Modelling Population Dynamics Using Grid Systems

Roberto Barbuti[1(✉)], Antonio Cerone[2(✉)], Andrea Maggiolo-Schettini[1(✉)], Paolo Milazzo[1(✉)], and Suryana Setiawan[1,2(✉)]

[1] Dipartimento di Informatica, Università di Pisa, Pisa, Italy
{barbuti,maggiolo,milazzo,setiawan}@di.unipi.it
[2] UNU-IIST — International Institute for Software Technology, United Nations University, Macau SAR, China
ceroneantonio@gmail.com, setiawan@iist.unu.edu

Abstract. A new formalism, *Grid Systems*, aimed at modelling population dynamics is presented. The formalism is inspired by concepts of Membrane Computing (P Systems) and spatiality dynamics of Cellular Automata. The semantics of Grid Systems describes how stochasticity is exploited for reaction duration as well as reaction selection. Grid Systems perform reactions in maximally parallel manner, imitating natural processes. Environmental events that change population behaviour can be defined in Grid Systems as rewrite rules.

A population model of a species of mosquitoes, *Aedes albopictus*, is presented. The model considers three types of external events: temperature change, rainfall, and desiccation. The events change the behaviour of the species directly or indirectly. Each individual in the population can move around in the ecosystem. The simulation of the model was performed by using a semantics based tool.

1 Introduction

An important component of Sustainable Development, sometimes called Ecologically Sustainable Development, is the use of strategies and measures to prevent environmental degradation [1]. A critical aspect of the use of preventive strategies and measures is the evaluation of their effectiveness as well as their global long-term impact on the ecosystem. In this respect, ecological modelling may represent an important tool to compare the effect of alternative prevention measures before they are implemented in the actual ecosystem.

Ecological modelling requires analytical tools able to deal with the representation of complex real world parameters. Among them we can cite parameters depending on environment characteristics (light, humidity, presence or absence of water, ...) and parameters depending on periodic or non-periodic events (temperature, level of water bodies,...). Thus, formalisms designed for ecological computational modelling should provide mechanisms for representing the influence of these parameters on the model outcome. Recently, some new formalisms were introduced [2,20,26,33]. Many of them were inspired by biological phenomena

A. Cerone et al. (Eds.): SEFM 2012 Satellite Events, LNCS 7991, pp. 172–189, 2014.
DOI: 10.1007/978-3-642-54338-8_14,

such as cells structure [31] and DNA chains [3]. Some were intended as new computing paradigms and others were defined to provide biologists with a modelling framework for analysing phenomena. For instance, stochastic P Systems [10] and BlenX [12,21], a modelling language inspired by π-calculus, along with its interface language LIME, have been used to model population dynamics. However, none of these works takes spatiality into account. In fact, most of the formalisms designed for describing biological systems were tailored for specific applications, thus lacking features for representing all the parameters that are crucial for modelling ecological systems.

In this paper we present a new formalism, *Grid Systems*, inspired by concepts of Membrane Computing (P Systems) [28,31]. A Grid System consists of an envelop compartment (the outer membrane) containing a grid of adjacent inner compartments (inner membranes). Each inner membrane is characterized by its position in the grid, and, as in P Systems, owns its rules and objects. Rules are applied with maximal parallelism and can be promoted or inhibited by objects in the system. The outer membrane has rules too, which represent events common to the entire Grid System. Differently from standard P Systems, the application of Grid System rules has a duration, which can be different for each rule.

Grid Systems can easily model the heterogeneity of environments. Each membrane in the grid may represent a distinct part of the environment with specific parameters and behavioural rules. Rules can move objects across membranes, thus describing the migration of resources or individuals. To illustrate these features, we use Grid Systems to model the ecology of the Asian tiger mosquito (*Aedes albopictus*). Spatiality aspect, such as the movement of individual mosquitos are described as movement among grid's cells.

The paper is organized as follows. Section 2 formally introduces Grid Systems and discusses some features such as stochasticity and spatiality. Section 3 shows how to use Grid Systems for illustrative model of mosquito. Finally, Sect. 4 reports the results of stochastic simulations performed using our model.

2 The Grid Systems

Several existing formalisms for modelling concurrent as well as biological systems embody the concepts of spatiality and stochasticity [4,9,18,20,22,23,25,30–34].

P Systems (Membrane Computing) are appropriate for ecological modelling for their ability in describing dynamics at object level as reactions: each reaction performs its action according to the principle of maximal parallelism [31]. Formal semantics of P Systems have been defined in [5–7]. Pardini extended P Systems with spatiality [28], but the introduced concept deals with objects position, which cannot easily represent environment heterogeneity. Cellular Automata have the ability of describing systems in which the dynamics can vary spatially according to local/neighbouring situations [2,27,35]. Grid Systems combine features of P-systems and Cellular Automata in order to model ecological systems. The independence of events occurring at the same time is modelled by maximal parallelism, while different zones of the environment are modelled by adjacent membranes, like cells in a Cellular Automaton. A variant of Gillespie's Stochastic Simulation Algorithm [15,16] is used to model stochasticity.

2.1 Syntax

Definition 1. *A Grid System $G(N, M, \Sigma, R, A, C^{(0)})$ is defined as follows:*

- *G is the grid name;*
- *N and M are two integers indicating that G has $N \times M$ cells, also called local membranes;*
- *Σ is the alphabet of object types;*
- *R is a set of transition rules;*
- *A is the set of associations of the rules with the membranes, i.e.*

$$A = \{(\rho, \gamma) \mid \rho \in R,\ \gamma \in \{G_{i,j} | 0 \leq i < N,\ 0 \leq j < M\} \cup \{G_E\}\};$$

  where
  - *$G_{i,j}$, with $0 \leq i < N$ and $0 \leq j < M$, denotes the cell in position (i, j);*
  - *G_E is the global membrane surrounding the cells;*
- *$C^{(0)}$ is the initial configuration of the grid.*

A graphical representation of a Grid System is illustrated in Fig. 1. The global membrane describes the environment and is represented by a dashed square. Local membranes are represented by a grid of squares. The two associations $(r_k, G_{3,1})$ and $(r_k, G_{2,3})$ in Fig. 1 indicate that r_k is applicable in local membranes $G_{3,1}$ and $G_{2,3}$.

A transition rule, or a rule for short, defines how a reaction can be performed in its associated membrane. Therefore a reaction is an instance of a rule.

Definition 2. *A transition rule $\rho : \alpha \xrightarrow{c,\pi} \beta\ [\psi \mid \chi]$ is defined as follows:*

- *ρ is the unique identifier of the rule;*
- *α is a non-empty multiset of reactants;*
- *β is a multiset of products;*
- *ψ is a multiset of promoters;*
- *χ is a multiset of inhibitors;*
- *$c \in \mathbb{R}^+$ is the rate with which the rule may be applied to perform a reaction.*
- *$\pi \in \{M, D\}$, with M indicating that the elapsed time of the rule is an exponential random variable with parameter c, and D indicating that the elapsed time of the rule is exactly $1/c$.*

Reactants must be in the membrane with which the rule is associated, promoters and inhibitors may be in any membrane, and products can be sent into any membrane. Note that the positions of promoters, inhibitors and products can be either absolute or relative to the current membrane.

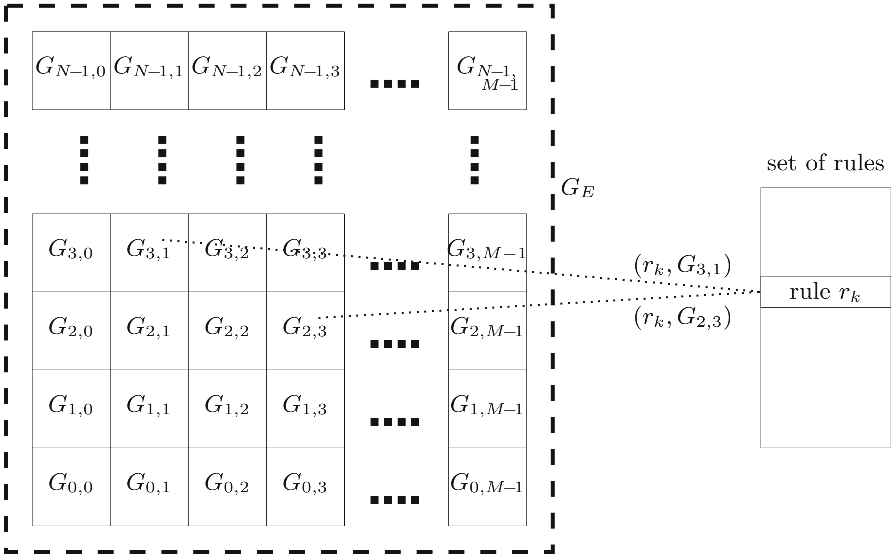

Fig. 1. Grid of membranes and association of rules with membranes

In the rest of the paper we will be using the following notations:

- $[m]_{G_A}$ refers to a membrane $m \in \{G_{i,j} | 0 \leq i < N, 0 \leq j < M\} \cup \{G_E\}$ of Grid System G_A;
- $[m]$ can be used instead of $[m]_{G_A}$ when the name of the Grid System is obvious from the context;
- $\alpha(r)$ is the multiset of reactants required by rule r;
- $\beta(r)$ is the multiset of products given by rule r;
- $\psi(r)$ is the multiset of promoters required by rule r;
- $\chi(r)$ is the multiset of inhibitors of rule r;
- $Assoc(r)$ is the set of membranes that are associated with reaction r;
- $Assoc([m])$ is the set of rules that are associated with membrane $[m]$.

The multiset of n instances of object a in membrane $G_{i,j}$ is denoted by $a^n_{i,j}$. An alternative notation for $a^n_{i,j}$ is $(a, n, G_{i,j})$. Analogously n instances of object a in membrane $[m]_{G_A}$, or $[m]$ for short, is denoted by $(a, n, [m]_{G_A})$, or by $(a, n, [m])$ for short. Finally, for a promoter, inhibitor or product a of a rule, $a_{\langle \pm v, \pm h \rangle}$ denotes that a is in $G_{i \pm v, j \pm h}$, being $G_{i,j}$ the cell in which the rule is applied; thus a can be seen as short form for $a_{\langle +0, +0 \rangle}$.

2.2 Semantics

The evolution over time of a Grid System starts from an initial configuration $C^{(0)}$ to be defined by the modeller. The system evolves passing through a trajectory of configurations $C^{(0)}$, $C^{(t_1)}$, $C^{(t_2)}$, ..., where, $0 \leq t_1 \leq t_2$.... An evolution step at time t_k consists in a maximally parallel application of rules to available objects in all membranes of the system. In order to apply a rule, reactants have to be present among available objects. Application of rules with maximal parallelism means that several rules can be applied (also more than once) to different objects in the same evolution step until no further rule is applicable. Because rules have a duration, reactants of the applied rules become committed objects of the configuration, and they remain committed until the completion of the rules. For each applied rule a termination time is computed and a new entry in the list of ongoing reactions $\Omega^{(t_k)}$ is created.

Once all applied rules of the current evolution step have been handled, the set of rules to be completed first, $\{(r_i, t, [m_i])$ such that t is minimal in $\Omega^{(t_k)}\}$, is extracted from $\Omega^{(t_k)}$. Rules in this set are the next ongoing reaction to be completed in membrane $[m]$, and t is the time of the next evolution step, namely $t_{k+1} = t$. Completion of reactions consists in the removal of its reactants from the committed objects and in the addition of the products to the available objects. Note that if there exist in $\Omega^{(t_k)}$ several ongoing reactions associated with the same time t all of these will be extracted and handled. Subsequently, the procedure is repeated to perform the next evolution step.

Definition 3. *A Configuration $C^{(t)}$ is a pair*

$$(\{C^{(t)}_{[m]} |\ m \in \{G_{i,j} | 0 \leq i < N, 0 \leq j < M\} \cup \{G_E\}\}, \Omega^{(t)})$$

where

- $C^{(t)}_{[m]}$ *is the multiset over objects that exist inside membrane $[m]$ at time t;*
- $\Omega^{(t)} = \{(r_k, t_k, [m_k]) | k = 1, ..., n$ *and* $t \leq t_1 \leq ... \leq t_n\}$ *is the set of ongoing reactions where each $(r_k, t_k, [m_k])$, $k = 1, ..., n$, denotes ongoing reactions that instantiate rule r_k in membrane $[m_k]$ and will terminate at time t_k.*

Moreover,

- $Committed^{(t)}_{[m]}$ *denotes objects of $C^{(t)}_{[m]}$ that are committed to any of the $(r_k, t_k, [m_k]) \in \Omega^{(t)}$ such that $m = m_k$;*
- $Avail^{(t)}_{[m]} = C^{(t)}_{[m]} \setminus Committed^{(t)}_{[m]}$.

Given a transition rule r and a membrane $[m]$ of a configuration $C^{(t)}$, let $applicable(r, [m])$ be a predicate that holds if and only if r is applicable in membrane $[m]$. Formally:

$$\begin{aligned} applicable(r, [m]) = \ & \alpha(r) \subseteq Avail^{(t)}_{[m]} \\ & \wedge\ \forall (a, n, [p]) \in \psi(r). a^n \subseteq (Avail^{(t)}_{[p]} \cup Committed^{(t)}_{[p]}) \\ & \wedge\ \forall (a, n, [q]) \in \chi(r). a^n \not\subseteq (Avail^{(t)}_{[q]} \cup Committed^{(t)}_{[q]}) \end{aligned}$$

Note that both available and committed objects can provide promoters and inhibitors.

2.2.1 Evolution Algorithm

The evolution of a Grid System starts from a given initial configuration $C^{(0)}$ in which $\Omega^{(0)}$ is assumed to be an empty list.

Given a configuration $C^{(t_k)}$ at time t_k, the evolution step giving $C^{(t_{k+1})}$ as result is obtained by performing the following steps (where + and − denote multiset union and subtraction, respectively):

- Step-1: For each membrane $[m]$, compute a maximal multiset $Cand([m])$ of candidate reactions (transition rule instances) over $Assoc([m])$. Each reaction in $Cand([m])$ must be an instance of an applicable rule, namely $Cand([m]) \subseteq \{r \mid applicable(r, [m])\}$. Maximality is defined as follows: let $\alpha(Cand[m]) = \bigcup_{r \in Cand([m])} \alpha(r)$, it must hold:

$$\forall r \in Assoc([m]). \quad \big(applicable(r, [m]) \implies \alpha(r) \not\subseteq \big(Avail_{[m]} - \alpha(Cand([m]))\big)\big)$$

  Namely, it must be impossible to apply any rule to the objects to which no candidate reaction is applied.
- Step-2: For each membrane $[m]$, perform each reaction $r \in Cand([m])$ by changing the states of its reactants from available to committed. Compute the duration of the reaction $elapsed(r)$ as will be described later in Subsect. 2.2.2. Insert into $\Omega^{(t_k)}$ the new tuple $(r, term(r), [m])$, where $term(r) = t_k + elapsed(r)$. Formally, for each $r \in Cand([m])$ perform the following assignments:

$$\begin{aligned} Avail_{[m]} &:= Avail_{[m]} - \alpha(r) \\ Committed_{[m]} &:= Committed_{[m]} + \alpha(r) \\ term(r) &:= t_k + elapsed(r) \\ \Omega^{(t_k)} &:= \text{insert } (r, term(r), [m]) \text{ into } \Omega^{(t_k)} \end{aligned}$$

- Step-3: Compute the multiset $First(\Omega^{(t_k)})$ of earliest reactions to terminate in $\Omega^{(t_k)}$. This set is formally defined as follows:

$$\begin{aligned} First(\Omega^{(t_k)}) = \{(r_i, t, [m_i]) \mid (r_i, t, [m_i]) \in \Omega^{(t_k)} \text{ and} \\ \forall (r', t', [m']) \in \Omega^{(t_k)}.t \le t'\} \end{aligned}$$

- Step-4: For each reaction $(r_i, t, [m_i]) \in First(\Omega^{(t_k)})$, remove its reactants from the committed objects, add its products to the available objects (by taking into account the target membrane of each product) and remove the corresponding entry from $\Omega^{(t_k)}$. Formally, for each $(r_i, t, [m_i]) \in First(\Omega^{(t_k)})$ perform the following assignments, where $[m]$ is the membrane associated with r:

$$\begin{aligned} Committed_{[m]} &:= Committed_{[m]} - \alpha(r) \\ \forall (a, n, [p]) \in \beta(r).Avail_{[p]} &:= Avail_{[p]} + a^n \\ \Omega^{(t_k)} &:= \text{remove } (r, t, [m]) \text{ from } \Omega^{(t_k)} \end{aligned}$$

– Step-5: Let t_{k+1} be the termination time t associated with reactions in $First(\Omega^{(t_k)})$ in Step-3. The configuration obtained after the assignments in the previous steps has to be considered as configuration $C^{t_{k+1}}$ to be used in the next evolution step.

If $\Omega^{(t_{k+1})}$ is empty and for each membrane $[m]$ no rule is applicable in $C^{(t_{k+1})}$ (namely $\forall[m].Cand[m] = \emptyset$) then the evolution terminates with $C^{t_{k+1}}$ as final configuration.

We remark that in Step-2 and in Step-4 of the algorithm the state of a reactant is changed from being available to being committed then to being removed. During its committed state, a reactant cannot be involved in other reactions as a reactant, but it can still affect any reaction as promoter or inhibitor.

2.2.2 Stochasticity and Reaction Duration

The Grid Systems enable stochasticity in two aspects: resolving non-deterministic choices and computing reaction elapsed times.

Nondeterministic choices may arise when some possible reactions from different rules are conflicting/competing to consume the same available objects as their reactants. In this case, reaction propensities are computed as in Gillespie's Stochastic Simulation Algorithm (SSA) as follows: for each reaction r, assume $a_r(t) = h_r(t)c_r$, where, c_r is the rate of the rule r and $h_r(t)$ is the number of different combination of reactants that can be taken from available objects at the time t. The reactions to be applied are then randomly chosen with a probability proportional to their propensities.

When a reaction r is selected to be included in $Cand([m])$, its duration $elapsed(r)$ has to be computed in Step-2 of the algorithm. The way in which $elapsed(r)$ is computed depends on the parameter π of the transition rule:

– For fixed elapsed time ('D' in the rule notation), $elapsed(r) = 1/c$, where c is the rule's rate.
– For exponentially distributed time ('M' in the rule notation), $elapsed(r)$ is an exponential random number with rate c.

This exponential random number can be generated be using the inverse method, i.e., $elapsed(r) = -1/c \log(X)$, where X is a uniform random number from the interval [0, 1).

2.2.3 Discrete Time Approximation

In a Grid System consisting mostly of rules with exponentially distributed duration there is a very small probability of having more than one reaction in $First(\Omega^{(t_k)})$. Stochasticity will cause reactions to terminate at different times, and consequently at each evolution step a single terminating reaction is handled.

In this situation the Evolution Algorithm of Grid Systems from the efficiency viewpoint behaves similarly to Gillespie's SSA (in which one reaction per step is handled). Several approximated variants of Gillespie's SSA have been proposed to improve performances [14,17,24,29]. However, such approximations are not applicable to Grid Systems because of the differences in rule selection of the latter due to maximal parallelism.

For the sake of improving simulation performances we consider a simple approximation approach for Grid Systems that consists in discretizing the simulation time. In particular, we assume time to consist of a sequence of time intervals of a fixed duration. All reactions terminating in the same time interval are then considered as terminating at the same time. This causes a simulation error proportional to the duration of the time interval.

Given an interval length δ_t, the discrete time approximation can be included in the Evolution Algorithm by replacing the definition of $First(\Omega^{(t_k)})$ in Step-3 with the following new definition:

$$First(\Omega^{(t_k)}) = \{(r,t,[m]) \mid (r,t,[m]) \in \Omega^{(t_k)} \text{ and } \forall (r',t',[m']) \in \Omega^{(t_k)}.(t \le t' \text{ or } SameTime(t,t'))\}$$

where $SameTime(t,t')$ is $true$ when $\exists k \in I^+$, such that $k\delta_t \le t, t' < (k+1)\delta_t$.

Moreover, the assignment to $\Omega^{(t_k)}$ in Step-4 of the evolution algorithm has to be modified as follows:

$$\Omega^{(t_k)} := \text{remove } (r,t',[m]) \text{ from } \Omega^{(t_k)} \text{ s.t. } SameTime(t,t') = true.$$

The interval length should be defined as a trade off between simulation performance and its precision.

2.3 Additional Notation

To ease the writing of the model by using the formalism of Grid Systems some additional notation is introduced:

Region. A group of local membranes may be identified as a region, and rules can be associated to regions. Therefore, for region R, membrane $[m]$ and rule r:
if $[m] \in R$, and $r \in Assoc(R)$, then $r \in Assoc([m])$.

Constant. A constant is an identifier that can be associated with a value (either number or string). Constants provide values to variables used within rule templates.

Rule Template. Rules with similar structures may be identified as a single template containing one or more template variables $\langle X \rangle$, each representing a finite set of values. Rules can be obtained by instantiating each $\langle X \rangle$ in the body of the template with one of its possible values.

3 Modelling: *Aedes albopictus*

3.1 Population Behaviour

Species may be potential vectors for diseases or they may be considered as endangered. Both situations are likely to have a negative impact on the ecosystem to which these species belong. It is therefore important to develop strategies to control disease vectors and preserve endangered species. Modelling population dynamics provides a way to test and compare alternative strategies and support the process of defining and implementing policies in population control and reintroduction biology.

Models of population dynamics range from differential equation to general stochastic models [11]. Recently, Bearded Vulture population was modelled using P Systems [10] where a probability measure was attached to each rule in a Markov Chain model. *Aedes albopictus* population was modelled using Stochastic CLS [8].

Some species might be endemic in one wide area due to specific local conditions that affect population behaviour spatially. Spatiality based on Cellular Automata was used for computing population coverage on a specific area [27].

Aedes albopictus, also known as Asian tiger mosquito, is a mosquito species that originated from Asia [13,19]. This species is an important subject of study since it is well-known as vector of some deadly pathogens, such as the West Nile virus, Yellow fever virus, St. Louis encephalitis, Dengue fever and Chikungunya fever. *Aedes albopictus* life-cycle has 4 phases: egg, larval, pupa, and adult [19].

Our model consists of a simple mosquito life cycle: an egg becomes an immature (first larva, then pupa), which becomes an adult. The ecosystem affects the population in terms of volume of water containers and temperature. Temperature affects the population behaviour directly by changing reaction rates. Higher temperatures will increase the rates of the transition in the life cycle. Temperature affects the population indirectly by changing desiccation rate of water in the containers. Temperature in the environment may increase or decrease through temperature change events. Water volume in the environment may increase or decrease through rainfall events.

3.2 Formal Model

We consider a 5×5 *grid* to model the space where the mosquito population lives. However, the actual movement area is 3×3. Additional rows (top and bottom) and columns (leftmost and rightmost) are set for isolating the movement area. Access to these additional cells is denied to mosquito by an initial state in which such cells contain inhibitor Z. We assume that only the central cell contains water, although we do not model how the water is distributed within the cell (e.g. uniformly or in separate containers).

We use *object* E to represents eggs, I to represent immatures and A to represent adults. The environment state is defined by the temperature level and the quantity of water. Temperature is represented by the number of objects T located in the global membrane and the quantity of water is represented by the number of objects W located in central cell $G_{2,2}$.

Environmental state may change depending on events that are either continuously triggered by the environmental state itself or are scheduled to occur at specific times. For instance, desiccation is a continuous event triggered by temperature and quantity of water while rainfall and temperature change are scheduled events. Objects $Rain100$ and $Rain200$ represent light rain and heavy rain, respectively; their affect the environment by increasing the quantity of water (number of objects W). Objects $TempUp1$ and $TempUp2$ represent increases of temperature of two different scales. Objects $TempDwn1$ and $TempDwn2$ represent decreases of temperature of two different scales. They modify the number of objects T.

We consider the following *regions*.

- $MovingSpace = \{G_{1,1}, G_{1,2}, G_{1,3}, G_{2,1}, G_{2,2}, G_{2,3}, G_{3,1}, G_{3,]}, G_{3,3}\}$ consists of all cells that adult mosquitoes may access.
- $WaterContainer = \{G_{2,2}\}$ is the central cell.
- $Boundary = \{G_{0,0}, G_{0,1}, G_{0,2}, G_{0,3}, G_{0,4}, G_{1,0}, G_{1,4}, G_{2,0}, G_{2,4}, G_{3,0}, G_{3,4}, G_{4,0}, G_{4,1}, G_{4,2}, G_{4,3}, G_{4,4}\}$ consists of the boundary cells, inaccessible to mosquitoes.

In Sects. 3.2.1 and 3.2.2 we define the rules that model the dynamics of our ecosystems. Such rules are written using templates. Table 1 shows the constants that instantiate the variables used in templates.

Table 1. Constants that instantiate the variables used in templates

$\langle L\rangle$	1	2	3	4
$Prom\langle L\rangle$	λ	T_E	T_E^3	T_E^5
$Inh\langle L\rangle$	T_E	T_E^3	T_E^5	λ
$Hatchrate\langle L\rangle$	0.3	0.4	0.45	0.5
$Failrate\langle L\rangle$	0.3	0.4	0.45	0.5
$Metarate\langle L\rangle$	0.1	0.2	0.5	1.0
$Deathimrate\langle L\rangle$	0.2	0.25	0.3	0.35
$Deathadrate\langle L\rangle$	0.4	0.45	0.5	0.55
$Ovirate\langle L\rangle$	0.1	0.2	0.33	0.5
$Desicrate\langle L\rangle$	0.75	1.0	1.5	2.5

3.2.1 Population Dynamics

In this section we introduce the rule templates that define the dynamics of *Aedes albopictus* population. Dependence of rates on temperature is expressed by combining instantiations of variables $Prom\langle L\rangle$ (promoters) and $Inh\langle L\rangle$ (inhibitors), which define ranges of temperature, and instantiations of variable for rates, as shown in Table 1.

Hatched eggs become immature mosquitoes. Reaction rate depend on temperature. Egg hatching is modelled by the following rule template.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\ Hatch\langle L\rangle : E \xrightarrow{hatchrate\langle L\rangle, M} I \quad [Prom\langle L\rangle | Inh\langle L\rangle\,]$$

if $Hatch\langle L\rangle \in Assoc(WaterContainer)$.

If we consider the following instantiations of the above template for $L = 1, 2$, according to the constants in Table 1,

$$Hatch1 : E \xrightarrow{0.3,M} I \quad [\lambda|T_E\,] \qquad and \qquad Hatch2 : E \xrightarrow{0.4,M} I \quad [T_E|T_E^3\,]$$

we can observe that a temperature up to T_E corresponds to rate 0.3 for rule $Hatch1$ and a higher temperature, over T_E and below T_E^3, corresponds to higher rate 0.4 for rule $Hatch2$.

Eggs failing to hatch are modelled by the following rule template, whose death rate depends on temperature.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\ Fail\langle L\rangle : \quad E \xrightarrow{failrate\langle L\rangle,M} \lambda \quad [Prom\langle L\rangle|Inh\langle L\rangle\,]$$

if $Fail\langle L\rangle \in Assoc(WaterContainer)$.

When water level is extremely low (below a minimum threshold) eggs may die due to dehydration. Moreover, when water level is extremely high (above or equal a maximum threshold) eggs may be flooded away.

$$DryE : \quad E \xrightarrow{8.000,M} \lambda \;\; [\lambda|W^{500}\,]$$
$$FloodE : E \xrightarrow{8.000,M} \lambda \;\; [W^{1500}\,|\lambda]$$

if $DryE, FloodE \in Assoc(WaterContainer)$.

In this rule template the maximum threshold W^{500} for dehydration is modelled as an inhibitor and the minimum threshold W^{1500} for flooding is modelled as a promoter. Here rates are much higher than 0.5 and dominate the effect of $Hatch\langle L\rangle$ and $Fail\langle L\rangle$ rules above.

Immature mosquitoes may become adult with a rate that depends on temperature.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\ Meta\langle L\rangle : \quad I \xrightarrow{metarate\langle L\rangle 0,M} A \;\; [Prom\langle L\rangle|Inh\langle L\rangle\,]$$

if $Meta\langle L\rangle \in Assoc(WaterContainer)$.

Immature mosquito failing to become an adult are modelled by the following rule template, whose rate depends on temperature.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\ DeathI\langle L\rangle : \quad I \xrightarrow{deathimrate\langle L\rangle,M} \lambda \;\; [Prom\langle L\rangle|Inh\langle L\rangle\,]$$

if $DeathI\langle L\rangle \in Assoc(WaterContainer)$.

Moreover, immature mosquitoes may be either flooded away or dehydrated by extreme water conditions.

$$DryI : \quad I \xrightarrow{8.000,M} \lambda \;\; [\lambda|W^{500}\,]$$
$$FloodI : I \xrightarrow{8.000,M} \lambda \;\; [W^{1500}\,|\lambda]$$

if $DryI, FloodI \in Assoc(WaterContainer)$.

Mosquitoes may either move from one cell to one of its four adjacent cells or remain in the initial cell. Boundary cells cannot be entered.

$$\begin{aligned}
Move1 : A &\xrightarrow{0.500,M} A_{\langle -1,+0\rangle} \quad [\lambda | Z_{\langle -1,+0\rangle}\,] \\
Move2 : A &\xrightarrow{0.500,M} A_{\langle +0,-1\rangle} \quad [\lambda | Z_{\langle +0,-1\rangle}\,] \\
Move3 : A &\xrightarrow{0.500,M} A_{\langle +0,+1\rangle} \quad [\lambda | Z_{\langle +0,+1\rangle}\,] \\
Move4 : A &\xrightarrow{0.500,M} A_{\langle +1,+0\rangle} \quad [\lambda | Z_{\langle +1,+0\rangle}\,] \\
Move5 : A &\xrightarrow{0.500,M} A
\end{aligned}$$

if $Move1, Move2, Move3, Move4, Move5 \in Assoc(MovingSpace)$.

For example, $Move1$ models the movement of object A to the cell immediately to the left of the cell where the rule is applied (from $A = A_{\langle +0,+0\rangle}$ to $A_{\langle -1,+0\rangle}$), provided that the arrival cell does not contain inhibitor Z ($Z_{\langle -1,+0\rangle}$).

Adults may lay eggs only in the cell containing water. We assume that every individual lays exactly 20 eggs.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\;\; Ovi\langle L\rangle : \quad A \xrightarrow{ovirate\langle L\rangle,M} A\,E^{20} \;\; [W\;\, Prom\langle L\rangle \mid Inh\langle L\rangle\,]$$

if $Ovi\langle L\rangle \in Assoc(WaterContainer)$.

Adult death rate depends on temperature.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\;\; DeathA\langle L\rangle : \quad A \xrightarrow{deathadrate\langle L\rangle,M} \lambda\,[Prom\langle L\rangle | Inh\langle L\rangle\,]$$

if $DeathA\langle L\rangle \in Assoc(MovingSpace)$.

3.2.2 Environment Dynamics

Desiccation is a continuously occurring event that decreases the volume of water according to a desiccation factor. Such factor depends on temperature. We assume that a desiccation event decreases the quantity of water by 4 %. This is modelled in the following rule template by removing one object W out of 25.

$$\forall\langle L\rangle \in \{1,2,3,4\}.\;\; Desic\langle L\rangle : W^{25} \xrightarrow{desicrate\langle L\rangle,M} W^{24}\,[Prom\langle L\rangle | Inh\langle L\rangle\,]$$

if $Desic\langle L\rangle \in Assoc(WaterContainer)$.

Scheduled events, such as rainfalls and temperature changes, are defined by rules that consume dummy object that are in the initial environmental state and produce rain or temperature change objects. In this way, if the rule deterministically spends time t to consume the dummy object, then the rain or temperature change object is produced exactly at the scheduled time t. Once the dummy object is produced, the rule that has produced it is disabled forever. For example, a high increase of temperature ($TempUp2$) at time $\langle S\rangle$ is modelled by the following rule template

$$\forall\langle S\rangle \in \mathbb{N}.\;\; SchedTempUp2\langle S\rangle : TempUp2At\langle S\rangle \xrightarrow{1/\langle S\rangle,D} TempUp2$$

if $SchedTempUp2\langle S\rangle \in Assoc(G_E)$.

Exactly one dummy object $TempUp2At\langle S\rangle$ in the initial state enables rule $SchedTempUp2\langle S\rangle$ just once at time $\langle S\rangle$, thus producing temperature increase object $TempUp2$ at time $\langle S\rangle$.

For simplicity we assume that each rainfall lasts 1/5 of a day (4.8 h) and that water flows away without being collected in all cells apart from central cell $G_{2,2}$.

$$\forall\langle V\rangle \in \{100, 200\}.SchedRain\langle V\rangle : Rain\langle V\rangle \xrightarrow{5,D} W_{2,2}^{\langle V\rangle}$$

if $SchedRain\langle S\rangle \in Assoc(G_E)$.

The number of objects T represents four temperature thresholds as shown by the values for promoter and inhibitor variables in Table 1. We model two possible decrement or increment of temperature using the following rule templates

$$\begin{aligned}\forall\langle C\rangle \in \{1,2\}.\ & SchedTempDwn\langle C\rangle : TempDwn\langle C\rangle\, T^{\langle C\rangle} \xrightarrow{10,D} \lambda \\ & SchedTempUp\langle C\rangle : \quad TempUp\langle C\rangle \qquad \xrightarrow{10,D} T^{\langle C\rangle}\end{aligned}$$

if $SchedTempDwn\langle C\rangle, SchedTempUp\langle C\rangle \in Assoc(G_E)$.

4 Model Simulation

A running prototype for simulating the model was developed using Java. The input is a system model defined using Grid Systems; the output is the population size. Values are tabulated in text format, which can then be fed to a data visualiser or a charting tool, such as MS Excel for previewing or TikZ for embedding it inside a LaTeXdocument (as we did in this paper).

4.1 Results

Data in Figs. 2, 3, 4 were produced by a simulation with initial objects: 2000 eggs, 1000 immatures and 400 adults.

The simulation run for 190 time units (i.e. 190 days) using data for temperature and rainfalls collected in the period May–November 2009 in the province of Massa-Carrara (Tuscany, Italy). By running the simulation several times (using the same initial objects), it was observed that shapes of output curves presents only small differences. Differences become smaller for larger numbers of initial objects (e.g., 4000 eggs, 2000 immatures and 800 adults) and bigger for smaller numbers of initial objects (e.g., 50 eggs, 30 immatures and 20 adults).

Figure 5 shows the plot of sampling data collected during May–November 2009 in the province of Massa-Carrara using 11 CO_2 mosquito traps.

Spatiality representation can be appreciated by considering the number of adults in each cell. Since the spatial model is very simple this number tends to be stable over time, i.e. forming a bell shape distribution, unless extreme conditions of the environment occur. In our model the water level directly affects the number of eggs and immatures, and consequently, after some delay, the number of adults.

The simulation was performed using a PC (Pentium D 3 GHz, 1 GB RAM, Windows XP SP3), for about 32 s. By doubling each time the initial population the simulation takes respectively 47, 70, 100, 155 s.

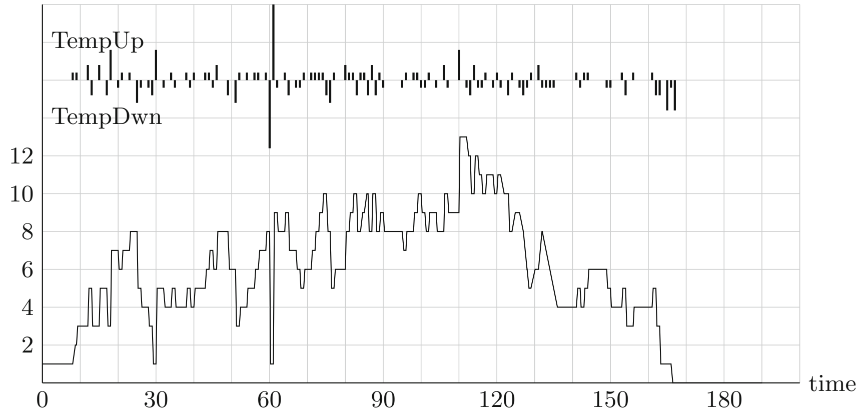

Fig. 2. Temperature level fluctuation

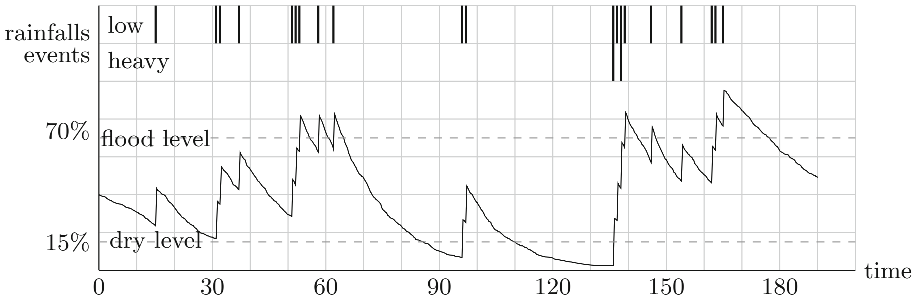

Fig. 3. Water level fluctuation due to desiccation and rainfalls

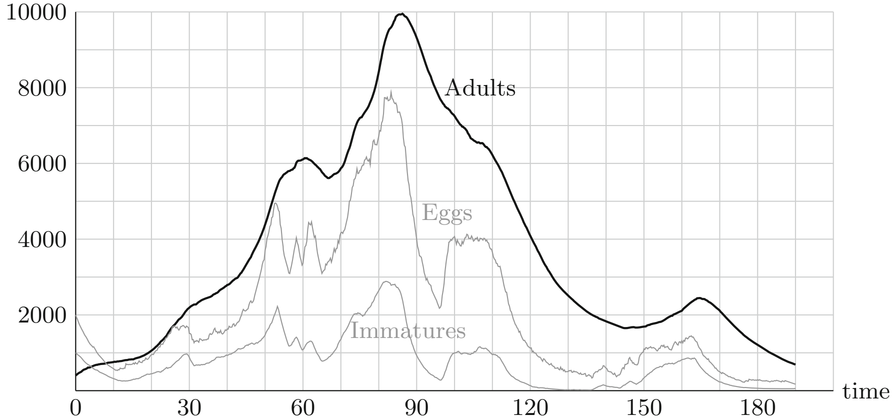

Fig. 4. Population growth: adult mosquitos, immature mosquitos (Pupae/Larvae) and their eggs

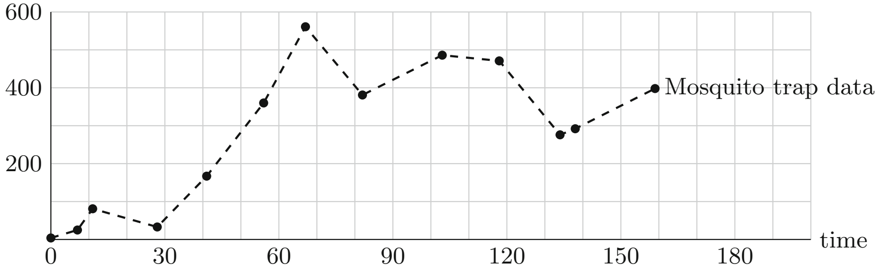

Fig. 5. Plot of CO_2 trap data: number of adults

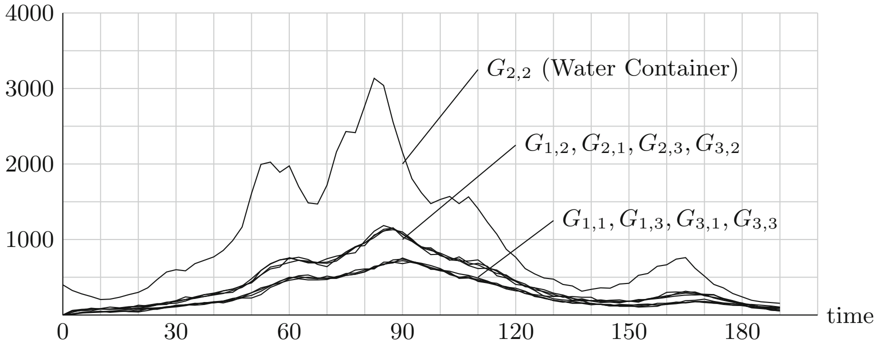

Fig. 6. Spatial distribution of adult mosquitoes over time

5 Conclusion and Future Work

Grid Systems, a new formalism for modelling population dynamics, has been presented. It supports modelling of spatiality in terms of cells in a grid. Objects may move between cells. Spatiality dynamics is obtained through transition rules. Location or time may change the applicability of rules. Grid Systems consider two aspects of stochasticity: choice and duration of rules. The enabling of rules depends on the presence/absence of promoters/inhibitors.

We have illustrated the use of Grid Systems by modelling the dynamics of an *Aedes albopictus* population. Although our model of biological aspects of the mosquito population is simpler than the model developed by Basuki *et al.* [8], the use of spatiality address a wider range of practical applications.

In our simple case study spatiality is used to model an isolated area (e.g., an island), which mosquitoes cannot enter or exit. This spatiality aspect is modelled by using *boundary cells*, which do not contain mosquitoes but, instead, contain inhibitors that prevent rules from moving mosquitoes into them. A more sophisticated use of promoters and inhibitors could support the modelling of occasional or regular flows of individuals between specific areas. For example, mosquitoes may occasionally move beyond their flight range carried inside vehicles; however, when there is a considerable traffic of vehicles between two specific areas (e.g., daily traffic between residential and commercial areas or through a bridge between two islands) the flow of mosquitoes becomes regular and facilitates the spreading of diseases associated with the vector. More generally, human movement can also contribute to the spreading of diseases carried by mosquitoes, even in absence of movement of mosquitoes beyond their flying range. Blood sucking of an infected human coming from an area where the disease is endemic by a mosquito living in an area where the disease is absent may cause the local mosquito population to become a disease vector. In this sense, our approach to spatiality dynamics supports the inclusion in the ecosystem model of aspects of the interaction between mosquito population and human population. This makes our approach suitable not only for analysis of population dynamics but also for epidemiological simulation. We intend to explore this capability of Grid Systems in our future work.

The semantics of Grid Systems has been implemented into a prototype tool written in Java to support simulation. The current version of the tool requires the model to be written in a XML tagging format to take advantages of using Javax XML Parser classes. As part of our future work we plan to develop a graphical interface to facilitate modelling and simulation control as well as display modules to provide direct visualisation through spatial animation and to produce the charts of the population growths.

Acknowledgments. This work has been supported partly by UNU-IIST and partly by Macao Science and Technology Development Fund, File No. 07/2009/A3, in the context of the EAE project. Suryana Setiawan is supported by a PhD scholarship under I-MHERE Project of the Faculty of Computer Science, University of Indonesia (IBRD Loan No. 4789-IND & IDA Credit No. 4077-IND, Ministry of Education and Culture, Republic of Indonesia).

References

1. Rio Declaration on Environment and Development. United Nations Conference on Environment and Development (UNCED), Rio de Janeiro, Brazil (1992)
2. Adamatzky, A.: Identification of Cellular Automata. Taylor and Francis, London (1994)
3. Barbuti, R., Caravagna, G., Maggiolo-Schettini, A., Milazzo, P., Pardini, G.: The calculus of looping sequences. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 387–423. Springer, Heidelberg (2008)
4. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Pardini, G., Rama, A.: A process calculus for molecular interaction maps. In: Membrane Computing and Biologically Inspired Process Calculi (MeCBIC), pp. 35–49 (2009)
5. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S.: Compositional semantics and behavioral equivalences for p systems. Theor. Comput. Sci. **395**(1), 77–100 (2008)
6. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S.: A p systems flat form preserving step-by-step behaviour. Fundam. Inform. **87**(1), 1–34 (2008)
7. Barbuti, R., Maggiolo-Schettini, A., Milazzo, P., Tini, S.: An overview on operational semantics in membrane computing. Int. J. Found. Comput. Sci. **22**(1), 119–131 (2011)

8. Basuki, T.A., Cerone, A., Barbuti, R., Maggiolo-Schettini, A., Milazzo, P.: Modelling the dynamics of an aedes albopictus population. In: Proceedings of Application of Membrane Computing, Concurrency and Agent-based Modelling in Population Biology (2010)
9. Cardelli, L., Gardner, P.: Processes in space. In: Ferreira, F., Löwe, B., Mayordomo, E. (eds.) CiE 2010. LNCS, vol. 6158, pp. 78–87. Springer, Heidelberg (2010)
10. Cardona, M., Colomer, M.A., Perez-Jimenez, M.J., Sanuy, D., Margalida, A.: A P system modeling an ecosystem related to the bearded vulture. In: Proceedings of Sixth Brainstowming Week on Membrane Computing, pp. 52–66 (2008)
11. Chaloupka, M.: Stochastic simulation modelling of southern great barrier reef green turtle population dynamics. Ecol. Model. **148**, 79–109 (2001)
12. Dematté, L., Priami, C., Romanel, A.: The BlenX language: a tutorial. In: Bernardo, M., Degano, P., Zavattaro, G. (eds.) SFM 2008. LNCS, vol. 5016, pp. 313–365. Springer, Heidelberg (2008)
13. Focks, D.A., Daniels, E., Haile, D.G., Keesling, J.E.: A simulation model of the epidemiology of urban dengue fever: literature analysis, model development, preliminary validation, and samples of simulation results. Am. J. Trop. Med. Hyg. **53**, 489–506 (1995)
14. Gibson, M.A., Bruck, J.: Efficient exact stochastic simulation of chemical systems with many species and many channels. J. Phys. Chem. **104**, 1876–1889 (2000)
15. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Comput. Phys. **22**, 403–434 (1976)
16. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**, 2340–2361 (1977)
17. Gillespie, D.T.: Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. **115**, 1716–1733 (2001)
18. Goss, P.J., Peccoud, J.: Quantitative modeling of stochastic system in molecular biology by using Petri Nets. J. Bioinform. Comput. Biol. **95**, 6750–6755 (1990)
19. Hawley, W.A.: The biology of aedes albopictus. J. Am. Mosq. Control Assoc. **4**, 1–39 (1988)
20. John, M., Ewald, R., Uhrmacher, A.M.: A spatial extension to the π-calculus. Electron. Notes Theoret. Comput. Sci. **194**, 133–148 (2009)
21. Kahramanoğullari, O., Jordan, F., Lynch, J.: CoSBiLab LIME: a language interface for stochastic dynamical modelling in ecology. Environ. Model Softw. **26**, 685–687 (2011)
22. Kohn, K.W., Aladjem, M.I., Weinstein, J.N., Pommier, Y.: Molecule interaction maps of bioregularity networks: a general rubric for systems biology. Mol. Biol. Cell **17**, 1–13 (2005)
23. Lanotte, R., Tini, S.: Probabilistic bisimulation as a congruence. ACM Trans. Comput. Log. **10**(2), 1 (2009)
24. Li, H., Petzold, L.: Logarithmic Direct Method for Discrete Stochastics Simulation of Chemically Reacting Systems. Technical report, University of California Santa Barbara (2006)
25. Milazzo, P.: Qualitative and quantitative formal modeling of biological systems. Ph.D. thesis, Università di Pisa (2007)
26. Milner, R.: Communication and mobile systems: the π-calculus. In: Proceeding of the Pacific Symposium on Biocomputing, pp. 459–470 (2001)
27. Moreno, D.H.R., Federico, P., Canziani, G.A.: Population dynamics models base on cellular automata that includes habitat quality indices defined through remote sensing. In: ISRSE RM (2001)

28. Pardini, G.: Formal modelling and simulation of biological systems with spatiality. Ph.D. thesis, Università di Pisa (2011)
29. Pineda-Krch, M.: Gillespie SSA: implementing the stochastic simulation algorithm in R. J. Stat. Softw. **25**, 1–18 (2008)
30. Priami, C., Regev, A., Silverman, W., Shapiro, E.Y.: Application of a stochastic name-passing calculus to representation and simulation a molecular processes. Inf. Process. Lett. **80**, 25–31 (2001)
31. Păun, G.: Computing with membranes. J. Comput. Syst. Sci. **61**, 108–143 (2000)
32. Păun, G.: Twenty six research topics about spiking neural P systems. In: Proceeding of Sixth Brainstowming Week on Membrane Computing (2008)
33. Regev, A., Silverman, W., Shapiro, E.Y.: Representation and simulation of biochemical processes using the π-calculus process algebra. In: Proceeding of the Pacific Symposium on Biocomputing, pp. 459–470 (2001)
34. Tini, S.: Non-expansive epsilon-bisimulations for probabilistic processes. Theor. Comput. Sci. **411**(22–24), 2202–2222 (2010)
35. Wolfram, S.: A New Kind of Science. Wolfram Media, Champaign (2002)