\lambda[n_{i_r}^{\lambda_r}] = n_{i_r}^{\lambda_r} \\
 n_i^\lambda \models n_i^\lambda & n_i^\lambda[n^{\lambda_r}] = n_i^{\lambda_r} \\
 \sigma \models \emptyset & \sigma[\emptyset] = \sigma \\
 \frac{s \models s_l \quad \sigma \models \sigma_l}{s, \sigma \models s_l, \sigma_l} & s, \sigma[s_r, \sigma_r] = s[s_r], \sigma[\sigma_r] \\
 \frac{\sigma \models \sigma_l}{N(\sigma) \models N(\sigma_l)} & N(\sigma)[N(\sigma_r)] = N(\sigma[\sigma_r]) \\
 E \models \emptyset & E[\emptyset] = E \\
 \frac{a \models a_l \quad E \models E_l}{a, E \models a_l, E_l} & (a, E)[a_r, E_r] = a[a_r], E[E_r]
 \end{array}$$

A replacement can be applied only if the corresponding matching is satisfied.

In order to apply a rule  $E_l \rightarrow E_r$  to a solution  $[E]$  the expression  $E$  representing

the solution must first be reordered to an equivalent expression  $E'$  that matches  $E_l$  (according to the definition of matching stated above).  $E'$  is then replaced with  $E'[E_r]$  (also defined above).

**Definition 2.7** Rule application is a mapping  $\tau : \mathcal{L} \times \mathcal{R} \rightarrow \mathcal{L}$  such that  $\tau([E], (E_l, E_r)) = [E'[E_r]]$  whenever  $\exists E' \in [E]. E' \models E_l$ .

Rules yield a *transition system* between solutions containing an edge  $[E] \rightarrow_{E_l, E_r} [E'[E_r]]$  whenever  $\exists E' \in [E]. E' \models E_l$ .

**Definition 2.8** An agent signature  $(\Sigma, I)$  is a pair of mappings  $\Sigma : \mathcal{A} \rightarrow 2^{\mathcal{S}}$  and  $I : \mathcal{A} \times \mathcal{S} \rightarrow 2^{\mathcal{V}}$ .

Informally,  $\Sigma$  restricts for each agent name  $A \in \mathcal{A}$  the set of site names that can occur in an agent with name  $A$ . And  $I$  restricts the set of internal states a particular site can attain.

**Definition 2.9**  $E$  satisfies agent signature  $(\Sigma, I)$ , denoted  $(\Sigma, I) \vdash E$ , if  $E$  satisfies one of the following conditions:

- (i)  $E \equiv \emptyset$
- (ii)  $E \equiv A()$  and  $A \in \text{dom}(\Sigma)$
- (iii)  $E \equiv A(x_c^\lambda)$  and  $x \in \Sigma(A)$
- (iv)  $E \equiv A(x_m^\lambda)$  and  $x \in \Sigma(A)$  and  $m \in I(A, x)$
- (v)  $E \equiv A(s, \sigma)$  where  $(\Sigma, I) \vdash A(s)$  and  $(\Sigma, I) \vdash A(\sigma)$
- (vi)  $E \equiv E_l, A(\sigma)$  where  $(\Sigma, I) \vdash E_l$  and  $(\Sigma, I) \vdash A(\sigma)$

If  $r = (E_l, E_r) \in \mathcal{R}$  and  $(\Sigma, I) \vdash E_l$  and  $(\Sigma, I) \vdash E_r$  then  $(\Sigma, I) \vdash r$ .

If  $R \subseteq \mathcal{R}$  and  $\forall r \in R. (\Sigma, I) \vdash r$  then  $(\Sigma, I) \vdash R$ .

**Definition 2.10** An agent  $a$  is complete with respect to signature  $(\Sigma, I)$ , denoted  $(\Sigma, I) \models a$ , if  $\text{sites}(a) = \Sigma(\text{name}(a)) \wedge \forall x \in \text{sites}(a). \text{state}(a)(x) \in I(\text{name}(a), x)$ .

An expression  $E$  is complete with respect to signature  $(\Sigma, I)$ , denoted  $(\Sigma, I) \models E$ , if it satisfies one of the following conditions:

- (i)  $E \equiv \emptyset$
- (ii)  $E \equiv a, E'$  where  $a \in \mathcal{E}_a, E' \in \mathcal{E}$  and  $(\Sigma, I) \models a$  and  $(\Sigma, I) \models E'$

$\mathcal{E}_{(\Sigma, I)} = \{E \in \mathcal{E} \mid (\Sigma, I) \models E\}$  is a set of all expressions that are complete with respect to signature  $(\Sigma, I)$ .

**Definition 2.11** A rule-based model  $\mathcal{M}$  is a tuple  $(\Sigma, I, R)$  that satisfies the condition  $(\Sigma, I) \vdash R$ . We use the notation  $\text{Signature}(\mathcal{M}) = (\Sigma, I)$ ,  $\text{Rules}(\mathcal{M}) = R$ ,  $\mathcal{M} \vdash E \iff (\Sigma, I) \vdash E$  for  $E \in \mathcal{E}$ ,  $\mathcal{M} \models E \iff (\Sigma, I) \models E$  for  $E \in \mathcal{E}$ , and  $\mathcal{E}_{\mathcal{M}} = \mathcal{E}_{(\Sigma, I)}$ .

**Definition 2.12** An initialised model  $M$  is a pair  $(\mathcal{M}, E_i)$  where  $\mathcal{M}$  is a rule-based model and  $E_i$  is an expression representing the initial solution such that  $\mathcal{M} \models E_i$ .

**Definition 2.13** A state space of an initialised model  $M = (\mathcal{M}, E_i)$  is a pair  $(\text{Solutions}(M) \subseteq \mathcal{L}, \text{Reactions}(M) \subseteq \mathcal{L} \times \mathcal{L})$  defined inductively as follows:

- (i)  $[E_i] \in \text{Solutions}(M)$
- (ii)  $[E] \in \text{Solutions}(M)$  and  $\exists r \in \text{Rules}(\mathcal{M}).\tau([E], r) = [E']$   
if and only if  $[E'] \in \text{Solutions}(M)$  and  $([E], [E']) \in \text{Reactions}(M)$

**Definition 2.14** *Initialised models  $M_1 = (\mathcal{M}_1, E_1)$  and  $M_2 = (\mathcal{M}_2, E_2)$  are structurally equivalent, denoted  $M_1 \equiv M_2$ , if and only if  $\text{Solutions}(M_1) = \text{Solutions}(M_2)$  and  $\text{Reactions}(M_1) = \text{Reactions}(M_2)$ .*

**Definition 2.15** *Models  $\mathcal{M}_1$  and  $\mathcal{M}_2$  are structurally equivalent, denoted  $\mathcal{M}_1 \equiv \mathcal{M}_2$ , if and only if  $\forall E_i \in \mathcal{E}_{\mathcal{M}_1} \cup \mathcal{E}_{\mathcal{M}_2}.(\mathcal{M}_1, E_i) \equiv (\mathcal{M}_2, E_i)$ .*

In BNGL, agents are called molecules and they are specified in a similar manner as in the simplified Kappa. An example of a molecule is  $\mathbf{A}(\mathbf{x}\sim\mathbf{n}!1)$  where the site  $\mathbf{x}$  has an internal state  $\mathbf{n}$  (separated from the site by a tilde) and a binding state  $1$  (separated by the exclamation mark). The BNGL alternatives to agent signatures are called molecule types they are defined using the notation demonstrated in the following example:  $\mathbf{A}(\mathbf{x}\sim\mathbf{n}\sim\mathbf{b}, \mathbf{y}\sim\mathbf{n}\sim\mathbf{a})$ . Here, the allowed internal states of the individual sites are separated by tildes (site  $\mathbf{x}$  can have an internal state  $\mathbf{n}$  or  $\mathbf{b}$ ). Rules are described by the *lhs -> rhs* notation (or *lhs <-> rhs* in the case of reversible rules). The individual model components (molecule types, reaction rules, seed species, observables) are in BNGL separated by the **begin keyword** and **end keyword** pairs.

### 3 Model Reductions

In this section, we formally define several syntactic operations that can be used to reduce rule-based models.

**Definition 3.1** *Model  $\mathcal{M}_1$  and model  $\mathcal{M}_2$  are in relation context enumeration elimination,  $((\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee})$ , iff  $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2) = (\Sigma, I)$  and  $\exists A \in \mathcal{A}, x \in \mathcal{S}, \lambda \in \{\epsilon\} \cup \mathbb{N}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma$  such that*

- (i)  $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^m, E_r^m) | m \in I(A, x)\}$  where  $\forall m \in I(A, x) :$   
 $E_l^m \equiv E_l, A(\sigma_l, x_m^\lambda)$  and  $E_r^m \equiv E_r, A(\sigma_r, x_m^\lambda)$ ,
- (ii)  $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E'_l, E'_r)\}$  where  $E'_l \equiv E_l, A(\sigma_l, x_\epsilon^\lambda)$  and  $E'_r \equiv E_r, A(\sigma_r, x_\epsilon^\lambda)$ .

**Theorem 3.2** *Context enumeration elimination preserves structural equivalence of models. If  $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee}$  then  $\mathcal{M}_1 \equiv \mathcal{M}_2$ .*

**Proof.** Let  $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{cee}$ . Then  $\mathcal{E}_{\mathcal{M}_1} = \mathcal{E}_{\mathcal{M}_2}$ . Let  $E_i \in \mathcal{E}_{\mathcal{M}_1}$ ,  $M_1 = (\mathcal{M}_1, E_i)$  and  $M_2 = (\mathcal{M}_2, E_i)$ . We prove that  $M_1 \equiv M_2$  by induction through the structure of their state spaces. Without a loss of generality we can fix the variables used in Definition 3.1.

- (i) From Definition 2.13:  $[E_i] \in \text{Solutions}(M_1)$ ,  $[E_i] \in \text{Solutions}(M_2)$
- (ii) *Completeness:*  
Let  $[E] \in \text{Solutions}(M_1)$  and  $r \in \text{Rules}(\mathcal{M}_1).\tau([E], r) = [E']$ .  
From induction we have  $[E] \in \text{Solutions}(M_2)$ .  
(a)  $r \in \text{Rules}(\mathcal{M}_1) \cap \text{Rules}(\mathcal{M}_2)$ . Then we have  $r \in \text{Rules}(\mathcal{M}_2)$ .

(b)  $r \in \text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2)$ . So  $\exists m \in I(A, x). r = (E_l^m, E_r^m)$ .

Let  $r' = (E_l', E_r') \in \text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1)$  and  $e \in [E]. e \models E_l^m$ . Then from Definition 2.6 we have  $e \models E_l'$  and  $e[E_r^m] = e[E_r']$ . Therefore,  $\tau([E], r') = [E']$ .

Thus  $\text{Solutions}(\mathcal{M}_1) \subseteq \text{Solutions}(\mathcal{M}_2)$ ,  $\text{Reactions}(\mathcal{M}_1) \subseteq \text{Reactions}(\mathcal{M}_2)$ .

(iii) *Soundness*:

Let  $[E] \in \text{Solutions}(\mathcal{M}_2)$  and  $r \in \text{Rules}(\mathcal{M}_2)$ .  $\tau([E], r) = [E']$ .

From induction we have  $[E] \in \text{Solutions}(\mathcal{M}_1)$ .

(a)  $r \in \text{Rules}(\mathcal{M}_2) \cap \text{Rules}(\mathcal{M}_1)$ . Then we have  $r \in \text{Rules}(\mathcal{M}_1)$ .

(b)  $r \in \text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1)$ . So  $r = (E_l', E_r')$ . Let  $e \in [E]. e \models E_l'$ . Then there must be  $x_m^\lambda$  in  $e$  that gets matched to the  $x_\epsilon^\lambda$  part  $E_l'$ . It must be that  $m \in I(A, x)$  and so  $r' = (E_l^m, E_r^m) \in \text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2)$ . From Definition 2.6 we have  $e \models E_l^m$  and  $e[E_r^m] = e[E_r']$ . Therefore,  $\tau([E], r') = [E']$ .

Thus  $\text{Solutions}(\mathcal{M}_2) \subseteq \text{Solutions}(\mathcal{M}_1)$ ,  $\text{Reactions}(\mathcal{M}_2) \subseteq \text{Reactions}(\mathcal{M}_1)$ . □

**Definition 3.3** *Model  $\mathcal{M}_1$  is in relation generic unbound context elimination with model  $\mathcal{M}_2$ ,  $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{guce}$ , iff  $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$  and  $\exists A \in \mathcal{A}, x \in \mathcal{S}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma$  such that*

- (i)  $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^1, E_r^1)\}$  where  $E_l^1 \equiv E_l, A(\sigma_l, x_\epsilon^\epsilon)$  and  $E_r^1 \equiv E_r, A(\sigma_r, x_\epsilon^\epsilon)$ ,
- (ii)  $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E_l^2, E_r^2)\}$  where  $E_l^2 \equiv E_l, A(\sigma_l)$  and  $E_r^2 \equiv E_r, A(\sigma_r)$ .

This reduction is useful in models where the removed contexts are guaranteed not to be bound.

**Definition 3.4** *Model  $\mathcal{M}_1$  is in relation specific unbound context elimination with model  $\mathcal{M}_2$ ,  $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{suce}$ , iff  $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$  and  $\exists A \in \mathcal{A}, x \in \mathcal{S}, E_l, E_r \in \mathcal{E}, \sigma_l, \sigma_r \in \mathcal{E}_\sigma, m \in \mathbb{V}$  such that*

- (i)  $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{(E_l^1, E_r^1)\}$  where  $E_l^1 \equiv E_l, A(\sigma_l, x_m^\epsilon)$  and  $E_r^1 \equiv E_r, A(\sigma_r, x_m^\epsilon)$ ,
- (ii)  $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \{(E_l^2, E_r^2)\}$  where  $E_l^2 \equiv E_l, A(\sigma_l)$  and  $E_r^2 \equiv E_r, A(\sigma_r)$ .

This reduction can be used if the set of reachable solutions in an intialized model to which this rule can be applied is not affected by the reduction.

Sometimes it may be useful to eliminate a rule from a model. Reasons for doing so can be different. For example, one might want to see how the behaviour of a model changes after the rule is removed. Or if the rule is not reachable in an initialised model then it can be removed to reduce the size of the model description.

**Definition 3.5** *Model  $\mathcal{M}_1$  is in relation rule elimination with model  $\mathcal{M}_2$ , denoted  $(\mathcal{M}_1, \mathcal{M}_2) \in \rho_{re}$ , if and only if  $\text{Signature}(\mathcal{M}_1) = \text{Signature}(\mathcal{M}_2)$  and  $\exists r \in \mathcal{R}$  such that  $\text{Rules}(\mathcal{M}_1) \setminus \text{Rules}(\mathcal{M}_2) = \{r\}$  and  $\text{Rules}(\mathcal{M}_2) \setminus \text{Rules}(\mathcal{M}_1) = \emptyset$ .*

If a rule is not reachable in some initialised model then we can safely remove it without affecting the semantics of the initialised model.

## 4 Application to Photosynthesis Models

In this section, we describe the application of the reductions to several models of photosynthesis. If the model is reaction-based, we first rewrite it to rule-based form.

### 4.1 Implementation

We used the library PySB [16] for the specification of the photosynthetic models and to automatise their export to BNGL. The scripts that implement syntactic operations have been written in Python. We used BioNetGen for constructing and simulating the models. The scripts are available at <https://github.com/jniznan/rbm-photosynthesis>.

We search the space of possible models that can be constructed by applying syntactic operations to the original model by depth-first search. We stop when we find a model that cannot be further reduced. There can be multiple models that cannot be further reduced. Our algorithm finds only one. We apply the syntactic operations in a given order: (i) context enumeration elimination, (ii) generic/specific context elimination, (iii) rule elimination. This approach is a heuristic that attempts to maximize the number of reductions.

### 4.2 Photosynthesis

Light-dependent reactions begin in photosystem II where the photons hit and excite the antenna molecules. The excitation then travels via a chain of proteins until it arrives to chlorophyll *a*. Or a photon can directly excite chlorophyll *a*. This excitation causes the primary electron acceptor (pheophytin) to accept an electron from chlorophyll *a* species called P680 – the primary electron donor. The electron is exchanged by multiple protein molecules until it reaches plastoquinone. The electron missing from chlorophyll *a* is replenished through a tyrosine residue from so called oxygen-evolving complex that strips electrons from water molecules, producing molecular oxygen and hydrogen protons into the lumen.

After plastoquinone accepts two electrons, it is converted to its PQH<sub>2</sub> form by accepting two hydrogen protons from the chloroplast stroma. Then it travels to cytochrome b<sub>6</sub>f where it is converted back to its original PQ form, leaving the two electrons in the cytochrome and the two hydrogen protons in the lumen. After that, plastoquinone returns to photosystem II, ready to accept other electrons. As is shown in Figure 1, the electrons travel through plastocyanin, photosystem I, ferredoxin to ferredoxin-NADP reductase where they are used for converting NADP<sup>+</sup> to NADPH. The process of the electrons travelling from the oxygen-evolving complex to the ferredoxin-NADP reductase is known as the *Z-scheme* of light. Hydrogen protons that are left in the lumen are pumped back into the chloroplast stroma by powering ATP synthase which uses that energy to convert ADP into ATP.



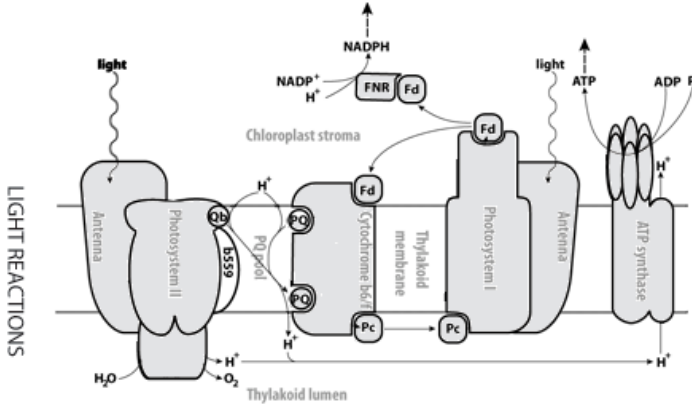


Fig. 1. Schema of the thylakoid membrane where light-dependent reactions occur.

### 4.3 Comprehensive Model of Photosystem II

Photosynthesis is well-adapted for spatially and temporally heterogeneous environments [17]. The light conditions under which photosynthesis occurs are constantly changing for many reasons, including day-night cycle, moving clouds or changing seasons. Mechanisms lying behind the adaptability of photosynthesis are not yet fully understood [18]. Evidence confirms that photosynthesis in fluctuating light is more dynamic than simply adapting to the light extremes.

Rules are used to informally capture the model, but no existing formal rule-based language is employed. This fact makes the CMS a perfect candidate for rewriting it into an RBM format. Since the reaction rates are not available the model is worked out at a qualitative level of view.

The original model contains 22 rules. By applying several syntactic operations we are able to reduce the model size to just 17 rules with smaller contexts.

We have also considered a reduced variant of the model that concentrates on light absorption by electron transfers inside PSII. That allowed the authors [10] to introduce a fully specified kinetic model. By employing the reductions we were able to compact the model rules but not to decrease their number. See [2] for details of the model and its reduction.

### 4.4 Integrated Model of Light-reactions

We consider a reaction-based model by Lazár [14] that encompasses not only photosystem II but also the other parts of thylakoid membrane participating in the *Z-scheme* of light. The model contains the following complexes and their parts: photosystem II (with parts P680, Qa, Qb); oxygen evolving complex (with states  $S_i$  where  $i = 0, 1, 2, 3$ ); PQ,PQH - plastoquinone; cytochrome  $b_6f$  (with parts bL, bHc, f); Pc - plastocyanin; photosystem I (with parts P700, Fb); Fd - ferredoxin; ferredoxin-NADP reductase.

The following molecule types are considered:

```
begin molecule types
  PSII (P680~n~p, Qa~n~m, Qb~n~m~2m)
  PQ()
```

```

PQH()
S(x~0~1~2~3)
CytB6F(bL~n~m,bHc~n~m~2m,f~n~m)
Fd(x~n~m)
Pc(x~n~p)
PSI(P700~n~p,Fb~n~m)
FNR(x~i~a~am~a2m)
end molecule types

```

The model has many rules that are perfect candidates for reduction using *context enumeration elimination*, such as the following three *charge separation* rules:

```

PSII(P680~n,Qa~n,Qb~n) -> PSII(P680~p,Qa~m,Qb~n)
PSII(P680~n,Qa~n,Qb~m) -> PSII(P680~p,Qa~m,Qb~m)
PSII(P680~n,Qa~n,Qb~2m) -> PSII(P680~p,Qa~m,Qb~2m)

```

These rules can be reduced to the following single *charge separation* rule:

```

PSII(P680~n,Qa~n,Qb) -> PSII(P680~p,Qa~m,Qb)

```

After applying all *context enumeration eliminations* we are able to reduce the model size from 69 rules down to just 22 rules. Since the model is not using any binding sites, we can automatically apply *generic unbound context eliminations* to further reduce the model. The rule stated above is reduced to the following form:

```

PSII(P680~n,Qa~n) -> PSII(P680~p,Qa~m)

```

We managed to reduce the model significantly. The original ideas of the author are much more obvious in this reduced model. The reduced model is much easier to modify and extend than its original version. We have also elaborated on several variants of PSII part of the model and demonstrated the compositionality of rule-based approach. The details are presented in [2].

## 5 Conclusions

We have demonstrated the unsuitability of traditional reaction-based modelling approaches for modelling complex biochemical processes, such as photosynthesis. We explored existing models of photosynthesis and described the simplifications that were made in those models in order to battle the problem of combinatorial explosion. We showed how these simplifications are undesirable. Rule-based modelling allows us to compactly model the processes of photosynthesis in their full mechanistic complexity without the need for such simplifying assumptions.

We set on to naively reformulate selected representative models of photosynthesis as rule-based models. These reformulated models were unnecessarily large, not exploiting the advantages of the rule-based format. Therefore, we formally defined several intuitive syntactic operations that can be used to reduce the size of these models. We provided a case study where we implemented these operations so they can be performed automatically and we managed to achieve large reductions in the size of the models. The order in which we applied the reductions turned out to be satisfactory.

We believe that in the future, the communities of biologists who are modelling photosynthesis consider the use of rule-based modelling. Rule-based modelling brings in many advantages and eliminates the reason of some artificial model simplifications.

## References

- [1] Brennan, L. and P. Owende, *Biofuels from microalgae: review of technologies for production, processing, and extractions of biofuels and co-products*, *Renewable and Sustainable Energy Reviews* **14** (2010), pp. 557 – 577.
- [2] Brim, L., J. Nižnan and D. Šafránek, *Compact representation of photosynthesis dynamics by rule-based models (full version)*, CoRR **arXiv:1410.3632** (2014).
- [3] Camporesi, F., J. Feret, H. Koepl and T. Petrov, *Combining model reductions*, *Electronic Notes in Theoretical Computer Science* **265** (2010), pp. 73–96.
- [4] Chylek, L. A., L. A. Harris, C.-S. Tung, J. R. Faeder, C. F. Lopez and W. S. Hlavacek, *Rule-based modeling: a computational approach for studying biomolecular site dynamics in cell signaling systems*, *Wiley Interdisciplinary Reviews: Systems Biology and Medicine* **6** (2014), pp. 13–36.
- [5] Chylek, L. A., E. C. Stites, R. G. Posner and W. S. Hlavacek, *Innovations of the rule-based modeling approach*, in: A. Prokop and B. Csuks, editors, *Systems Biology*, Springer Netherlands, 2013 pp. 273–300.
- [6] Danos, V., *Agile modelling of cellular signalling (invited paper)*, *Electron. Notes Theor. Comput. Sci.* **229** (2009), pp. 3–10.
- [7] Danos, V., J. Feret, W. Fontana and J. Krivine, *Abstract interpretation of cellular signalling networks*, in: F. Logozzo, D. Peled and L. Zuck, editors, *Verification, Model Checking, and Abstract Interpretation*, Lecture Notes in Computer Science **4905**, Springer Berlin Heidelberg, 2008 pp. 83–97.
- [8] Danos, V. and C. Laneve, *Formal molecular biology*, *Theoretical Computer Science* **325** (2004), pp. 69–110.
- [9] Hlavacek, W. S., J. R. Faeder, M. L. Blinov, R. G. Posner, M. Hucka and W. Fontana, *Rules for modeling signal-transduction systems*, *Science Signaling* **2006** (2006), p. re6.
- [10] Holzwarth, A., M. Müller, M. Reus, M. Nowaczyk, J. Sander and M. Rögner, *Kinetics and mechanism of electron transfer in intact photosystem ii and in the isolated reaction center: pheophytin is the primary electron acceptor*, *Proceedings of the National Academy of Sciences* **103** (2006), pp. 6895–6900.
- [11] Jansson, C. and T. Northen, *Calcifying cyanobacteria: the potential of biomineralization for carbon capture and storage*, *Current Opinion in Biotechnology* **21** (2010), pp. 365 – 371, energy biotechnology Environmental biotechnology.
- [12] Koepl, H., M. Hafner and V. Danos, *Rule-based modeling for protein-protein interaction networks - the cyanobacterial circadian clock as a case study*, in: *International Workshop on Computational Systems Biology 2009*, 2009.
- [13] Laisk, A., L. Nedbal and Govindjee, “Photosynthesis in silico,” Springer, 2009.
- [14] Lazár, D., *Modelling of light-induced chlorophyll a fluorescence rise (o-j-i-p transient) and changes in 820 nm-transmittance signal of photosynthesis*, *Photosynthetica* **47** (2009), pp. 483–498.
- [15] Lazár, D. and J. Jablonský, *On the approaches applied in formulation of a kinetic model of photosystem II: Different approaches lead to different simulations of the chlorophyll a fluorescence transients*, *Journal of theoretical biology* **257** (2009), pp. 260–269.
- [16] Lopez, C. F., J. L. Muhlich, J. A. Bachman and P. K. Sorger, *Programming biological models in python using pysb*, *Molecular systems biology* **9** (2013).
- [17] Nedbal, L., J. Červený, U. Rascher and H. Schmidt, *E-photosynthesis: a comprehensive modeling approach to understand chlorophyll fluorescence transients and other complex dynamic features of photosynthesis in fluctuating light*, *Photosynthesis research* **93** (2007), pp. 223–234.
- [18] Rascher, U. and L. Nedbal, *Dynamics of photosynthesis in fluctuating light*, *Current opinion in plant biology* **9** (2006), pp. 671–678.
- [19] Smith, A., W. Xu, Y. Sun, J. Faeder and G. E. Marai, *Rulebender: integrated modeling, simulation and visualization for rule-based intracellular biochemistry*, *BMC Bioinformatics* **13** (2012), p. S3.
- [20] Šafránek, D., J. Červený, M. Klement, J. Pospíšilová, L. Brim, D. Lazár and L. Nedbal, *E-photosynthesis: Web-based platform for modeling of complex photosynthetic processes*, *BioSystems* **103** (2011), pp. 115–124.
- [21] Xin, C.-P., J. Yang and X.-G. Zhu, *A model of chlorophyll a fluorescence induction kinetics with explicit description of structural constraints of individual photosystem ii units*, *Photosynthesis Research* **117** (2013), pp. 339–354.