Synchronous Parallel Composition in a Process Calculus for Ecological Models

Mauricio Toro[1] ⋆, Anna Philippou[1], Christina Kassara[2], and Spyros Sfenthourakis[3]

[1] Department of Computer Science, University of Cyprus
{mtoro, annap}@cs.ucy.ac.cy
[2] Department of Biology, University of Patras, Greece
cristina.kassara@gmail.com
[3] Department of Biology, University of Cyprus
sfendourakis.spyros@ucy.ac.cy

Abstract. In this paper we extend PALPS, a process calculus proposed for the spatially-explicit, individual-based modeling of ecological systems, with a synchronous parallel operator. The semantics of the resulting calculus, S-PALPS, is defined at the level of populations as opposed to the level of individuals as was the case with PALPS, thus, allowing a considerable reduction in a system's state space. Furthermore, we provide a translation of the calculus into the model checker PRISM for simulation and analysis. We apply our framework to model and study the population dynamics of the Eleonora's falcon in the Mediterranean sea.

1 Introduction

Population ecology is a sub-field of ecology that studies changes in the size and age composition of populations, and the biological and environmental processes influencing those changes. Its main aim is to gain a better understanding of population dynamics and make predictions about how populations will evolve and how they will respond to specific management schemes. To achieve this goal, scientists have been constructing models of ecosystems. These models are abstract representations of the systems in question which are subsequently studied to gain understanding of the real systems.

Recently, we have witnessed an increasing trend towards the use of formal frameworks for reasoning about biological and ecological systems [19,14,6]. In our work, we are interested in the application of process algebras for studying the population dynamics of ecological systems. Process algebras provide a number of features that make them suitable for capturing these systems. In particular, they are especially suited towards the so-called individual-based approach of modeling populations, as they enable one to describe the evolution of each individual of the population as a process and, subsequently, to compose a set of individuals (as well as their environment) into a complete ecological system. Features such as time, probability and stochastic behavior, which have been

⋆ This work was carried out during the tenure by the first author of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 246016.

extensively studied within the context of process algebras, can be exploited to provide more accurate models. One key question following a model construction is what one can do with a model other than just simulate trajectories. A possible answer is to use model-checking tools for automatically analyzing properties of the model.

In our previous work we presented PALPS, a process algebra developed for modeling and reasoning about spatially-explicit individual-based systems [13]. In PALPS, individuals are modeled as processes consisting of a species and a location that may change dynamically. Individuals may engage in any of the basic processes of reproduction, dispersal, predation and death and they may communicate with other individuals residing at the same location. We have also associated PALPS with a translation to the probabilistic model checker PRISM with the prospect of making more advanced analysis of ecological models as opposed to just simulations. Our initial experiments of [13,12] using our methodology for reasoning about the population dynamics of systems delivered promising results via the use of statistical model checking provided by PRISM. However, it also revealed limitations of our approach relating to two problems.

The first problem regards reproduction and the dynamic nature of the size of a population. In particular, given a system consisting of a set of individuals, our PRISM translation of the PALPS model associated one PRISM module to each individual. Given this, when an individual produces an offspring, one would require that a new module would be created dynamically. However, PRISM does not support the dynamic creation of new modules. Thus, in our translation we resorted to placing a limit max on the maximum number of individuals that could be active at any point in time and defining max modules which oscillated between the *active* and the *inactive* state as individuals experience birth and mortality, respectively. As a result, a state of our PRISM translation, at any point in time, involved a total of max individuals irrespectively of the number of existing individuals.

The second source of inefficiency in the PALPS translation of [13] relates to the high degree of nondeterminism arising in the PALPS semantics. For instance, consider individuals P_i, $1 \leq i \leq 5$, of species $\mathbf{s}$ at some location ℓ, each executing an action a_i and then becoming Q_i. In PALPS we would write this system as $S \stackrel{\text{def}}{=} P_1{:}\langle \mathbf{s}, \ell\rangle | \dots | P_5{:}\langle \mathbf{s}, \ell\rangle$, where $P_i \stackrel{\text{def}}{=} a_i.Q_i$. Then, according to the operational semantics of PALPS, S may execute the 5 actions $a_1, \dots, a_5$ in any order. As a result, there exist 5! possible executions of these actions eventually leading to state $S' \stackrel{\text{def}}{=} Q_1{:}\langle \mathbf{s}, \ell\rangle | \dots | Q_5{:}\langle \mathbf{s}, \ell\rangle$. This phenomenon leads to a very quick explosion of the state space. To alleviate this problem, in [12], we proposed the use of policies within the PALPS framework. Policies were defined as an entity that may place a priority on the order of execution between actions. On the one hand, they enable the modeling of *process ordering* often used in ecological models, while, on the other hand they reduce the state space. In the example above, if we consider a policy that assigns increasing priorities to actions a_1 to a_5, then, there is only one possible execution to reach state S'. Note, however, that this method will give reduced or no benefits in the case where some or all of the a_i's coincide.

In this work, our goal has been to address the above-mentioned issues by proposing a new semantics of PALPS and an associated PRISM translation that disassociates the number of modules from the maximum number of individuals and, if possible, removes the restriction on the maximum number of individuals altogether, while removing as much unnecessary nondeterminism as possible. Our proposal, consists of a synchronous extension S-PALPS which features the following two key design decisions.

1. We provide a synchronous semantics of PALPS which implements the concept of *maximum parallelism*: at any given time all individuals that may execute an action will do so simultaneously. For example, system S considered above, will evolve to state S' in just one step, during which the actions $a_1, \ldots, a_5$ will be executed simultaneously. As a result, the new parallel composition construct achieves a reduction in the state space while continuing to capture coherently the behavior of population systems which are generally considered to evolve in stages (e.g. birth, dispersal, reproduction, etc) in which all of the individuals are involved.
2. We structure our calculus at the level of local populations, grouping together identical individuals located at the same location. This is achieved by the introduction of the new construct $P{:}\langle \mathbf{s}, \ell, q\rangle$ which refers to q individuals of species $\mathbf{s}$ at location ℓ.

We provide S-PALPS with an encoding to the PRISM language and we prove its correctness. In this translation, it is natural to define one module for each component of the form $P{:}\langle \mathbf{s}, \ell, q\rangle$. For instance, a system where individuals can be in one of states $P_1, \ldots, P_m$ and located in one of locations $\ell_1, \ldots, \ell_n$ would be translated in a system composed of $m \times n$ modules. As a result, the number of modules of which the model is comprised is independent of the number of existing individuals.

As an example, we study the Eleonora's Falcon (falco eleonorae) [20] in S-PALPS. Eleonora's falcon is a migrant species that breeds on Mediterranean islands and winters on islands of the Indian Ocean and along the eastern African coast. A large part of the world population concentrates on a small number of islands in the Aegean Sea. In Europe, the species is considered as rare and hence of local conservation importance because, although it is not globally threatened, its world population is below 10,000 breeding pairs and its survival in Europe is highly dependent on the breeding conditions on the islands on which it concentrates. We employ our methodology to investigate the population dynamics of the species by statistical model checking in PRISM.

Various formal frameworks have been proposed in the literature for modeling ecosystems. One strand is based, like PALPS, on process calculi such as WSCCS [19]. WSCCS is a probabilistic extension of CCS [10] with synchronous communication that has been employed in various ecological studies by the author and others [18,8]. Like PALPS, it follows the discrete-time approach to modeling but does not include the notion of space. A different approach is that of *P systems* [14], conceived as a class of distributed and parallel computing inspired by the compartmental structure and the functioning of living cells. P-systems fall in the category of rewriting systems, where a structure may evolve by the application of rewriting rules. The semantics of P-systems are closely related to S-PALPS: rules are usually applied with maximal parallelism while several proposals have been considered on resolving the nondeterminism that may arise when more than one combination of rules is possible, e.g. [11,5]. Probabilistic P systems have been applied to model the population dynamics of various ecosystems [4,3,5] as well as to study evolution problems [2]. Finally, we mention that Stochastic P systems have been translated into PRISM in [17]. However, as far as we know, there has been no work on the use of model checking for probabilistic P-Systems.

The structure of the remainder of the paper is as follows. In Section 2 we present the syntax and the semantics of S-PALPS. In Section 3 we present a translation of S-PALPS into the Markov-decision-process component of the PRISM language and we prove its correctness. We then apply our techniques to study the population dynamics of the Eleonora's falcon in Sections 4 and 5. Section 6 concludes the paper.

2 Synchronous PALPS

In this section we introduce Synchronous PALPS, S-PALPS. S-PALPS extends PALPS in two ways. Firstly, S-PALPS differs to PALPS in the treatment of the parallel composition: in the semantics of S-PALPS this is treated synchronously, in the sense that in any composition $P|Q$ the actions of P and Q are taken simultaneously. Secondly, S-PALPS offers a new construct for modeling multiplicity of individuals. Specifically, we write $P{:}\langle \mathbf{s}, \ell, q\rangle$ for q copies of individual P of species $\mathbf{s}$ and location ℓ. This construct results in a more succinct representation of systems and in conjunction with the synchronous parallel composition allows for more compact transition systems. Other changes implemented to S-PALPS in comparison to PALPS is the removal of the nondeterministic choice at the individual level, which is replaced by a conditional choice, and the inclusion of the parallel composition at the individual level which allows an explicit modeling of reproduction.

2.1 Syntax

Similarly to PALPS, in S-PALPS we consider a system as a set of individuals operating in space, each belonging to a certain species and inhabiting a location. This location may be associated with attributes which describe characteristics of the location and can be used to define location-dependent behavior of individuals. Furthermore, individuals who reside at the same location may communicate with each other upon channels, e.g. for preying, or they may migrate to a new location. S-PALPS models probabilistic events with the aid of a probabilistic operator and uses a discrete treatment of time.

The syntax of S-PALPS is based on the following basic entities: (1) **S** is a set of species ranged over by $\mathbf{s}$, $\mathbf{s}'$. (2) **Loc** is a set of locations ranged over by ℓ, ℓ'. The habitat of a system is then implemented via a relation **Nb**, where $(\ell, \ell') \in \mathbf{Nb}$ exactly when locations ℓ and ℓ' are neighbors. For convenience, we write $\mathbf{Nb}(\ell)$ for the set of all neighbors of ℓ. (3) **Ch** is a set of channels ranged over by lower-case strings. (4) Ψ is a set of attributes, ranged over by ψ, ψ'. We write ψ_ℓ for the value of attribute ψ at location ℓ. Attributes may capture characteristics of a location e.g. its capacity or its temperature.

S-PALPS employs two sets of expressions: *logical expressions*, ranged over by e, and *arithmetic expressions*, ranged over by w. They are constructed as follows

$$e ::= true \mid \neg e \mid e_1 \wedge e_2 \mid w \bowtie c$$
$$w ::= c \mid \psi@\ell^\star \mid \mathbf{s}@\ell^\star \mid \mathbf{op}_1(w) \mid \mathbf{op}_2(w_1, w_2)$$

where c is a real number, $\bowtie \in \{=, \leq, \geq\}$, $\ell^\star \in \mathbf{Loc} \cup \{\mathsf{myloc}\}$ and $\mathbf{op}_1$ and $\mathbf{op}_2$ are the usual unary and binary arithmetic operations on real numbers. Expression $\psi@\ell^\star$ denotes the value of attribute ψ at location $\ell^\star$ and expression $\mathbf{s}@\ell^\star$ denotes the number of individuals of species s at location $\ell^\star$. In the case that $\ell^\star = \mathsf{myloc}$, then the expression refers to the value of ℓ at the actual location of the individual in which the expression appears and it is instantiated to this location when the condition needs to be evaluated.

The syntax of S-PALPS is given at two levels, the individual level ranged over by P and the system level ranged over by S. Their syntax is defined via the following BNFs

$$P ::= \mathbf{0} \mid \eta.P \mid \sum_{i\in I} p_i{:}P_i \mid \gamma?\,(P;Q) \mid \mathsf{cond}\ (e \rhd P; \mathsf{else}\ \rhd Q) \mid P_1|P_2 \mid C$$

$$S ::= \mathbf{0} \mid P{:}\langle \mathrm{s}, \ell, q\rangle \mid S_1 \,\|\, S_2 \mid S\backslash L$$

where $L \subseteq \mathbf{Ch}$, I is an index set, $p_i \in (0,1]$ with $\sum_{i\in I} p_i = 1$, C ranges over a set of process constants $\mathcal{C}$, each with an associated definition of the form $C \stackrel{\mathrm{def}}{=} P$, and

$$\eta ::= a \mid \overline{a} \mid go\,\ell \mid \surd \qquad \gamma ::= a \mid \overline{a}$$

Beginning with the individual level, P can be one of the following: Process $\mathbf{0}$ represents the inactive individual, that is, an individual who has ceased to exist. Process $\eta.P$ describes the action-prefixed process which executes action η before proceeding as P. In turn, an activity η can be an input action on a channel a, written simply as a, a complementary output action on a channel a, written as $\overline{a}$, a movement action with destination ℓ, $go\,\ell$, or the time-passing action, $\surd$. Actions of the form a, and $\overline{a}$, $a \in \mathbf{Ch}$, are used to model arbitrary activities performed by an individual; for instance, eating, preying and reproduction. The tick action $\surd$ measures a tick on a global clock and is used to separate the rounds of an individual's behavior.

Process $\sum_{i\in I} p_i{:}P_i$ represents the probabilistic choice between processes P_i, $i \in I$. The process randomly selects an index $i \in I$ with probability p_i, and then evolves to process P_i. We write $p_1{:}P_1 \oplus p_2{:}P_2$ for the binary form of this operator.

Operator $\gamma?\,(P;Q)$, is an operator new to S-PALPS. Its behavior depends on the availability of a communication on a certain channel as described by γ. Specifically, if a communication is available according to γ then the flow of control proceeds according to P, if not, the process proceeds as Q. This operator is a deterministic operator as, in any scenario, the process $\gamma?\,(P;Q)$ proceeds as either P or Q but not both, depending on the availability of the complementary action of γ in the environment in which the process is running. This construct has in fact replaced the nondeterministic choice of PALPS with the intention of yielding more tractable models. We believe this construct to be sufficient and appropriate for modeling ecosystems where choices are typically resolved either probabilistically or based on some precedence relation.

The conditional process $\mathsf{cond}\ (e \rhd P; \mathsf{else}\ \rhd Q)$ represents the conditional choice between two processes: it behaves as P, if e evaluates to $\mathtt{true}$ and as Q otherwise. Note that this choice is deterministic. The parallel composition construct $|$ models the synchronous composition between processes where its precise semantics will be explained in the next section. Finally, process constants provide a mechanism for including recursion in the calculus.

Moving on to the population level, population systems are built by composing in parallel sets of located individuals. A set of q individuals of species s located at location ℓ is written as $P{:}\langle \mathrm{s}, \ell, q\rangle$. In a composition $S_1 \| S_2$ the components may proceed while synchronizing on their actions following a set of restrictions. These restrictions enforce that probabilistic transitions take precedence over the execution of other actions and that time proceeds synchronously in all components of a system. That is, for $S_1 \| S_2$ to execute a $\surd$ action it must be the case that both S_1 and S_2 are willing to execute an $\surd$ action. Essentially, the intention is that, in any given round of the lifetime of a system, all individuals perform their available actions until they synchronize on their next $\surd$ action and proceed to their next round. Finally, $S \backslash L$ models the restriction of the channels in set L within S. This construct plays an important role in defining valid systems: We define a *valid* system to be any process of the form $S \backslash L$ where, for all of S's subprocesses of the form $a?(P, Q)$ and $\overline{a}?(P, Q)$ we have that $a \in L$. Hereafter, we consider all processes that are valid systems.

Example 1. Let us consider a species $\mathbf{s}$ where individuals cycle through a dispersal phase followed by a reproduction phase. In S-PALPS, we may model $\mathbf{s}$ by P_0, where

$$P_0 \stackrel{\text{def}}{=} \sum_{\ell \in \mathbf{Nb}(\mathsf{myloc})} \frac{1}{4} : go\, \ell . \surd . P_0$$

$$P_1 \stackrel{\text{def}}{=} p{:}\surd.(P_0|P_0) \quad \oplus \quad (1-p){:}\surd.(P_0|P_0|P_0)$$

According to the definition, during the dispersal phase, an individual moves to a neighboring location which is chosen probabilistically among the neighboring locations of the current location (myloc) of the individual. Subsequently, the flow of control proceeds according to P_1 which models the probabilistic production of one (case of $P_0|P_0$) or two offspring (case of $P_0|P_0|P_0$). A system that contains two individuals at a location ℓ and one at location ℓ' can be modeled as

$$System \stackrel{\text{def}}{=} (P_0{:}\langle \mathbf{s}, \ell, 2\rangle | P_0{:}\langle \mathbf{s}, \ell', 1\rangle).$$

Let us now extend the example into a two-species system. In particular, consider a competing species which preys on $\mathbf{s}$ defined as:

$$Q_0 \stackrel{\text{def}}{=} \overline{prey}\,?(\surd.Q_1, \surd.Q_2)$$

$$Q_1 \stackrel{\text{def}}{=} Q_0|Q_0$$

$$Q_2 \stackrel{\text{def}}{=} \overline{prey}\,?(\surd.Q_1, \mathbf{0})$$

An individual of species s' looks for a prey. This is implemented by the conditional process $\overline{prey}\,?(\surd.Q_1, \surd.Q_2)$. If it succeeds in communicating on channel $prey$, which implies that a prey is available, the individual will produce an offspring. If it fails for two consecutive time units it dies.

To implement the possibility of preying on the side of species $\mathbf{s}$, the definition must be extended by introducing the complementary input actions on channel $prey$ at the appropriate places:

$$P_0 \stackrel{\text{def}}{=} prey?\,(\mathbf{0}, \sum_{\ell \in \mathbf{Nb}(\mathsf{myloc})} \frac{1}{4} : go\, \ell . \surd . P_1)$$

$$P_1 \stackrel{\text{def}}{=} prey?\,(\mathbf{0},\ p{:}(\surd.(P_0|P_0)) \quad \oplus \quad (1-p) : \surd.(P_0|P_0|P_0))$$

Note that in the above model we have given higher priority to preying in comparison to other actions. If the success of preying was associated with a certain probability π, we would have written:

$$P_0 \stackrel{\text{def}}{=} \pi{:}prey?\,(\mathbf{0}, \sum_{\ell \in \mathbf{Nb}(\mathsf{myloc})} \frac{1}{4} : go\,\ell.\surd.P_1) + (1-\pi){:}\sum_{\ell \in \mathbf{Nb}(\mathsf{myloc})} \frac{1}{4} : go\,\ell.\surd.P_1$$

$$P_1 \stackrel{\text{def}}{=} \ldots$$

2.2 Semantics

We may now define the semantics of S-PALPS. This is given at the level of configurations of the form (E, S), where E is an *environment* and S is a population system. The environment E is an entity of the form $E \subset \mathbf{Loc} \times \mathbf{S} \times \mathbb{N}$, where each pair ℓ and $\mathbf{s}$ is represented in E at most once and where $(\ell, s, m) \in E$ denotes the existence of m individuals of species s at location ℓ. The environment E plays a central role in evaluating expressions. The satisfaction relation for logical expressions $\models$ is defined inductively on the structure of a logical expression in the same way as in PALPS. Before we proceed to the semantics we define some additional operations on environments that we will use in the sequel:

Definition 1. Consider an environment E, a location ℓ, a species $\mathbf{s}$ and an integer q.

- $E \oplus (\mathbf{s}, \ell, q)$ increases the count of individuals of species $\mathbf{s}$ at location ℓ in environment E by q:

$$E \oplus (\mathbf{s}, \ell, q) = \begin{cases} E' \cup \{(\ell, \mathbf{s}, m+q)\} & \text{if } E = E' \cup \{(\ell, \mathbf{s}, m)\} \text{ for some } m \\ E \cup \{(\ell, \mathbf{s}, q)\} & \text{otherwise} \end{cases}$$

- $E \ominus (\mathbf{s}, \ell, q)$ decreases the count of individuals of species $\mathbf{s}$ at location ℓ in environment E by q:

$$E \ominus (\mathbf{s}, \ell, q) = \begin{cases} E' \cup \{(\ell, \mathbf{s}, m-q)\} & \text{if } E = E' \cup \{(\ell, \mathbf{s}, m)\}, m > q \\ E' & \text{if } E = E' \cup \{(\ell, \mathbf{s}, q)\} \\ \bot & \text{otherwise} \end{cases}$$

The semantics of S-PALPS is defined in terms of a structural congruence, $\equiv$, presented in Table 1 and a structural operational semantics presented in Tables 2 and 3. Beginning with Table 1, of greatest interest are the following congruences: Equivalence (S4) states that operator "$:\langle \ldots \rangle$" distributes over the parallel composition construct and equivalence (S6) states that the parallel composition of q individuals of type P of species $\mathbf{s}$ at location ℓ and r of the same individuals is equivalent to a system with $q+r$ individuals.

Moving on to the structural operation semantics of S-PALPS, this is given in terms of two transition relations: the non-deterministic relation $\longrightarrow_n$ and the probabilistic relation $\longrightarrow_p$. A transition of the form $S \xrightarrow{\mu}_n S'$ means that a system S may execute

Table 1. Structural congruence relation

(S1) $(E, S) \equiv (E, S \| 0)$

(S2) $(E, S_1 \| S_2) \equiv (E, S_2 \| S_1)$

(S3) $(E, (S_1 \| S_2) \| S_3) \equiv (E, S_1 \| (S_2 \| S_3))$

(S4) $(E \cup \{(\mathbf{s}, \ell, n)\}, (P_1 | P_2){:}\langle \mathbf{s}, \ell, q\rangle) \equiv (E \cup \{(\mathbf{s}, \ell, n+q)\}, P_1{:}\langle \mathbf{s}, \ell, q\rangle \parallel P_2{:}\langle \mathbf{s}, \ell, q\rangle)$

(S5) $(E, P{:}\langle \mathbf{s}, \ell, 0\rangle) \equiv (E, 0)$

(S6) $(E, P{:}\langle \mathbf{s}, \ell, q\rangle \parallel P{:}\langle \mathbf{s}, \ell, r\rangle) \equiv (E, P{:}\langle \mathbf{s}, \ell, q+r\rangle)$

action μ and become S'. A transition of the form $S \xrightarrow{w}_p S'$ means that a configuration S may evolve into configuration S' with probability w. Whenever the type of the transition is irrelevant to the context, we write $S \xrightarrow{\alpha} S'$ to denote either $S \xrightarrow{\mu}_n S'$ or $S \xrightarrow{w}_p S'$. We write μ to range over system non-probabilistic activities, which we call actions. Actions are built based on activities of individuals which we call events and denote by β. Events β may have one of the following forms:

- $a_{\ell,\mathbf{s}}$ and $\overline{a}_{\ell,\mathbf{s}}$ denote the execution of a and $\overline{a}$ respectively at location ℓ by individuals of species $\mathbf{s}$.
- $a?_{\ell,\mathbf{s}}$ and $\overline{a}?_{\ell,\mathbf{s}}$ denote the conditional execution of a and $\overline{a}$ respectively at location ℓ by individuals of species $\mathbf{s}$. (This arises in processes of the form $\gamma?(P, Q)$.)
- $\tau_{a,\ell,\mathbf{s}}$ denotes an internal action that has taken place on channel a at location ℓ where the output was carried out by an individual of species $\mathbf{s}$. This may arise when two complementary actions take place at the same location ℓ or when a move action takes place at location ℓ by an individual of species $\mathbf{s}$.
- $\surd$ denotes the time passing action.

In turn μ can have one of the following forms:

- $\beta_1^{k_1} \# \ldots \# \beta_n^{k_n}$ where for all $1 \leq i \leq n$, $\beta_i \neq \surd$, $k_i \geq 1$ and $n \geq 1$, denotes the simultaneous execution of k_i actions of type β_i for $1 \leq i \leq n$, and
- $\surd$ denotes the time passing action.

We may now move on to the semantics of S-PALPS. We begin with the semantics of processes of the form $P{:}\langle \mathbf{s}, \ell, q\rangle$. We discuss these rules separately below:

- Rule (Act) states that a system composed of q individuals, where each can perform an action η, can perform simultaneously q times the action η.

Table 2. Transition rules for single populations

(Act) $\quad (E, (\eta.P){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{(\eta_{\ell,\mathbf{s}})^q}_n (E^{P,\mathbf{s},\ell,q}, P{:}\langle \mathbf{s}, \ell, q\rangle) \qquad \eta \neq go\,\ell', \surd$

(Tick) $\quad (E, (\surd.P){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{\surd}_n (E^{P,\mathbf{s},\ell,q}, P{:}\langle \mathbf{s}, \ell, q\rangle)$

(Go) $\quad (E, (go\,\ell'.P){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{(\tau_{go,\ell,\mathbf{s}})^q}_n ((E \oplus \{(\mathbf{s}, \ell', q)\} \ominus \{(\mathbf{s}, \ell, q)\})^{P,\mathbf{s},\ell',q}, P{:}\langle \mathbf{s}, \ell', q\rangle)$
$(\ell, \ell') \in \mathbf{Nb}$

(Choice) $\quad (E, (\gamma?(P,Q)){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{(\gamma?_{\ell,\mathbf{s}})^n}_n ((E^{P,\mathbf{s},\ell,n})^{Q,\mathbf{s},\ell,q-n}, P{:}\langle \mathbf{s}, \ell, n\rangle || Q{:}\langle \mathbf{s}, \ell, q-n\rangle)$
$0 \leq n \leq q$

(Cond) $$\frac{(E, P{:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{\alpha} (E', Q{:}\langle \mathbf{s}, \ell, q\rangle), P = P_1 \text{ if } E \models e \text{ and } P = P_2, \text{otherwise}}{(E, \mathsf{cond}\ (e \rhd P_1; \text{else } \rhd P_2){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{\alpha} (E', Q{:}\langle \mathbf{s}, \ell, q\rangle)}$$

(PSum) $\quad (E, (\sum_{1\leq i \leq n} p_i{:}P_i){:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{w\langle p_1:q_1,\ldots,p_n:q_n\rangle}_p (E^{\langle P_i,q_i\rangle_I,\mathbf{s},\ell}, P_1{:}\langle \mathbf{s}, \ell, q_1\rangle \| \ldots \| P_n{:}\langle \mathbf{s}, \ell, q_n\rangle)$
$\sum q_i = q$

(RConst) $$\frac{(E, P{:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{\alpha} (E', P'{:}\langle \mathbf{s}, \ell, q\rangle)}{(E, C{:}\langle \mathbf{s}, \ell, q\rangle) \xrightarrow{\alpha} (E', P'{:}\langle \mathbf{s}, \ell, q\rangle)} \quad C \stackrel{\text{def}}{=} P{:}\langle \mathbf{s}, \ell, q\rangle$$

$$\text{where } E^{P,\mathbf{s},\ell,q} = \begin{cases} E \ominus (\mathbf{s}, \ell, q) & \text{if } P = \mathbf{0} \\ E & \text{otherwise} \end{cases}$$

$$E^{\langle P_i,q_i\rangle_I,\mathbf{s},\ell} = \begin{cases} E \ominus (\mathbf{s}, \ell, \sum_{j\in J} q_j) & \text{if } J = \{j \mid P_j = \mathbf{0}\} \\ E & \text{otherwise} \end{cases}$$

- Rule (Tick) states that a system of q individuals, where each can perform action $\surd$, can also perform action $\surd$.
- Rule (Go) states that a system of q individuals, where each individual can perform a moving action, can perform simultaneously q times the action $\tau_{go,\ell,\mathbf{s}}$.
- Rule (Choice) states that a system of q individuals, executing the conditional choice $\gamma?(P,Q)$ may have any number $n \leq q$ of its components executing the action $(\gamma?_{\ell,\mathbf{s}})^n$ and proceedings to state P whereas the remaining $q-n$ of its components will proceed to Q. Note that the nondeterminism apparent in this rule will be resolved once this process is placed in a wider system context. Recall that a valid system including this process would have the form $(\gamma?(P,Q)\|S)\backslash L$, where the channel of action γ belongs to L. As a result, the semantics of the hiding operator $\backslash L$ will resolve the nondeterminism by selecting the value n where n is the number of times action $\overline{\gamma}$ is available in S.

– Rule (Cond) states that a system of q individuals, executing the conditional cond $(e \triangleright P, \text{else } \triangleright Q)$ may either proceed to P or to Q, depending on whether e evaluates to true or false, respectively, in the current environment.
– Rule (PSum) says that a system of q individuals each consisting of the probabilistic choice $\sum_{1 \leq i \leq n} p_i{:}P_i$, can evolve into a parallel composition of q_i processes of process P_i for each $1 \leq i \leq n$, for all combinations of the $q_i \geq 0$, where $\sum_{1 \leq i \leq n} q_i = q$, with probability w given as

$$w_{\langle p_1:q_1,\ldots p_n:q_n\rangle} = \prod_{1 \leq i \leq n} p_i^{q_i} \cdot \binom{q - \sum_{1 \leq j \leq i-1} q_j}{q_i}$$

– Rule (RConst) expresses the semantics of process constants in the expected way.

Note that the rules also update the state of the environment: in case an individual ceases to exists, that is, it becomes $\mathbf{0}$, then it is removed from the environment (see $E^{P,\mathbf{s},\ell,q}$ and $E^{\{(P_i,q_i)\}_I,\mathbf{s},\ell}$) and, if an individual moves from one location to another then the appropriate fields in the environment are updated.

We point that we have not included a rule for the process $(P_1|P_2){:}\langle \mathbf{s}, \ell, q\rangle$ as its semantics is given using structural congruence via the equivalence $(P_1|P_2){:}\langle \mathbf{s}, \ell, q\rangle \equiv P_1{:}\langle \mathbf{s}, \ell, q\rangle \| P_2{:}\langle \mathbf{s}, \ell, q\rangle$ and rule (Struct) presented below.

We may now define the semantics for general systems presented in Table 3:

– Rule (Time) specifies that if two systems may execute a timed action then their parallel composition may also execute a timed action.
– Rule (Par1) considers the case where one of the components in a parallel composition may execute a timed action and the other a non-timed action. According to the rule the non-timed action takes precedence over the timed action. The latter is postponed until both processes may execute the timed action. Note that this as well as the previous rule ensure that systems execute their timed actions in lockstep. In this way time evolves according to a global clock.
– Rules (Par2) and (Par4) consider probabilistic actions of a parallel composition. The first one specifies that if both components of the composition may execute a probabilistic transition then the composition executes a probabilistic transition with probability the product of the two probabilities. The second rule states that if exactly one of the processes may execute a probabilistic transition then the parallel composition may also execute the transition.
– Rule (Par3) says that if two systems can perform non-deterministic actions β^k and μ, respectively, then their parallel composition can perform the combination of the two actions assuming that neither of them is the $\surd$ action. The combination of these actions is defined according to Definition 2 below.
– Rule (Res) states that a restricted process may only execute actions involving channels that do not belong to the restriction set L.
– Rule (Struct) specifies that congruent processes have the same transitions.

Note that in case that the components proceed simultaneously then the environment of the resulting configuration should take into account the changes applied in both of the constituent transitions (rules (Par1), (Par3) and (Time)) as follows:

$$E \otimes (E_1, E_2) = \{(\ell, \mathbf{s}, m + i_1 + i_2) \mid (\ell, \mathbf{s}, m) \in E,\\ (\ell, \mathbf{s}, m + i_1) \in E_1, (\ell, \mathbf{s}, m + i_2) \in E_2, i_1, i_2 \in \mathbb{Z}\}$$

Table 3. Transition rules for systems

(Time)	$\dfrac{(E, S_1) \xrightarrow{\surd}_n (E_1, S_1'), (E, S_2) \xrightarrow{\surd}_n (E_2, S_2')}{(E, S_1 \| S_2) \xrightarrow{\surd}_p (E \otimes (E_1, E_2), S_1' \| S_2')}$
(Par1)	$\dfrac{(E, S_1) \xrightarrow{\mu}_n (E_1, S_1'), (E, S_2) \xrightarrow{\surd}_n (E_2, S_2'), \mu \neq \surd}{(E, S_1 \| S_2) \xrightarrow{\mu}_p (E_1, S_1' \| S_2)}$
(Par2)	$\dfrac{(E, S_1) \xrightarrow{w_1}_p (E_1, S_1'), (E, S_2) \xrightarrow{w_2}_p (E_2, S_2')}{(E, S_1 \| S_2) \xrightarrow{w_1 \cdot w_2}_p (E \otimes (E_1, E_2), S_1' \| S_2')}$
(Par3)	$\dfrac{(E, S_1) \xrightarrow{\beta^k}_n (E_1, S_1'), (E, S_2) \xrightarrow{\mu}_n (E_2, S_2'), \mu \neq \surd}{(E, S_1 \| S_2) \xrightarrow{\beta^k \diamond \mu}_p (E \otimes (E_1, E_2), S_1' \| S_2')}$
(Par4)	$\dfrac{(E, S_1) \xrightarrow{w}_p (E', S_1'), (E, S_2) \not\longrightarrow_p}{(E, S_1 \| S_2) \xrightarrow{w}_p (E', S_1' \| S_2)}$
(Res)	$\dfrac{(E, S) \xrightarrow{\mu} (E', S'), \{a \mid a_{\mathbf{s},\ell}, \overline{a}_{\mathbf{s},\ell} \in \mu\} \cap L = \emptyset}{(E, S \backslash L) \xrightarrow{\mu} (E', S' \backslash L)}$
(Struct)	$\dfrac{(E, S) \equiv (E', S'), (E', S') \xrightarrow{\alpha} (E'', S'')}{(E, S) \xrightarrow{\alpha} (E'', S'')}$

We conclude the semantics with the definition of operator $\diamond$. This operation combines a species action β^k and an action μ by grouping together all actions that are the same and turning complementary transitions into τ actions. Formally:

Definition 2. Consider actions β^k and $\mu \neq \surd$ then

$$\beta^k \diamond \mu = \begin{cases} (\tau_{a,\ell,\mathbf{s}})^k \# \mu' & \text{if } \beta = a_{\mathbf{s},\ell}, \mu = (\overline{a}_{\mathbf{s},\ell})^k \# \mu' \\ (\tau_{a,\ell,\mathbf{s}})^k \# (\overline{a}_{\mathbf{s},\ell})^{k'-k} \# \mu' & \text{if } \beta = a_{\mathbf{s},\ell}, \mu = (\overline{a}_{\mathbf{s},\ell})^{k'} \# \mu', k < k' \\ (\tau_{a,\ell,\mathbf{s}})^{k'} \# (a_{\mathbf{s},\ell})^{k-k'} \# \mu' & \text{if } \beta = a_{\mathbf{s},\ell}, \mu = (\overline{a}_{\mathbf{s},\ell})^{k'} \# \mu', k > k' \\ (a_{\mathbf{s},\ell})^{k+k'} \# \mu' & \text{if } \beta = a_{\mathbf{s},\ell}, \mu = (a_{\mathbf{s},\ell})^{k'} \# \mu', k' \geq 0 \\ (\overline{a}_{\mathbf{s},\ell})^{k+k'} \# \mu' & \text{if } \beta = \overline{a}_{\mathbf{s},\ell}, \mu = (\overline{a}_{\mathbf{s},\ell})^{k'} \# \mu', k' \geq 0 \\ (\tau_{a,\ell,\mathbf{s}})^{k+k'} \# \mu' & \text{if } \beta = \tau_{a,\ell,\mathbf{s}}, \mu = (\tau_{a,\ell,\mathbf{s}})^{k'} \# \mu', k' \geq 0 \\ (\tau_{a,\ell,\mathbf{s}})^k \# \mu' & \text{if } \beta = a?_{\mathbf{s},\ell}, \mu = (\overline{a}_{\mathbf{s},\ell})^{k'} \# \mu', k' \geq k \\ (\tau_{a,\ell,\mathbf{s}})^k \# \mu' & \text{if } \beta = \overline{a}?_{\mathbf{s},\ell}, \mu = (a_{\mathbf{s},\ell})^{k'} \# \mu', k' \geq k \\ \bot & \text{otherwise} \end{cases}$$

Initial configuration. Based on this machinery, the semantics of a system S is obtained by applying the semantic rules to the initial configuration. The initial configuration, (E, S), is such that $(\ell, \mathbf{s}, m) \in E$ if and only if S contains exactly m non-**0** individuals of species $\mathbf{s}$ located at ℓ. In general, we say that E is *compatible* with S whenever $(\ell, \mathbf{s}, m) \in E$ if and only if S contains exactly m active (non-0) individuals of species $\mathbf{s}$ located at ℓ. It is possible to prove that the defined semantics preserves compatibility of configurations:

Lemma 1. *Whenever $(E, S) \stackrel{\alpha}{\longrightarrow} (E', S')$ and E is compatible with S, then E' is also compatible with S'.*

Example 2. Consider $P_1 \stackrel{\mathrm{def}}{=} a?(P_2, P_3)$, $Q_1 \stackrel{\mathrm{def}}{=} a.Q_2$ and $R_1 \stackrel{\mathrm{def}}{=} \overline{a}.R_2$. Further, suppose that $S \stackrel{\mathrm{def}}{=} (P_1{:}\langle \mathbf{s}_1, \ell, 3\rangle \| Q_1{:}\langle \mathbf{s}_2, \ell, 4\rangle \| R_1{:}\langle \mathbf{s}_3, \ell, 5\rangle)\backslash\{a\}$. Then we have the following transitions, where for simplicity we abbreviate $(E, T) \stackrel{\alpha}{\longrightarrow} (E, T')$ by $T \stackrel{\alpha}{\longrightarrow} T'$ for $E = \{(\mathbf{s}_1, \ell, 3), (\mathbf{s}_2, \ell, 4), (\mathbf{s}_3, \ell, 4)\}$.

$$\begin{aligned}
P_1{:}\langle \mathbf{s}_1, \ell, 3\rangle &\stackrel{(a?_{\ell,\mathbf{s}_1})^i}{\longrightarrow_n} P_2{:}\langle \mathbf{s}_1, \ell, i\rangle \| P_3{:}\langle \mathbf{s}_1, \ell, 3-i\rangle, \quad 0 \le i \le 3 \\
Q_1{:}\langle \mathbf{s}_2, \ell, 4\rangle &\stackrel{(a_{\ell,\mathbf{s}_2})^4}{\longrightarrow_n} Q_2{:}\langle \mathbf{s}_2, \ell, 4\rangle \\
R_1{:}\langle \mathbf{s}_3, \ell, 5\rangle &\stackrel{(\overline{a}_{\ell,\mathbf{s}_3})^5}{\longrightarrow_n} R_2{:}\langle \mathbf{s}_3, \ell, 5\rangle
\end{aligned}$$

Additionally,

$$Q_1{:}\langle \mathbf{s}_2, \ell, 4\rangle \| R_1{:}\langle \mathbf{s}_3, \ell, 5\rangle \stackrel{(\tau_{\alpha,\ell,\mathbf{s}_3})^4 \# (\overline{a}_{\ell,\mathbf{s}_3})^1}{\longrightarrow_n} Q_2{:}\langle \mathbf{s}_2, \ell, 4\rangle \| R_2{:}\langle \mathbf{s}_3, \ell, 5\rangle$$

and now $P_1{:}\langle \mathbf{s}_1, \ell, 3\rangle$, by the definition of $\diamond$, may only communicate with the system above via its action $(a?_{\ell,\mathbf{s}_1})^1$, thus yielding:

$$S \stackrel{(\tau_{\alpha,\ell,\mathbf{s}_3})^5}{\longrightarrow_n} (P_2{:}\langle \mathbf{s}_1, \ell, 1\rangle \| P_3{:}\langle \mathbf{s}_1, \ell, 2\rangle \| Q_2{:}\langle \mathbf{s}_2, \ell, 4\rangle \| R_2{:}\langle \mathbf{s}_3, \ell, 5\rangle)\backslash\{a\}$$

3 Translating S-PALPS into PRISM

In this section we turn to the problem of model checking S-PALPS. As is the case of PALPS, the operational semantics of S-PALPS gives rise to transition systems that can be easily translated to Markov decision processes (MDPs). As such, model checking approaches that have been developed for MDPs can also be applied to S-PALPS models. PRISM is one such tool developed for the analysis of probabilistic systems. Specifically, it is a probabilistic model checker for Markov decision processes, discrete time Markov chains, and continuous time Markov chains. For our study we are interested in the MDP support of the tool which offers model checking and simulation capabilities.

In [13] we defined a translation of PALPS into the MDP subset of the PRISM language and we explored the possibility of employing PRISM to perform analysis of the semantic models derived from PALPS processes. In this paper, we redefine a translation which implements the synchronous parallel operator of S-PALPS. In the remainder of this section, we give a brief presentation of the PRISM language, present an encoding of S-PALPS into PRISM and prove its correctness.

3.1 The PRISM language

The PRISM language is a simple, state-based language, based on guarded commands. A PRISM model consists of a set of *modules* which can interact with each other on shared actions following the CSP-style of communication [1]. Each module possesses a set of *local variables* which can be written by the module and read by all modules. In addition, there are *global variables* which can be read and written by all modules. The behavior of a module is described by a set of *guarded commands*. When modeling Markov decision processes, these commands take the form:

[act] guard p_1 : u_1 + ... + p_m :u_m;

where act is an optional action label, guard is a predicate over the set of variables, $p_i \in (0,1]$ and u_i are updates of the form:

(x'_1 = $u_{i,1}$) & ... & (x'_k = $u_{i,k}$)

where $u_{i,j}$ is a function over the variables. Intuitively, such an action is enabled in global state s if s satisfies guard. If a command is enabled then it may be executed in which case, with probability p_i, the update u_i is performed by setting the value of each variable x_j to $u_{i,j}(s)$ (where x'_j denotes the new value of variable x_j). We refer the reader to [1] for the full description and the semantics of the PRISM language.

3.2 Encoding S-PALPS into the PRISM language

Consider an S-PALPS model. This consists of a set of locations, the neighborhood relation **Nb** and a process $System$. In turn, the process $System$ satisfies $System \equiv (P_1{:}\langle \mathbf{s}_1, \ell_1, q_1\rangle \| \ldots \| P_n{:}\langle \mathbf{s}_n, \ell_n, q_n\rangle)\backslash L$, where each P_i is a process that may evolve to a set of states, say $P_i^j, 1 \le j \le m_i$. This allows us to conclude that in any state $System'$ reachable from $System$, we have $System' \equiv (\prod_{i\in I, j\in J, \ell\in \mathbf{Loc}} P_i^j{:}\langle \mathbf{s}_i, \ell, q_{i,j,\ell}\rangle)\backslash L$, that is, at any point in time, there may be an arbitrary number of individuals in each location and of each of the reachable states of the populations.

Based on this observation, our translation of $System$ in PRISM consists of a set of $(m_1 + \ldots + m_n) \cdot |\mathbf{Loc}|$ modules, where $|\mathbf{Loc}|$ is the total number of locations existing in the system. Each module captures the behavior of the individuals in the specific state and location. Note that the total number of modules is stable and independent of the precise number of individuals existing in the model. This comes in contrast with our PALPS translation of [13] where the translation of a model contained one module for each individual a fact that resulted in restrictions in space and expressiveness.

In addition to these module definitions, a system translation in PRISM should contain the following global information relating to the system.

- For each species $\mathbf{s}_i$ and each state j in the process description of $\mathbf{s}_i$, the model contains the $|\mathbf{Loc}|$ global variables $s_{i,j,\ell}$, $\ell \in \mathbf{Loc}$ capturing the number of individuals of species $\mathbf{s}_i$ in state j for location ℓ. The variables should be appropriately initialized based on the definition of $System$.
- For each channel a on which synchronization may take place we introduce a variable a_y which counts the number of available inputs on a at location y and a variable $\overline{a_y}$ which counts the number of available outputs on a at location y.

- There exists a global variable $pact$ which may take values from $\{0, 1\}$ and expresses if there is a probabilistic action enabled. It is used to give precedence to probabilistic actions over nondeterministic actions. Initially, $pact = 0$. Furthermore, all non-probabilistic actions have $pact = 0$ as a precondition.
- There exists a global variable $tact$ which may take values from $\{0, 1\}$ and expresses whether a timed action may take place. For such an action to take place it must be that $tact = 1$. If any process is unable to execute the $\surd$ action then it sets $tact$ to 0.

As an example, consider processes P_1, Q_1, R_1 and $System$ of Example 2 and suppose that our the system is located on a habitat consisting of 2 patches $\{\ell, \ell'\}$. Then the skeleton of the PRISM translation is as follows:

global $s_{1,1,1} : [0, max]$ init 3;
global $s_{2,1,1} : [0, max]$ init 4;
global $s_{3,1,1} : [0, max]$ init 5;
global $s_{1,1,2}, s_{2,1,2}, s_{3,1,2} : [0, max]$ init 0;
global $s_{1,2,1}, s_{2,2,1}, s_{3,2,1} : [0, max]$ init 0;
global $s_{1,2,2}, s_{2,2,2}, s_{3,2,2}$ init 0;
global $pact : [0, 1]$ init 0;
global $tact : [0, 1]$ init 0;

module $S_{1,1,1}$
$\quad\ldots$

We now continue to describe how a specific module is described by considering the above example. Specifically, consider process $Q_1 \stackrel{\text{def}}{=} a.Q_2$ and an initial population $Q_1{:}\langle \mathrm{s}_2, \ell, 4\rangle$. Then, according to the semantics of S-PALPS, these 4 individuals should synchronize on channel a and become individuals in state Q_2. To model this in PRISM these 4 individuals should *flow* from their current state to their next state. To achieve this we need to make the necessary updates on the global variables $\mathrm{s}_{2,1,1}$ and $\mathrm{s}_{2,2,1}$, specifically: $\mathrm{s}'_{2,1,1} = \mathrm{s}_{2,1,1} - 4$ and $\mathrm{s}'_{2,2,1} = \mathrm{s}_{2,2,1} + 4$. Furthermore, if state Q_2 is a probabilistic state, the module should set $pact' = 1$.

Now to implement the synchronization of the module with all other modules executing an action we need to execute a sequence of actions as illustrated below:

module $S_{2,1,1}$
$st_{2,1,1} : [0..5]\ \ init\ \ 1;$
$n_{2,1,1} : [0..max]$
$\quad[\,]\quad (st_{2,1,1} = 1)\&(s_{2,1,1} > 0) \longrightarrow (st'_{2,1,1} = 2)\&(tact' = 0)\&(n'_{2,1,1} = s_{2,1,1});$
$\quad[\,]\quad (st_{2,1,1} = 1)\&(s_{2,1,1} = 0) \longrightarrow (st'_{2,1,1} = 2)\&(n'_{2,1,1} = s_{2,1,1});$
$\quad[synch]\ (st_{2,1,1} = 2) \longrightarrow (st'_{2,1,1} = 3);$
$\quad[\,]\quad (st_{2,1,1} = 3)\&(pact = 0) \longrightarrow (a'_\ell = a_\ell + n_{2,1,1})\&(st'_{2,1,1} = 4);$
$\quad[a_\ell]\quad (st_{2,1,1} = 4) \longrightarrow (st'_{2,1,1} = 5);$
$\quad[\,]\quad (st_{2,1,1} = 5) \longrightarrow (a'_\ell = 0)\&\mathrm{updates}(Q_1, Q_2)\&(st'_{2,1,1} = 1);$
$\quad[prob]\quad (st_{2,1,1} = 3)\&(pact = 1) \longrightarrow (st'_{2,1,1} = 1)$
endmodule

Variable $st_{2,1,1}$ in module $S_{2,1,1}$ (initially set to 1) will guide the flow of execution of the required sequence of actions. It begins by testing whether there are active individuals of this module (state 1) and then proceeds to synchronize with the other modules. This synchronization will take place on action $synch$. Subsequently, if there are 1 or more modules in a probabilistic state the module will synchronize with them via action $prob$, otherwise, the module will proceed to make its necessary updates: $s'_{2,1,1} = s_{2,1,1} - n_{2,1,1} \& s'_{2,2,1} = s_{2,2,1} + n_{2,1,1}$. Furthermore, if $Q_2 = \surd.Q_3$ then the update $tact = 1$ is included, whereas if $Q_2 = p_1 : T_1 \oplus \dots p_n : T_n$, then the update $pact = 1$ is included.

Let us now discuss some characteristics of the above translation which are also relevant to the translations of process constructs other than $a.P$. To begin with, the module begins by setting $tact = 0$, assuming that there are active individuals in this state. Thus, it is ensured that nondeterministic actions take precedence over timed actions. In addition, variable $n_{2,1,1}$ is used to store the initial population of the module. This is necessary because other processes may 'flow' into this module and the value of $s_{2,1,1}$ may subsequently not reflect the initial size of the population. Furthermore, we point out that if a probabilistic action is available $(pact = 1)$ then the process will synchronize on this action and return to its initial state. We also note that it is not possible to collapse e.g. states 2 and 3 of the module because PRISM does not allow to execute updates on global variables within synchronization actions. Finally, we observe that in the case of channel communication, the module records the number of available inputs and outputs on a channel at a certain location (update $a'_\ell = a_\ell + n$) and continues to synchronize on action a_ℓ. This is required for translating the restriction construct where we must check that the number of inputs and outputs performed on the channel are equal.

In a similar manner we may translate all constructs of the S-PALPS syntax by allowing processes to flow from one module to the next. In the next subsection we consider the complete translation.

3.3 Formal translation

In this section, we formalize the intuitions of the previous example into a formal translation of S-PALPS into PRISM and we prove its correctness. Specifically, we will describe how a population of individuals may flow from one state to a next state by considering the translation of a set of individuals of species $\mathbf{s}_i$ in state j and location ℓ. The translation depends on the definition of the S-PALPS state and it is defined inductively on its structure.

The translation assumes the set of global variables considered in the previous subsection and for each process of the form $P_j{:}\langle \mathbf{s}_i, \ell, k\rangle$, it employs a local variable $st_{i,j,\ell}$ which records the current state in the flow of execution of the translation and $s_{i,j,\ell}$, initially set to k which records the number of individuals of the specific species, state and location. In what follows, we write $\mathrm{prob}(P)$ and $\mathrm{timed}(P)$ for the logical values that express whether P is a probabilistic or a timed process respectively. Furthermore, we write $\mathrm{updates}(P_i, P_j, \ell_1, \ell_2)$ for the set of updates necessary when P_i at location ℓ_1, evolves into state P_j at location ℓ_2. If $P_m = \mathbf{0}$, we define $\mathrm{updates}(P_j, \mathbf{0}, \ell, \ell') =$

$(tact' = 1)\&(st'_{i,j,\ell} = st_{i,j,\ell} - n_{i,j,\ell})$, otherwise we have

$$\begin{aligned}\text{updates}(P_j, P_m, \ell, \ell') = (pact' = \text{prob}(P_m))\&(tact' = 1)\\ \&(st'_{i,j,\ell} = st_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,m,\ell'} = st_{i,m,\ell'} + n_{i,j,\ell})\end{aligned}$$

Moving on to the translation, let us consider state P_j of species $\mathbf{s}_i$ at location ℓ. The following cases exist:

Case 1: $P_j = go\,\ell'.P_m$. We translate this activity as follows:

$$\begin{aligned}
[]\ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[]\ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(n'_{i,j,\ell} = 0);\\
[synch]\ & (st_{i,j,\ell} = 2) \longrightarrow (s'_{i,j,\ell} = 3);\\
[]\ & (st_{i,j,\ell} = 3)\&(pact = 0)\&((\ell, \ell') \in \mathbf{Nb}) \longrightarrow \text{updates}(P_j, P_m, \ell, \ell')\&(st'_{i,j,\ell} = 1);\\
[prob]\ & (st_{i,j,\ell} = 3)\&(pact = 1) \longrightarrow (st'_{i,j,\ell} = 1);
\end{aligned}$$

According to this definition, the module begins by setting $tact$ to 0 if there exist individuals in the state, thus proclaiming its inability of performing a timed action. It also initializes its local variable $n_{i,j,\ell}$ recording the initial number of individuals in the state. Subsequently, the module continues to synchronize on action $synch$ with all other modules. Then, if no probabilistic action is enabled, it proceeds to make all its necessary updates: the number of individuals of state P_i is reduced by n and the number of individuals of state P_j is increased by n. If, however, a probabilistic action is enabled then the module will synchronize on action $prob$, thus giving precedence to the probabilistic action.

Case 2: $P_j = a.P_m$. This state is translated as follows:

$$\begin{aligned}
[]\ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[]\ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[synch]\ & (st_{i,j,\ell} = 2) \longrightarrow (st'_{i,j,\ell} = 3);\\
[]\ & (st_{i,j,\ell} = 3)\&(pact = 0) \longrightarrow (a'_\ell = a_\ell + n_{i,j,\ell})\&(st'_{i,j,\ell} = 4);\\
[a_\ell]\ & (st_{i,j,\ell} = 4) \longrightarrow (st'_{i,j,\ell} = 5);\\
[]\ & (st_{i,j,\ell} = 5) \longrightarrow (a'_\ell = 0)\&\text{updates}(P_j, P_m, \ell, \ell)\&(st'_{i,j,\ell} = 1);\\
[prob]\ & (st_{i,j,\ell} = 3)\&(pact = 1) \longrightarrow (st'_{i,j,\ell} = 1)
\end{aligned}$$

The translation follows along similar lines to the previous case. The module begins by initializing variables according to the initial value of $s_{i,j,\ell}$. It then proceeds to synchronize with all other modules on action $synch$. After the synchronization, depending on the value of $pact$ it either proceeds to execute its transition in states $3-5$ or it participates in the $prob$ synchronization and postpones its own execution. In the former case, it increases the value of a_ℓ signifying the availability of $n_{i,j,\ell}$ additional occurrences of a at location ℓ. All such actions then synchronize on action a_ℓ before setting the variable a_ℓ to 0 and moving the $n_{i,j,\ell}$ individuals to their next state while appropriately changing the values of variables $s_{i,j,\ell}$ and $s_{i,m,\ell}$. Note that the counter of input actions, that is variable a_ℓ, is necessary for handling the restriction operator, and will be revisited at the end of this section.

Case 3: $P_j = \overline{a}.P_m$. The translation of this state follows similarly to the previous case.

$$\begin{array}{rl}
[] & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[] & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[synch] & (st_{i,j,\ell} = 2) \longrightarrow (st'_{i,j,\ell} = 3);\\
[] & (st_{i,j,\ell} = 3)\&(pact = 0) \longrightarrow (\overline{a_\ell}' = \overline{a}_\ell + n_{i,j,\ell})\&(st'_{i,j,\ell} = 4);\\
[a_\ell] & (st_{i,j,\ell} = 4) \longrightarrow (st'_{i,j,\ell} = 5);\\
[] & (st_{i,j,\ell} = 5) \longrightarrow (\overline{a_\ell}' = 0)\&\text{updates}(P_j, P_m, \ell, \ell)\&(st'_{i,j,\ell} = 1);\\
[prob] & (st_{i,j,\ell} = 3)\&(pact = 1) \longrightarrow (st'_{i,j,\ell} = 1)
\end{array}$$

We point out that the module synchronizes through action a_ℓ with all modules executing output as well as input actions on a at location ℓ. This is important for defining the restriction operator as we will see below.

Case 4: $P_j = \surd.P_m$. We translate the process by including the commands

$$\begin{array}{rl}
[synch] & (st_{i,j,\ell} = 1) \longrightarrow (st'_{i,j,\ell} = 2);\\
[tick] & (st_{i,j,\ell} = 2)\&(tact = 1) \longrightarrow (st'_{i,j,\ell} = 3);\\
[] & (st_{i,j,\ell} = 3) \longrightarrow \text{updates}(P_j, P_m, \ell, \ell)\&(st'_{i,j,\ell} = 1);\\
[] & (st_{i,j,\ell} = 2)\&(tact = 0) \longrightarrow (st'_{i,j,\ell} = 1);
\end{array}$$

In the first step of this translation the module synchronizes with other modules on action $synch$. Subsequently, if $tact = 1$, it synchronizes with all other modules on action $tick$ and then continues to perform its associated updates before returning to the initial state. Alternatively, if a timed action is not enabled then the module returns to its initial state.

Case 5: $P_j = p_1 : P_{j_1} + \ldots + p_m : P_{j_m}$. The translation consists of the following commands.

$$\begin{array}{rl}
[] & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[] & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(n'_{i,j,\ell} = s_{i,j,\ell});\\
[synch] & (st_{i,j,\ell} = 2) \longrightarrow (st'_{i,j,\ell} = 3);\\
[] & (st_{i,j,\ell} = 3)\&(n_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 1);\\
[prob] & (st_{i,j,\ell} = 3)\&(n_{i,j,\ell} = 1) \longrightarrow S_1\\
 & \ldots\\
[prob] & (st_{i,j,\ell} = 3)\&(n_{i,j,\ell} = max) \longrightarrow S_{max}
\end{array}$$

where for $n = x$ we define S_x by

$$w_{p_1:q_1^1,\ldots,p_m:q_m^1} : st'_{i,j,\ell} = st_{q_1^1,\ldots q_m^1} + \ldots + w_{p_1:q_1^k,\ldots,p_m:q_m^k} : st'_{i,j,\ell} = st_{q_1^k,\ldots q_m^k}$$

denoting all possible ways of choosing for each of the x components one of the processes P_{j_k} and where the probabilities $w_{p_1:q_1^1,\ldots,p_m:q_m^1}$ are as defined in the semantics, and where state $st_{q_1^k,\ldots q_n^k}$ is responsible for appropriately updating the multiplicity of each of the P_{j_r} components as shown below.

$$
\begin{aligned}
st_{i,j,\ell} = st_{q_1^k,\ldots q_n^k} \longrightarrow\ & (s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell}) \\
& \&(s_{i,j_1,\ell} = s_{i,j_1,\ell} + q_1^k)\&\ldots\&(s_{i,j_m,\ell} = s_{i,j_m,\ell} + q_m^k)
\end{aligned}
$$

Case 6: $P_j = a?(P_{j_1}, P_{j_2})$. The conditional choice construct translation is treated differently in the case of an input and an output action. We begin be considering the input action. The translation needs to take into account the fact that, given a set of outputs on a then all available inputs on a must be satisfied before executing the $a?$ action of the conditional choice. If there remain some output actions, then the appropriate number of P_{j_1} derivatives will be chosen whereas the remaining processes will proceed as P_{j_2}. The translation follows:

$$
\begin{aligned}
[]\ \ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell}); \\
[]\ \ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2); \\
[synch]\ \ & (st_{i,j,\ell} = 2) \longrightarrow (st'_{i,j,\ell} = 3); \\
[]\ \ & (st_{i,j,\ell} = 3)\&(pact = 0) \longrightarrow (st'_{i,j,\ell} = 4)\&(a'_\ell = a_\ell + n_{i,j,\ell}); \\
[a_\ell]\ \ & (st_{i,j,\ell} = 4) \longrightarrow (st'_{i,j,\ell} = 5); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(\overline{a}_\ell \geq a_\ell) \longrightarrow (s'_{i,j_1,\ell} = s_{i,j_1,\ell} + n_{i,j,\ell}) \\
& \qquad\qquad \&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(0 \leq a_\ell - \overline{a_\ell} \leq n_{i,j,\ell}) \longrightarrow (a'_\ell = \overline{a}_\ell)\&(s'_{i,j_1,\ell} = s_{i,j_1,\ell} + n_{i,j,\ell} - a_\ell + \overline{a}_\ell) \\
& \quad \&(s'_{i,j_2,\ell} = s_{i,j_2,\ell} + a_\ell - \overline{a}_\ell)\&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(a_\ell - \overline{a_\ell} > n_{i,j,\ell}) \longrightarrow (a'_\ell = a_\ell - n_{i,j,\ell}) \\
& \quad \&(s'_{i,j_2,\ell} = s_{i,j_2,\ell} + n_{i,j,\ell})\&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[prob]\ \ & (st_{i,j,\ell} = 3)\&(pact = 1) \longrightarrow (st'_{i,j,\ell} = 1)
\end{aligned}
$$

Case 7: $P_j = \overline{a}?(P_{j_1}, P_{j_2})$. This is symmetric to the previous case:

$$
\begin{aligned}
[]\ \ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0) \longrightarrow (st'_{i,j,\ell} = 2)\&(tact' = 0)\&(n'_{i,j,\ell} = s_{i,j,\ell}); \\
[]\ \ & (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} = 0) \longrightarrow (st'_{i,j,\ell} = 2); \\
[synch]\ \ & (st_{i,j,\ell} = 2) \longrightarrow (st'_{i,j,\ell} = 3); \\
[]\ \ & (st_{i,j,\ell} = 3)\&(pact = 0) \longrightarrow (st'_{i,j,\ell} = 4)\&(\overline{a_\ell}' = \overline{a_\ell} + n_{i,j,\ell}); \\
[a_\ell]\ \ & (st_{i,j,\ell} = 4) \longrightarrow (st'_{i,j,\ell} = 5); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(a_\ell \geq \overline{a_\ell}) \longrightarrow (s'_{i,j_1,\ell} = s_{i,j_1,\ell} + n_{i,j,\ell}) \\
& \qquad\qquad \&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(0 \leq \overline{a}_\ell - a_\ell \leq n_{i,j,\ell}) \longrightarrow (\overline{a_\ell}' = a_\ell)\&(s'_{i,j_1,\ell} = s_{i,j_1,\ell} + n_{i,j,\ell} - \overline{a_\ell} + a_\ell) \\
& \quad \&(s'_{i,j_2,\ell} = s_{i,j_2,\ell} + \overline{a_\ell} - a_\ell)\&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[]\ \ & (st_{i,j,\ell} = 5)\&(\overline{a}_\ell - a_\ell > n_{i,j,\ell}) \longrightarrow (\overline{a}'_\ell = \overline{a}_\ell - n_{i,j,\ell}) \\
& \quad \&(s'_{i,j_2,\ell} = s_{i,j_2,\ell} + n_{i,j,\ell})\&(s'_{i,j,\ell} = s_{i,j,\ell} - n_{i,j,\ell})\&(st'_{i,j,\ell} = 1); \\
[prob]\ \ & (st_{i,j,\ell} = 3)\&(pact = 1) \longrightarrow (st'_{i,j,\ell} = 1)
\end{aligned}
$$

Case 8: $P_j = \mathsf{cond}\ (e \rhd P_{j_1}, else \rhd P_{j_2})$. The translation of this construct transforms the process into the appropriate continuation depending on the value of e. Note that this module does not synchronize with any other module since it is not actually performing an action but instead it rewrites itself by spontaneous transformation into the appropriate module, assuming that no probabilistic action is available. If such an action is available, the rewriting is postponed since it is possible that execution of a probabilistic actions will change the state and hence the evaluation of $e@\ell$.

$$[]\ \ (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0)\&[\![e@\ell]\!]\&(pact = 0) \longrightarrow (s'_{i,j_1,\ell} = s_{i,j,\ell} + s_{i,j,\ell})\&(s'_{i,j,\ell} = 0);$$
$$[]\ \ (st_{i,j,\ell} = 1)\&(s_{i,j,\ell} > 0)\&![\![e@\ell]\!]\&(pact = 0) \longrightarrow (s'_{i,j_2,\ell} = s_{i,j_2,\ell} + s_{i,j,\ell})\&(s'_{i,j,\ell} = 0);$$

where $[\![e@\ell]\!]$ is the translation of the PALPS expression $e@\ell$ into the PRISM language.

Case 9: $P_j = C$, $C \stackrel{\text{def}}{=} P_k$. We translate the process by

$$[](s_{i,j,\ell} > 0) \longrightarrow (s'_{i,k,\ell} = (s_{i,k,\ell} + s_{i,j,\ell})\&((s'_{i,j,\ell} = 0);$$

Case 10: $P_i = 0$. We include no translation for this state.

As a final note let us consider a system of the form $S\backslash L$. As already discussed, the process S can be translated into a set of modules according to the set of species involved in S, the possible states each of these species can engage in and the set of existing locations. To further take into account the $\backslash L$ construct, we simply introduce the condition $a_\ell = \overline{a}_\ell$ to all actions named a_ℓ for all $a \in L$.

3.4 Correctness of the translation

We now turn to consider the correctness of the proposed translation. This is demonstrated via the following two theorems. In what follows, given a PRISM model M, we write $M \xrightarrow{\alpha,p_i} M_i$ if M contains an action `[`α`] guard ->` $p_1 : u_1 + \ldots + p_m : u_m$; where `guard` is satisfied in model M and execution of u_i gives rise to model M_i. Note that if a guarded command has no action label, then we simply write $M \longrightarrow M'$. Furthermore, we write $M \stackrel{\alpha_1\ldots\alpha_n}{\Longrightarrow} M'$ if $M \xrightarrow{\alpha_1,1} M_1 \ldots \xrightarrow{\alpha_m,1} M'$, that is, M may evolve into M' after a sequence of m moves each of which is executed with probability 1. Finally, we write $M \stackrel{p}{\Longrightarrow} M'$ if $M \Longrightarrow M_1 \xrightarrow{\alpha,p} M_2 \Longrightarrow M'$ for some $0 \le p \le 1$.

Theorem 1. For any configuration (E, Sys), where E is compatible with Sys, the following hold:

1. If $(E, Sys) \xrightarrow{\mu}_n (E', Sys')$ then $[\![(E, Sys)]\!] \stackrel{\widetilde{a}}{\Longrightarrow} [\![(E', Sys')]\!]$, for some $\widetilde{a} = a_1 \ldots a_m$.
2. If $(E, Sys) \xrightarrow{w}_p (E', Sys')$ then $[\![(E, Sys)]\!] \stackrel{synch}{\Longrightarrow}\stackrel{w}{\Longrightarrow} [\![(E', Sys')]\!]$.

Theorem 2. For any configuration (E, Sys), where E is compatible with Sys, the following hold:

1. If $[\![(E, Sys)]\!] \stackrel{\widetilde{a}}{\Longrightarrow} \stackrel{prob,w}{\longrightarrow} M$ then $\widetilde{a} = synch$ and, in addition, $(E, Sys) \stackrel{w}{\longrightarrow}_p (E', Sys')$ and for all $M \Longrightarrow M'$ there exists M'' such that $M \Longrightarrow M'' \Longrightarrow M'$ and $M'' = [\![(E', Sys')]\!]$.
2. If $[\![(E, Sys)]\!] \stackrel{a_1 \ldots a_{m-1}}{\Longrightarrow} \stackrel{a_m}{\longrightarrow} M$ then $a_1 = synch$ and if there exists no $synch$ among the $a_2 \ldots a_m$ and $M \stackrel{synch}{\Longrightarrow}$, then $(E, Sys) \stackrel{\alpha}{\longrightarrow} (E', Sys')$ and for all $M \Longrightarrow M'$ there exists M'' such that $M \Longrightarrow M'' \Longrightarrow M'$ and $M'' = [\![(E', Sys')]\!]$,

Theorem 1 establishes that each transition of (E, Sys) can be mimicked by its translation module in a sequence of steps. Theorem 2 considers the other direction of the correctness and it illustrates that any sequence of transitions of a PRISM translation corresponds to a sequence of transitions at the PALPS level. Given a transition of a PRISM module there are two possibilities. On the one hand, if the transition leads to a $prob$ transition, then a probabilistic action with the same probability may take place at the PALPS level and the PRISM module will inevitably lead to the translation of the resulting PALPS state. On the other hand, if the transition of the module executes a sequence of actions, then the first action ought to be a $synch$ action and furthermore, if the subsequent transitions correspond to an execution fragment proceeding the next $synch$ action in the sequence, then an action may take place at the PALPS level and the PRISM module will inevitably lead to the translation of the resulting PALPS state.

Sketch of the proof of Theorem 1: The proof considers two cases. To begin with, we assume that Sys consists of one component and the possible transitions of the form $(E, Sys) \stackrel{\alpha}{\longrightarrow} (E', Sys')$ are as specified in Table 2. A case analysis on the possible forms of Sys and their translation into PRISM confirms that on the completion of the translation state $[\![(E', Sys')]\!]$ will be reached. The second case consists of the case of multiple populations where the transition has arisen by application of one of the rules of Table 3. A case analysis again confirms that the step can be mimicked by the PRISM module. In all cases, we may construct the PRISM transition is phases: first all preparatory steps of the PRISM modules are taken. Consequently, the processes synchronize on their $synch$ actions and then they may each complete execution in a sequence that matches the PALPS transition. □

Sketch of the proof of Theorem 2: The proof consists of an induction proof on the number of components of Sys. It follows along similar lines to the proof of Theorem 1. The important point to note here is that, in all cases, the intermediate step M captures correctly both the environment E' as well as the state Sys' in the transition $(E, Sys) \stackrel{\alpha}{\longrightarrow} (E', Sys')$. Note that this intermediate state is reached exactly when all modules execute the code relating to their participating action. □

4 Case study: Eleonora's falcon population dynamics

In this section we study the Eleonora's falcon [20] using S-PALPS. Eleonora's falcon is a migrant species that breeds on Mediterranean islands and winters on islands of the Indian Ocean and along the eastern African coast. A large part of the world population concentrates on a small number of islands in the Aegean Sea [7]. In Europe, the species is considered as rare and hence of local conservation importance because, although not globally threatened, its world population is below 10,000 breeding pairs and its survival in Europe is highly dependent on the breeding conditions on the islands on which it concentrates. In particular, the breeding calendar of the Eleonora's falcon overlaps with the summer months when tourism peaks in most Mediterranean islands while the climatic changes may also have consequences on the reproduction of the species.

The life cycle of the Eleonora's Falcon is defined as follows. The juveniles disperse from the island during their first year of life. It takes them approximately four years to achieve sexual maturity and they only come back to the island when they reach this age. When they return, they choose a nest. For the sake of model simplicity we consider two types of nests in terms of provision of shelter to the breeding pairs and their young: exposed nests (e.g., to predators, sun, humans and wind) and less-exposed nests. The choice of the nest determines the survival probability of the offspring. According to studies, first-year breeders usually do not choose less-exposed nests. This choice is reserved for mature adults, who are not guaranteed to acquire a less-exposed nest due to the limited number of such nests [16]. The life cycle of this species is presented in Figure 1. In what follows we construct a model of the Eleonora's falcon ecosystem in S-PALPS.

Spatial domains. We consider two spatial domains which we model as two S-PALPS locations: The island where the colony lives, ℓ_1, and the territory outside the island, ℓ_2. The spatial location of the nests on the island is not crucial for our model, hence the use of a single breeding location.

Species. To enable the modeling of the system we define two S-PALPS species in our model: the *Eleonora's falcon* (f) and the *less-exposed nests* (le). We then model the selection of less-exposed nests as a predator-prey problem.

Processes. We associate each of the above species with an S-PALPS description. To model nests, we create a group of n less-exposed nests as $\mathit{LeNest}{:}\langle le, l_1, n\rangle$ such that

$$\mathit{LeNest} \stackrel{\text{def}}{=} \mathit{prey}.\mathit{LeNest}' + \surd.\mathit{LeNest}$$
$$\mathit{LeNest}' \stackrel{\text{def}}{=} \overline{\mathit{release}}.\mathit{LeNest} + \surd.\mathit{LeNest}'$$

The life cycle of a falcon begins in the newborn/juvenile state (process J_0 below). In this state an individual disperses to location ℓ_2 and waits for 4 years which, in our model, consists of 4 occurrences of action $\surd$, before becoming a first-year breeder adult (process $A_1^{\bigcirc}$ below). Note that not all juveniles will mature to adults. In fact, a juvenile may die with a mortality rate of 78% [15].

$$J_0 \stackrel{\text{def}}{=} \mathit{go}\,\ell_2.\surd.\surd.\surd.\surd.J_4$$
$$J_4 \stackrel{\text{def}}{=} (0.78 : 0 \oplus 0.22 : A_1^{\bigcirc})$$

Moving on to the adult population, we observe that while male adults are responsible for choosing the nest and the female, and to hunt, in our model, for the sake of simplicity, we have opted to abstract away from a falcon's gender. We believe that this

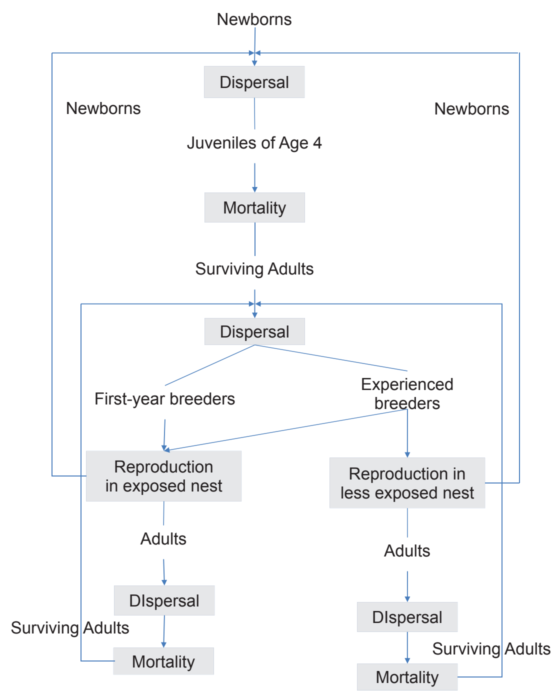

Fig. 1. The life-cycle of Eleonora's Falcon.

simplification does not affect the faithfulness of the model as there is no indication that the percentages of males and females differ significantly, nor that the probability to die during dispersal depends on the gender, and also because adult males and females live in pairs and are considered monogamous.

Thus we model by A the notion of an adult pair. There are two types of such adult pairs: first-year breeders who have no experience in choosing less-exposed nests and second-year or older adult pairs whose experience allows them to select less-exposed nests, if such nests are available [16]. Depending on the nest that a pair chooses, there are different probabilities to have an offspring of size 0,1,2 or 3 during the breeding season. We adopt the reproduction rates from [20] appropriately weighted so that only half of the offspring is produced (to account for pairs). In the model below we write ε_i for the probability that an offspring of size i is produced in an exposed nest and λ_i for the probability that an offspring of size i is produced in a less-exposed nest. Furthermore, we write A_1, A and M for a first-year breeder pair, a mature pair in the phase of reproduction and a mature pair in the phase of possible mortality, respectively. Finally, we use the superscripts $^{\bigcirc}$, $^{\LEFTcircle}$ and $^{\CIRCLE}$ to denote a state of no nest, an exposed nest and a less-exposed nest, respectively.

The behavior of a pair proceeds as follows. A first-year breeder pair, returns to the island. It chooses an exposed nest and proceeds as a mature adult pair in an exposed nest. A mature adult pair selects a less-exposed nest, if one is available (i.e. there is an input available on channel $prey$) and an exposed nest, otherwise. It then produces offspring, leaves the island and goes through a mortality phase. If it survives it executes action $\surd$ and returns to its initial phase. The mortality rate of an adult pair is equal to 13%. Note that, in the mortality phase, a pair in a less-exposed nest releases its nest.

$$
\begin{aligned}
A_1^{\bigcirc} &\stackrel{\text{def}}{=} go\,\ell_1.A^{\CIRCLE}\\
A^{\bigcirc} &\stackrel{\text{def}}{=} \overline{prey}?(A^{\LEFTcircle}, A^{\CIRCLE})\\
A^{\CIRCLE} &\stackrel{\text{def}}{=} \varepsilon_0 : M^{\CIRCLE} \oplus \varepsilon_1 : J_0|M^{\CIRCLE} \oplus \varepsilon_2 : J_0|J_0|M^{\CIRCLE} \oplus \varepsilon_3 : J_0|J_0|J_0|M^{\CIRCLE}\\
A^{\LEFTcircle} &\stackrel{\text{def}}{=} \lambda_0 : M^{\LEFTcircle} \oplus \lambda_1 : J_0|M^{\LEFTcircle} \oplus \lambda_2 : J_0|J_0|M^{\LEFTcircle} \oplus \lambda_3 : J_0|J_0|J_0|M^{\LEFTcircle}\\
M^{\CIRCLE} &\stackrel{\text{def}}{=} go\,\ell_2.(0.13 : 0 \oplus 0.87 : \surd.A^{\bigcirc})\\
M^{\LEFTcircle} &\stackrel{\text{def}}{=} \overline{release}.go\,\ell_2.\,(0.13 : 0 \oplus 0.87 : \surd.A^{\bigcirc})
\end{aligned}
$$

Our system is defined below. It consists of n nests and m adult pairs with no nest.

$$System \stackrel{\text{def}}{=} (LeNest{:}\langle le, l_1, n\rangle | A^{\bigcirc}{:}\langle f, l_1, m\rangle)\backslash\{prey\}$$

5 Analysis in PRISM

In this section, we report on some of the results we obtained by applying our methodology for studying the population dynamics of the Eleonora's falcon. To begin with we translated our PALPS model into PRISM by following the encoding presented in Section 3.

For our experiments, we took advantage of the model checking capabilities of PRISM and we checked properties by using the *model-checking by simulation* option, referred to as *confidence interval (CI)* simulation method. The property we experimented with is `R =?[I = k]`. This property is a reward-based property that computes the average state instant reward at time k. We were interested to study the expected size of the population. For this, we associate to each state a reward representing this size.

We were interested in studying various properties of this model. One of these properties involved assessing the stability of the model for different sizes of the initial population. To achieve this, we considered initial populations of 20, 40 and 60 adult pairs and we studied the growth of the population for a duration of approximately 10 years. These results are reported in Figure 2. The composition of the population in terms of

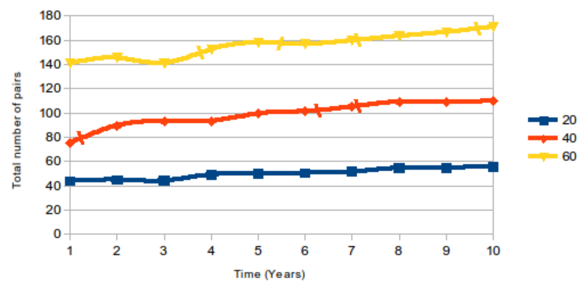

Fig. 2. Expected number of total pairs (juveniles and adults) vs time for an initial population of 20, 40 and 60 pairs of adults.

juveniles and adults and their evolution is presented in figures Figure 3 and Figure 4 .

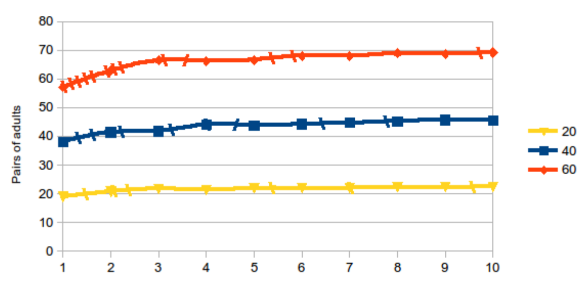

Fig. 3. Expected number of adult pairs vs. time for an initial population of 20, 40 and 60 pairs of adults.

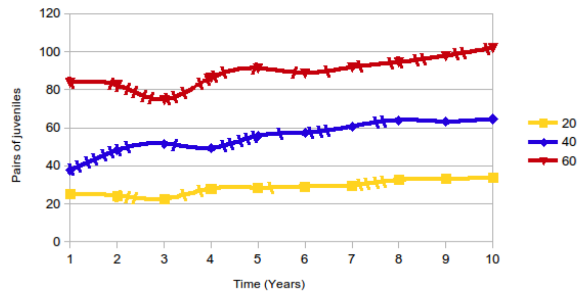

Fig. 4. Expected number of juvenile pairs vs. time for an initial population of 20, 40 and 60 pairs of adults.

A combination of these results for the case of 40 adults papers is summarized in Figure 5.

Another property we were interested to study is the sensitivity of the population to changes in the local conditions. These conditions may affect the probabilities associated with reproduction and, in particular, the survival rate of the offspring of a falcon pair. To study this property we analyzed the impact of changing the reproduction rates in both exposed and less-exposed nests. Specifically, we increased (decreased) the probabilities of 0 fledglings surviving by 3% and 6% while appropriately decreasing (increasing) the probabilities of 1, 2 and 3 fledglings surviving. These results are presented in Figure 6.

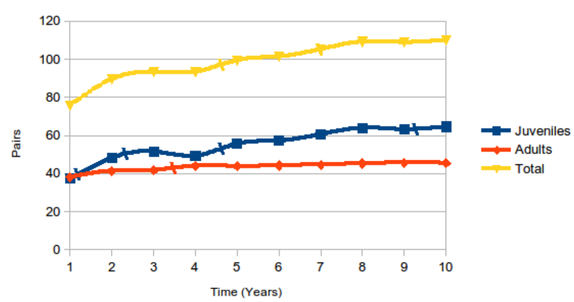

Fig. 5. Expected number of total pairs, juveniles pairs and adult pairs vs time for an initial population of 40 pairs of adults.

We have also conducted similar experiments varying the initial number of juvenile pairs. For the case in which the initial population is conformed by juveniles only, we

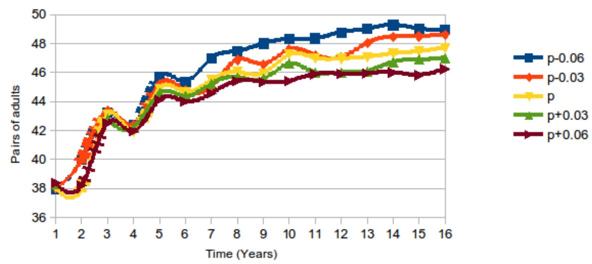

Fig. 6. Expected number of pairs of adults vs time with an initial population of 40 pairs of adults, for different values of the probability p that zero fledglings survive from an offspring of a pair.

made an analysis to determine the impact on the evolution of the colony through time for different values of the mortality rates in Figure 7.

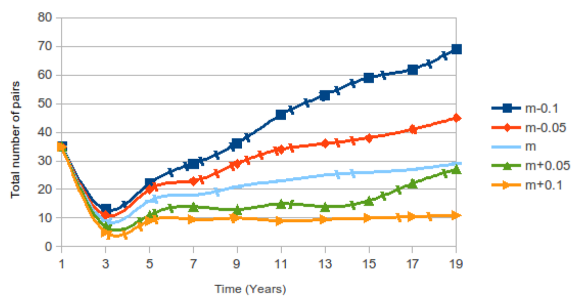

Fig. 7. Expected total number of pairs vs. time with an initial population of 40 juveniles, for different values of the mortality rates. The original values of mortality are $m_1 = 0.78$ for juveniles and $m_2 = 0.13$ for adults. As an example, legend $m + 0.1$ in the graph means that both values m_1 and m_2 were incremented by 0.1.

Overall, our experiments have demonstrated a fair degree of stability in the evolution of the species and a relative insensitivity to the local conditions on the island (Figure 6).

6 Conclusions

In this paper we have presented S-PALPS, an extension of PALPS with synchronous parallel composition. Furthermore, we have described a translation of S-PALPS into the

PRISM language and we have proved its correctness. This encoding can be employed for simulating and model checking S-PALPS systems using the PRISM tool. Furthermore, we have applied our methodology for studying the population dynamics of the Eleonora's falcon, a species of local conversation interest in the Mediterranean sea.

We have observed that the adoption of a synchronous parallel composition in S-PALPS enables a more succinct presentation of the state space of a system by removing a lot of redundant nondeterminism that was present in the asynchronous framework. Furthermore, the treatment of the multiplicity of individuals in S-PALPS was very beneficial by allowing a more efficient translation of populations (as opposed to individuals) into PRISM modules and removing restrictions that were present in our previous work. These benefits were apparent while carrying out our case study of the Eleonora's falcon. Our experiments towards studying the population dynamics of this species revealed a fair degree of stability in the evolution of the species and a relative insensitivity to small changes in the local conditions.

As future work, we are interested in applying our methodology to other case studies from the local habitat and, in particular, to employ model checking for studying their behavior. Finally, an interesting future research direction would be extend the work of [9] towards the development of mean-field analysis to represent the average behavior of systems within a spatially-explicit framework.

References

1. Online PRISM documentation. `http://www.prismmodelchecker.org/doc/`.
2. R. Barbuti, P. Bove, A. M. Schettini, P. Milazzo, and G. Pardini. A computational formal model of the invasiveness of eastern species in European water frog populations. In *Proceedings of MOKMASD'13*, LNCS 8368, pages 207–218. Springer, 2013.
3. T. A. Basuki, A. Cerone, R. Barbuti, A. Maggiolo-Schettini, P. Milazzo, and E. Rossi. Modelling the dynamics of an aedes albopictus population. In *Proceedings of AMCA-POP'10*, volume 33 of *EPTCS*, pages 18–36, 2010.
4. D. Besozzi, P. Cazzaniga, D. Pescini, and G. Mauri. Modelling metapopulations with stochastic membrane systems. *BioSystems*, 91(3):499–514, 2008.
5. M. Cardona, M. A. Colomer, A. Margalida, A. Palau, I. Pérez-Hurtado, M. J. Pérez-Jiménez, and D. Sanuy. A computational modeling for real ecosystems based on P systems. *Natural Computing*, 10(1):39–53, 2011.
6. S. C. Fu and G. Milne. A Flexible Automata Model for Disease Simulation. In *Proceedings of ACRI'04*, LNCS 3305, pages 642–649. Springer, 2004.
7. C. Kassara, J. Fric, M. Gschweng, and S. Sftenthourakis. Complementing the puzzle of Eleonora's falcon (Falco eleonorae) migration: new evidence from an eastern colony in the Aegean Sea. *Journal of Ornithology*, 153:839848, 2012.
8. C. McCaig, R. Norman, and C. Shankland. Process Algebra Models of Population Dynamics. In *Proceedings of AB'08*, LNCS 5147, pages 139–155. Springer, 2008.
9. C. McCaig, R. Norman, and C. Shankland. From individuals to populations: A mean field semantics for process algebra. *Theoretical Computer Science*, 412(17):1557–1580, 2011.
10. R. Milner. *A Calculus of Communicating Systems*. Springer, 1980.
11. D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Dynamical probabilistic P- systems. *Journal of Foundations of Computer Science*, 17(1):183–204, 2006.

12. A. Philippou and M. Toro. Process Ordering in a Process Calculus for Spatially-Explicit Ecological Models. In *Proceedings of MOKMASD'13*, LNCS 8368, pages 345–361. Springer, 2013.
13. A. Philippou, M. Toro, and M. Antonaki. Simulation and Verification for a Process Calculus for Spatially-Explicit Ecological Models. *Scientific Annals of Computer Science*, 23(1):119–167, 2013.
14. G. Păun. *Membrane Computing: An Introduction*. Springer-Verlag, 2002.
15. D. Ristow, W. Scharlau, and M. Wink. Population structure and mortality of Eleonora's falcon Falco eleonorae. *Raptors in the Modern World Working Group on Birds of Prey*, pages 321–326, 1989.
16. D. Ristow, M. Wink, C. Wink, and H. Friemann. Biologie des Eleonorenfalken (Falco eleonorae). *Das Brutreifealter der Weibchen*, 124:291–293, 1983.
17. F. J. Romero-Campero, M. Gheorghe, L. Bianco, D. Pescini, M. Pérez-Jiménez, and R. Ceterchi. Towards Probabilistic Model Checking on P Systems Using PRISM. In *Membrane Computing*, volume 4361 of *LNCS*, pages 477–495. Springer, 2006.
18. D. J. T. Sumpter, G. B. Blanchard, and D. S. Broomhear. Ants and Agents: a Process Algebra Approach to Modelling Ant Colony Behaviour. *Bulletin of Mathematical Biology*, 63:951–980, 2001.
19. C. Tofts. Processes with probabilities, priority and time. *Formal Aspects of Computing*, 6:536–564, 1994.
20. S. Xirouchakis, J. Fric, C. Kassara, D. Portolou, A. Dimalexis, G. Karris, C. Barboutis, P. Latsoudis, S. Bourdakis, E. Kakalis, and S. Sftenthourakis. Variation in breeding parameters of Eleonora's falcon (Falco eleonorae) and factors affecting its reproductive performance. *Ecological Research*, 27:407–416, 2012.