4
Propositional Dynamic Logic for Stochastic Petri Nets

Probabilistic algorithms are often present and deal with important problems in computer science (Fallis, 2000; Motwani & Raghavan, 1996). In this chapter we present Stochastic Petri Nets (Haas, 2002; Lyon, 1995; Marsan, 1990*b*; Marsan & Chiola, 1987*b*) and propose the $\mathcal{DS}_3$ logic as a stochastic approach for Petri-PDL. This model of Petri Nets is widely used for non-linear time-modelling (Coleman et al., 1996; Henderson et al., 2009; Marin et al., 2012).

There are some other well-known stochastic approaches to PDL, but all of them have some disadvantages. For instance, there is the system P-Pr(DL) (Feldman, 1983, 1984) which has no finite axiomatization, do not allow boolean combination of propositional variables and is defined only for regular programs. There is also Pr(DL) (Feldman & Harel, 1984) which has the same limitations as P-Pr(DL) and is undecidable. The system PPDL (Kozen, 1983) computes the probability of a proposition being true in some state but the program is replaced by a measurable function. The logic PPDL$> r$ (Tiomkin & Makowsky, 1985) can only describe situations where some probability is greater then a constant $r \in \mathbb{R}$ and PPDL> 0 (Tiomkin & Makowsky, 1991) that can only describe situations where some probability is greater then zero.

The approach proposed in here aims to provide a compositional way to deal with Stochastic Petri Nets similarly to Petri-PDL where the user is able to verify if the probability of a firing is greater than zero or if it is equal to one. The user takes advantage of the intuitive graphical interpretation of Stochastic Petri Nets that simplifies the modelling process.

4.1
A stochastic approach for Petri Nets

A Stochastic Petri Net (SPN) (Haas, 2002; Lyon, 1995; Marsan, 1990*b*; Marsan & Chiola, 1987*b*) is a 5-tuple $\mathcal{P} = \langle P, T, L, M_0, \Lambda \rangle$, where P is a finite set of places, T is a finite set of transitions, with $P \cap T = \emptyset$, and $P \cup T \neq \emptyset$ and L is a function which defines directed edges between places and transitions and assigns a multiplicative weight $w \in \mathbb{N}$ to the transition, that is

$L\colon (P \times T) \cup (T \times P) \to \mathbb{N}$ (in this work we assume $w = 1$ for all edges), M_0 is the initial markup and $\Lambda = \lambda_1, \lambda_2, \ldots, \lambda_n$ the firing rates of each transition.

In an SPN the firing of a transition is determined by the markups and by the firing rate. For each transition $t_i \in T$ is associated a unique random variable with an exponential distribution with parameter $\lambda_i \in \Lambda$.

In the initial markup (M_0) each transition gets a firing delay through an occurrence of the random variable associated to it. Each firing delay is marking-dependent and the transition $t_i \in T$ firing rate at marking M_j is defined as $\lambda_i(M_j)$ and its average firing delay is $[\lambda_i(M_j)]^{-1}$. After a firing, each previously non-marking-enabled transition gets a new firing delay by sampling its associated random variable. A transition previously marking-enabled that keeps marking-enabled has its firing delay decreased in a constant speed. When a transition firing delay reaches zero, this transition fires.

We define the preset of $t \in T$, denoted by ${}^\bullet t$, as the set of all $s_k \in S$ that origins an edge to t. The postset of t, denoted by $t^\bullet$ is defined as the set of all $s_\ell \in S$ that t origins an edge to. We say that a transition t is enabled if, and only if, there is at least one token in each place $p \in {}^\bullet t$.

Given a markup M_j of a Petri Net, a transition t_i is enabled on M_j if and only if $\forall x \in {}^\bullet t_i, M_j(x) \geq 1$ and $\lambda_i(M_j) = \min(\lambda_1(M_j), \lambda_2(M_j), \ldots, \lambda_n(M_j))$, where ${}^\bullet t_i$ is the preset of t_i, that is, a transition t_i is enabled if and only if there is at least a token in each place of its preset and its timing is the smallest of the SPN. A new markup generated by setting a transition which is enabled is defined in the same way as in a Marked Petri Net, i.e.

$$M_{j+1}(x) = \begin{cases} M_j(x) - 1 & \forall x \in {}^\bullet t \setminus t^\bullet \\ M_j(x) + 1 & \forall x \in t^\bullet \setminus {}^\bullet t \\ M_j(x) & \text{otherwise} \end{cases} . \tag{4-1}$$

A new firing delay for a transition t_i for a markup M_j is defined as:

(i) if t_i fires then a new occurrence of the random variable associated with it is the new firing delay;

(ii) if t_i was disabled and has just been enabled then a new occurrence of the random variable associated with it is the new firing delay;

(iii) otherwise, the value of the firing delay of t_i must be decreased.

That is:

$$\lambda_i(M_{j+1}) \begin{cases} = \text{new}_e(\lambda_i) & \text{if} \begin{cases} \begin{cases} \forall x \in {}^\bullet t_i, M_j(x) \geq 1 \\ \lambda_i(M_j) \leq \min(\lambda_1(M_j), \ldots, \lambda_n(M_j)) \end{cases} \\ \text{or} \\ \begin{cases} \exists x \in {}^\bullet t_i, M_j(x) < 1 \\ \forall x \in {}^\bullet t_i, M_{j+1}(x) \geq 1 \end{cases} \end{cases} \\ < \lambda_i(M_j) & \text{otherwise} \end{cases} \tag{4-2}$$

where $\text{new}_e(\lambda)$ denotes a new occurrence of the random variable exponentially distributed with parameter λ associated to t_i.

The minimum of two random variables with parameters, respectively, λ_1 and λ_2, is a random variable with exponential distribution of parameter $\lambda_1 + \lambda_2$. The sojourn time in a marking M_j is a random variable exponentially distributed with mean

$$\left[\sum_{i:\forall k \in {}^\bullet t_i, M_j(k)>0} \lambda_i(M_j) \right]^{-1}. \tag{4-3}$$

As all random variables have an exponential distribution, then it is possible to compute the probability of an enabled transition t_i which has the minimum firing delay (i.e. the probability of t_i fires immediately) at a marking M_j:

$$\Pr(t_i \mid M_j) = \frac{\lambda_i(M_j)}{\sum_{k:\forall \ell \in {}^\bullet t_k, M_j(\ell)>0} \lambda_k(M_j)}. \tag{4-4}$$

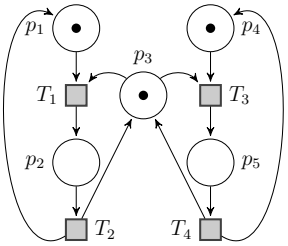
The long-run time of availability of a resource (Haas, 2002) can be computed by the limit

$$r(h) = \lim_{t \to \infty} \frac{1}{t} \int_0^t h(M_u) du \tag{4-5}$$

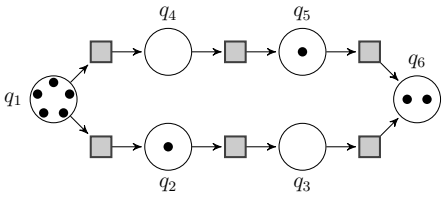
where $f \colon M \to \mathbb{R}$ is a function which describes the availability of a resource in a Stochastic Petri Net.

To illustrate the usage of Stochastic Petri Nets, we can model a two processes system that share a resource. Process 1 is I/O bound and process 2 is CPU bound, as in Figure 4.1(a). The great difference in the amount of requests of input can be modelled by setting the Λ values (i.e. $\lambda_1 > \lambda_3$). Figure 4.1(b) presents a simple parallel system modelled in a SPN where the tokens denote processes. The Λ values determines if it would be faster in some of the ways. The probability of a process goes from q_1 to q_2 instead of to q_4 can be computed according to equation (4-4).

As pointed out in the work of Mazurkiewicz (1987, 1989), logics that deal with Petri Nets use to be incomplete due to the possibility of a place always increase its token amount (up to countable infinity). To restrict a subset of Petri Nets where we can achieve decidability and completeness, we call



4.1(a): A two processes system



4.1(b): A simple parallel system

Figure 4.1: Stochastic Petri Net examples

normalised Stochastic Petri Net any Stochastic Petri Net that do not contain any place which can accumulate an infinite amount of tokens. From now on, all the proofs deal only with normalised Stochastic Petri Nets.

4.2 A stochastic approach for Petri-PDL

A stochastic approach for Petri-PDL (Section 3.1) is presented here as the logic system $\mathcal{DS}_3$ (Propositional Dynamic Logic for Stochastic Petri Nets).

The language of $\mathcal{DS}_3$ is the same language of Petri-PDL. The difference is that the Petri Net program is replaced by a Stochastic Petri Net program (more details on how deal with its behaviour given in the frame Definition 60, where a program is defined as a pair of transitions and the parameters of the exponential random variables associated with them). The language of $\mathcal{DS}_3$ is the same language than Petri-PDL.

Definition 58 $\mathcal{DS}_3$ program

A $\mathcal{DS}_3$ program is a pair (Π, Λ) where Π is a composition of transitions defined as (we use s a sequence of names – the markup of Π). The transitions may be from three types, $T_1 : xt_1y$, $T_2 : xyt_2z$ and $T_3 : xt_3yz$, each transition has a unique type.

Basic programs: $\pi_b ::= at_1b \mid at_2bc \mid abt_3c$ *where* t_i *is of type* $T_i, i = 1, 2, 3$

Stochastic Petri Net Programs: $\pi ::= s, \pi_b \mid \pi \odot \pi$

$\Lambda(\pi) = \langle \lambda_1, \lambda_2, \ldots \lambda_n \rangle$ *is a function that associates a positive real value with each basic transition* $\pi \in \Pi$ *where* $\pi = \pi_1 \odot \pi_2 \odot \cdots \pi_n$. *The function* Λ *denotes the value of the parameter of the exponential random variable associated with each transition.*

Definition 59 $\mathcal{DS}_3$ formula

A $\mathcal{DS}_3$ formula is defined as

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle s, \pi\rangle\varphi.$$

We use the standard abbreviations $\bot \equiv \neg\top$, $\varphi \vee \phi \equiv \neg(\neg\varphi \wedge \neg\phi)$, $\varphi \rightarrow \phi \equiv \neg(\varphi \wedge \neg\phi)$ and $[s, \pi]\varphi \equiv \neg\langle s, \pi\rangle\neg\varphi$, and π is a Stochastic Petri Net program.

The firing of a transition in $\mathcal{DS}_3$ is defined according the firing function in Definition 25.

Definition 60 $\mathcal{DS}_3$ frame

A frame for $\mathcal{DS}_3$ is a 4-tuple $\mathcal{F}_3 = \langle W, R_\pi, M, (\Pi, \Lambda), \delta\rangle$ where

- *W is a non-empty set of states*
- *$M\colon W \rightarrow S$*
- *(Π, Λ) is a Stochastic Petri Net program*
- *$\delta(w, \pi) = \langle d_1, d_2, \ldots, d_n\rangle$ is the sequence of firing delays of the program $\pi \in \Pi$ in the world $w \in W$ respectively for each program $\pi_1 \odot \pi_2 \odot \cdots \odot \pi_n = \pi$, satisfying the following conditions (where $s = M(w)$ and $r = M(v)$)*
  - *if $wR_{\pi_b}v$ and $f(r, \pi_b) = \epsilon$ then $\delta(w, \pi_b) = \delta(v, \pi_b)$*
  - *if $f(s, \pi_b) = \epsilon$, $f(r, \pi_b) \neq \epsilon$ and $wR_{\pi_b}v$, then $\delta(v, \pi_b)$ is an occurrence of a random variable of exponential distribution with parameter $\Lambda(\pi_b)$, i.e., by the inversion theorem, $\delta(v, \pi_b) = \frac{\ln(1-u)}{-\Lambda(\pi_b)}$ where u is an occurrence of a uniform random variable*
  - *if $f(s, \pi_b) \neq \epsilon$, $f(r, \pi_b) \neq \epsilon$ and $wR_{\pi_b}v$, $\delta(v, \pi_b) < \delta(w, \pi_b)$*
- *R_η is a binary relation over W, for each basic program $\eta \in \pi_b$, satisfying the following conditions (where $s = M(w)$)*
  - *if $f(s, \eta) \neq \epsilon$ and $\delta(w, \eta) = \min(\delta(w, \Pi))$, then $wR_\eta v$ iff $f(s, \eta) \prec M(v)$*
  - *if $f(s, \eta) = \epsilon$ or $\delta(w, \eta) \neq \min(\delta(w, \Pi))$, then $wR_\eta v$ iff $w = v$*
- *we inductively define the binary relation R_η, for each Stochastic Petri Net program as follows*
  $\eta = \eta_1 \odot \eta_2 \odot \cdots \odot \eta_n$, as $R_\eta = \{(w, v) \mid \exists\eta_i, \exists u$ such that $s_i \prec M(u)$ and $wR_{\eta_i}u$ and $\delta(w, \eta_i) = \min(\delta(w, \Pi))$ and $uR_\eta v\}$ where $s_i = f(s, \eta_i)$, for all $1 \leq i \leq n$.

Lemma 61 *Reflexivity over empty occurrences*

For any Petri Net program π, $f(\epsilon, \pi) = \epsilon$, $R_{\epsilon,\pi}$ is reflexive.

Proof: This proof is straightforward from the firing function definition (Definition 25) and frame Definition 60. ■

Definition 62 $\mathcal{DS}_3$ model

A model for $\mathcal{DS}_3$ is a pair $\mathcal{M}_3 = \langle \mathcal{F}_3, \mathbf{V} \rangle$, where $\mathcal{F}_3$ is a $\mathcal{DS}_3$ frame and $\mathbf{V}$ is a valuation function $\mathbf{V} \colon \Phi \to 2^W$.

Lemma 63 *Truth Probability of a Modality*

The probability of $\mathcal{M}_3$,$w \Vdash \langle s, \pi_b \rangle \varphi$ is given by (where $s = M(w)$)

$$\Pr(\mathcal{M}_3,w \Vdash \langle s, \pi_b \rangle \varphi \mid \delta(w, \Pi)) = \frac{\delta(w, \pi_b)}{\sum\limits_{\pi_b \in \Pi : f(s,\pi_b) \neq \epsilon} \delta(w, \pi_b)}$$

Proof: This proof is straightforward from relation (4-4) *and Definition 60.* ■

Definition 64 Semantic notion of $\mathcal{DS}_3$

Let $\mathcal{M}_3$ be a model for $\mathcal{DS}_3$. The notion of satisfaction of a formula φ in $\mathcal{M}_3$ at a state w, namely $\mathcal{M}_3$,$w \Vdash \varphi$ is inductively defined as follows.

- $\mathcal{M}_3$,$w \Vdash p$ *iff* $w \in \mathbf{V}(p)$
- $\mathcal{M}_3$,$w \Vdash \top$
- $\mathcal{M}_3$,$w \Vdash \neg\varphi$ *iff* $\mathcal{M}_3$,$w \nVdash \varphi$
- $\mathcal{M}_3$,$w \Vdash \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{M}_3$,$w \Vdash \varphi_1$ *and* $\mathcal{M}_3$,$w \Vdash \varphi_2$
- $\mathcal{M}_3$,$w \Vdash \langle s, \eta \rangle \varphi$ *if there exists* $v \in W$, $wR_\eta v$ *and* $\Pr(\mathcal{M}_3,v \Vdash \langle s, \eta_b \rangle \varphi \mid \delta(v, \Pi)) > 0$

If φ is satisfied in all states of $\mathcal{M}_3$ then φ is satisfied in $\mathcal{M}_3$, namely $\mathcal{M}_3 \Vdash \varphi$; and if φ is valid in any model then φ is valid, namely $\Vdash \varphi$.

4.3 Axiomatic system

We consider the following set of axioms and rules, where p and q are proposition symbols, φ and ψ are formulae, $\eta = \eta_1 \odot \eta_2 \odot \cdots \odot \eta_n$ is a Petri Net program and π is a Marked Petri Net program.

(PL) Enough propositional logic tautologies

(K) $[s,\pi](p \to q) \to ([s,\pi]p \to [s,\pi]q)$

(Du) $[s,\pi]p \leftrightarrow \neg\langle s,\pi\rangle\neg p$

($\mathbf{PC}_3$) $\langle s,\eta\rangle\varphi \leftrightarrow \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi$, where $s_i = f(s,\eta_i)$, for all $1 \leq i \leq n$ and π is not a basic program

($\mathbf{R}_{3_\epsilon}$) $\langle s,\eta\rangle\varphi \leftrightarrow \varphi$, if $f(s,\eta) = \epsilon$

(Sub) If $\Vdash \varphi$, then $\Vdash \varphi^\sigma$, where σ uniformly substitutes proposition symbols by arbitrary formulae

(MP) If $\Vdash \varphi$ and $\Vdash \varphi \to \psi$, then $\Vdash \psi$

(Gen) If $\Vdash \varphi$, then $\Vdash [s,\pi]\varphi$

4.4 Soundness and completeness

In this section we use some well-known results for Stochastic Petri Nets (Haas, 2002) to fulfil the requirements of soundness and completeness. The axioms **(PL)**, **(K)** and **(Du)** and the rules **(Sub)**, **(MP)** and **(Gen)** are standard in the modal logic literature.

Lemma 65 *Validity of $\mathcal{DS}_3$ axioms*

1. $\Vdash \mathbf{PC}_3$

   Proof: Supose that there is a world w from a model $\mathcal{M}_3 = \langle W, R_\eta, M, (\Pi,\Lambda), \delta, \mathbf{V}\rangle$ where PC_3 is false. For PC_3 to be false in w, there are two cases:

   (a) *Suppose $\mathcal{M}_3$,$w \Vdash \langle s,\eta\rangle\varphi$ (1) and $\mathcal{M}_3$,$w \nVdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi$ (2). By (1) iff there is a world v such that $wR_\eta v$ and $\Pr(\mathcal{M}_3,v \Vdash \langle s,\eta_b\rangle\varphi \mid \delta(v,\Pi)) > 0$ (3). By Definition 60 $R_\eta = \{(w,v) \mid \exists\eta_i, \exists u$ such that $s_i \prec$*

$M(u)$ and $wR_{\eta_i}u$ and $\delta(w,\eta_i) = \min(\delta(w,\Pi))$ and $uR_\eta v\}$,
from (3) $\mathcal{M}_3$,$u \Vdash \langle s_i,\eta\rangle\varphi$ *and* $\mathcal{M}_3$,$w \Vdash \langle s,\eta_i\rangle\langle s_i,\eta\rangle\varphi$, *which implies that* $\Pr(\mathcal{M}_3,w \Vdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi \mid \delta(w,\Pi)) > 0$ *(4).*
From (4) $\mathcal{M}_3$,$w \Vdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi$, *which contradicts (2).*

(b) *Suppose*
$\mathcal{M}_3$,$w \Vdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi$ *(2), then, by semantics of disjunction, for some* i $(1 \leq i \leq n)$, *we have that* $M,w \Vdash \langle s,\eta_i\rangle\langle s_i,\eta\rangle\varphi$ *iff*
there is a u *such that* $wR_{\eta_i}u$, $\Pr(\mathcal{M}_3,w \Vdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi \mid \delta(w,\Pi)) > 0$ *(3)*
iff there is a v *such that* $uR_\eta v$ *and* $\mathcal{M}_3$,$v \Vdash \varphi$ *(4).*
By Definition 60, (3) and (4) we have $wR_\eta v$, $\Pr(\mathcal{M}_3,w \Vdash \langle s,\eta_1\rangle\langle s_1,\eta\rangle\varphi \vee \langle s,\eta_2\rangle\langle s_2,\eta\rangle\varphi \vee \cdots \vee \langle s,\eta_n\rangle\langle s_n,\eta\rangle\varphi \mid \delta(w,\Pi)) > 0$ *and* $\mathcal{M}_3$,$v \Vdash \varphi$. *Thus,* $\mathcal{M}_3$,$w \Vdash \langle s,\eta\rangle\varphi$.

So, PC_3 *is valid.* ■

2. $\Vdash \mathbf{R}_{3_\epsilon}$

Proof: Suppose that there is a world w *from a model* $\mathcal{M}_3 = \langle W, R_\eta, M, (\Pi,\Lambda), \delta, \mathbf{V}\rangle$ *where* R_ϵ *is false. For* R_ϵ *to be false in* w, *there are two cases:*

(a) *Suppose* $\mathcal{M}_3$,$w \Vdash \langle\epsilon,\eta\rangle\varphi$ *(1) and*
$\mathcal{M}_3$,$w \nVdash \varphi$ *(2)*
(1) iff there is a v *such that* $wR_{\epsilon,\eta}v$ *and* $\Pr(\mathcal{M}_3, v \Vdash \langle\epsilon,\eta\rangle\varphi \mid \delta(v,\Pi)) > 0$.
As $f(\epsilon,\eta) = \epsilon$, *by Definition 61,* $w = v$, $wR_\eta w$ *and* $\mathcal{M}_3, w \Vdash \varphi$, *which contradicts (2).*

(b) *Suppose* $\mathcal{M}_3$,$w \nVdash \langle\epsilon,\eta\rangle\varphi$ *(1) and*
$\mathcal{M}_3$,$w \Vdash \varphi$ *(2). (1) iff* $\Pr(\mathcal{M}_3$,$w \Vdash \langle\epsilon,\eta\rangle\varphi \mid \delta(w,\Pi)) = 0$.
As $f(\epsilon,\eta) = \epsilon$, *by Definition 61,* $wR_\eta w$ *and, by Definition 60,* M_3,$w \nVdash \varphi$, *which contradicts (2).*

So, R_{3_ϵ} *is valid.* ■

As for Petri-PDL, the completeness proof goes as in the work of Blackburn et al. (2001); Harel et al. (2000) and Goldblatt (1992*b*).

Definition 66 Canonic Model

A canonic model for $\mathcal{DS}_3$ with language $\mathcal{L}$ is a 5-tuple $\mathcal{C}_3^{\mathcal{L}} = \langle W_3^{\mathcal{L}}, R_3^{\mathcal{L}}, M_3^{\mathcal{L}}, (\Pi_3^{\mathcal{L}}, \Lambda_3^{\mathcal{L}}), \delta_3^{\mathcal{L}}, \mathbf{V}_3^{\mathcal{L}} \rangle$, where $W_3^{\mathcal{L}}$ is the set of all maximal consistent sets; $\mathbf{V}_3^{\mathcal{L}}$ is a valuation function where for all $w \in W_3^{\mathcal{L}}$, $w \in \mathbf{V}_3^{\mathcal{L}}(\varphi)$ iff $\varphi \in w$; $(\Pi_3^{\mathcal{L}}, \Lambda_3^{\mathcal{L}})$ is a Stochastic Petri Net program whose firing delays for each world are defined by $\delta_3^{\mathcal{L}} \colon W_3^{\mathcal{L}} \times \Pi_3^{\mathcal{L}} \to \vec{\mathbb{R}}^+$; $M_3^{\mathcal{L}}$ is the markup of the Petri Net programs, defined as

$$M_3^{\mathcal{L}}(w) = \{s_1, \ldots, s_n \mid \langle s_i, \pi \rangle \varphi \in w, 1 \leq i \leq n, w \in W_3^{\mathcal{L}}\};$$

and $R_3^{\mathcal{L}}$ is a binary relation between the elements of $W_3^{\mathcal{L}}$ defined for each program $\pi \in \Pi$ as

$$R_{3_\pi}^{\mathcal{L}} = \{(n, m) \mid n,m \in W_3^{\mathcal{L}}, \{\varphi / [s, \pi]\varphi \in n, s \prec M_3^{\mathcal{L}}(n)\} \subseteq m\}.$$

Lemma 67 *$\mathcal{C}_3^{\mathcal{L}}$ is a canonic model for $\mathcal{DS}_3$*
Proof: By Definition 66 we have that:

- *$W_3^{\mathcal{L}}$ is finite a set of states.*
- *$M_3^{\mathcal{L}} \colon W_3^{\mathcal{L}} \to S$.*
- *$R_{3_\pi}^{\mathcal{L}} = \{(w, v) \mid$ for some π_i $\exists u$ such that $s_i \prec M^{\mathcal{L}}(u)$ and $w R_{\pi_i} u$ and $u R_\pi v\}$ for any program $\pi = \pi_1 \odot \cdots \odot \pi_n$, where $s_i = f(s, \pi_i)$ and $1 \leq i \leq n$.*
- *$(\Pi_3^{\mathcal{L}}, \Lambda_3^{\mathcal{L}})$ is a Stochastic Petri Net program where Π is a composition of transitions and $\Lambda_3^{\mathcal{L}} \colon \Pi_3^{\mathcal{L}} \to \vec{\mathbb{R}}^+$ is a function that associates the parameter of the exponential random variable associated with each transition of Π.*
- *$\delta_3^{\mathcal{L}} \colon W_3^{\mathcal{L}} \times \Pi_3^{\mathcal{L}} \to \vec{\mathbb{R}}^+$ such that (let $\pi_b \in \Pi_3^{\mathcal{L}}$, $w \in W_3^{\mathcal{L}}$, $v \in W_3^{\mathcal{L}}$, $s = M\mathcal{L}_3(w)$ and $r = M\mathcal{L}_3(v)$)*
  - *if $w R_{3_{\pi_b}}^{\mathcal{L}} v$ then $f(r, \pi_b) = \epsilon$ and $\delta_3^{\mathcal{L}}(w, \pi_b) = \delta_3^{\mathcal{L}}(v, \pi_b)$ (once there was no firing);*
  - *if $f(s, \pi_b) = \epsilon$, $f(r, \pi_b) \neq \epsilon$ and $w R_{3_{\pi_b}}^{\mathcal{L}} v$, $\delta^{\mathcal{L}}(v, \pi_b)$ is an occurrence of a random variable of exponential distribution with parameter $\Lambda^{\mathcal{L}}(\pi_b)$*
  - *if $f(s, \pi_b) \neq \epsilon$ then $f(r, \pi_b) \neq \epsilon$ and $w R_{3_{\pi_b}}^{\mathcal{L}} v$, $\delta^{\mathcal{L}}(v, \pi_b) < \delta^{\mathcal{L}}(w, \pi_b)$*

So, $\mathcal{C}_3^{\mathcal{L}}$ is a model for $\mathcal{DS}_3$. ■

Lemma 68 *Let $\mathcal{C}_3^{\mathcal{L}}$ a Canonic Model for $\mathcal{DS}_3$ as in Definition 66. Then, for any $w \in W_3^{\mathcal{L}}$, $w \Vdash \varphi$ iff $\varphi \in w$.*
Proof:

1. *If φ is an atomic formula, it holds by the definition of $\mathbf{V}_3^{\mathcal{L}}$.*

2. *If φ is $\neg\phi$ then $w \Vdash \varphi$ iff $w \nVdash \phi$.*

3. *If φ is in the form $\phi_1 \wedge \phi_2$ then, as w is maximal consistent and by the inductive hypothesis, $w \Vdash \varphi$ iff $w \Vdash \phi_1$ and $w \Vdash \phi_2$, and $w \in \mathbf{V}_3^{\mathcal{L}}(\phi_1)$ and $w \in \mathbf{V}_3^{\mathcal{L}}(\phi_2)$.*

4. *If φ is a modal formula with a Stochastic Petri Net program such as $\langle s, \pi\rangle\phi$, then, $w \Vdash \varphi$ iff:*

   ($\mathbf{R}_{3_\epsilon}$): *$f(s,\pi) = \epsilon$, $wR_{3\,\pi}^{\mathcal{L}}w$, $w \Vdash \phi$ and $w \in \mathbf{V}_3^{\mathcal{L}}(\phi)$ by the inductive hypothesis;*

   ($\mathbf{PC}_3$): *$\exists u \exists v, wR_{3_{\pi_i}}^{\mathcal{L}}u$, $uR_{3_\pi}^{\mathcal{L}}v$, $1 \leq i \leq n$, $v \Vdash \phi$ and $v \in \mathbf{V}_3^{\mathcal{L}}(\phi)$ by the inductive hypothesis and by Lemma 69, where $\pi = \pi_1 \odot \cdots \odot \pi_n$;*

So, this lemma is valid. ■

Lemma 69 *$[s,\pi]\varphi \in u$ iff in all v such that $uR_3^{\mathcal{L}}v$, $\varphi \in v$.*
Proof:

Supose $[s,\pi]\varphi \in u$ and there is no $v \in W_3^{\mathcal{L}}$ such that $uR_3^{\mathcal{L}}v$ and $\varphi \in v$ (1).

As $\pi = \pi_1 \odot \cdots \odot \pi_n$, by $R_3^{\mathcal{L}}$ we have that all $[s,\pi_i][s_i,\pi]\varphi, 1 \leq i \leq n \in v$ for all $1 \leq i \leq n$ if $uR_3^{\mathcal{L}}v$ (2).

By (PC), all $[s,\pi_i][s_i,\pi]\varphi, 1 \leq i \leq n \in u$ for all $1 \leq i \leq n$ (3).

But if (3) then φ is in some v such that $uR_3^{\mathcal{L}}v$ by $R_3^{\mathcal{L}}$ definition, which contradicts (1).

So, in all v such that $uR_3^{\mathcal{L}}v$, $\varphi \in v$.

Supose that exists some $u \in W_3^{\mathcal{L}}$ where $[s,\pi]\varphi \notin u$ and such that in all v that $uR_3^{\mathcal{L}}v$, $\varphi \in v$ (4).

By $R_3^{\mathcal{L}}$ definition, if in all v that $uR_3^{\mathcal{L}}v$, $\varphi \in v$, then $[s,\pi]\varphi \in u$ (5).

Then, there is a contradiction.

So, this lemma is valid. ■

Lemma 70 *If $\varphi \in w$ for all w maximal consistent set of formulae, then $\Vdash \varphi$.*
Proof: Suppose $\nVdash \varphi$; then, by Lemma 68, $\neg\varphi \in w$. But if $\varphi \in w$ and $\neg\varphi \in w$ then there is a contradiction. ■

Theorem 71 *Completeness*

If $\Vdash \varphi$ then $\vdash \varphi$.

Proof: If φ is valid then it is valid in all models, including the canonic model. So it is valid in all worlds of $\mathcal{C}_3^{\mathcal{L}}$ (all maximal consistent sets). So by Lemma 70, φ is derivable. Therefore if $\Vdash \varphi$, then $\vdash \varphi$. ■

Definition 72 The Fischer-Ladner closure

It is inductively defined as follows, where $FL(\varphi)$ denotes the smallest set containing φ which is closed under sub formulae.

$FL\colon \Upsilon \to 2^{\Upsilon}$, where Υ is the set of all formulae

1. *$FL(\varphi)$ is closed under subformulae;*
2. *if $\langle s,\eta\rangle\psi \in FL(\varphi)$, then $\Pr(\mathcal{M}_3, v \Vdash \langle s,\eta_b\rangle\psi \mid \delta(v,\Pi)) > 0$, where η_b is a basic program of η, so $\langle s,\eta_i\rangle\langle s_i,\eta\rangle\psi \in FL(\varphi)$,*
   where $\eta = \eta_1 \odot \eta_2 \odot \cdots \odot \eta_n$ and $s_i = f(s,\eta_i)$, for all $1 \le i \le n$;

Lemma 73 *$FL(\varphi)$ is finite.*

Proof: The only possibility of construct $\varphi \succ \sigma$ (i.e. σ is a derivative of the formula φ) is iff φ is in the form $\langle s,\pi\rangle\Psi$ and σ is in the form $\langle s,\pi_i\rangle\langle s_i,\pi\rangle\Psi$ for some $1 \le i \le n$ where n is the size of the Stochastic Petri Net program π (i.e. the number of basic programs of π) and π_i is an atomic program. Then the smallest closed set Γ containing a formula ρ is obtained by closing $FL(\rho)$ under $\succ$; hence $\phi \in \Gamma$ iff there is a finite sequence of the form $\varphi = \varphi_1 \succ \cdots \succ \varphi_j = \phi$, where $\forall_{m\neq n}\varphi_m \neq \varphi_n$ and $\varphi \in FL(\rho)$. So, if $\langle s,\kappa\rangle\Psi \succ \langle p,\tau\rangle\phi$ for κ a normalised Stochastic Petri Net program, then τ is an atomic Stochastic Petri Net program or is equal to κ. Therefore there can be no infinitely-long $\succ$-sequences.

So, $FL(\varphi)$ is finite. ■

Lemma 74

(i) If $\sigma \in FL(\varphi)$, then $FL(\sigma) \subseteq FL(\varphi)$

(ii) If $\sigma \in FL(\langle s,\pi\rangle\varphi)$, then $FL(\sigma) \subseteq FL(\langle s,\pi\rangle\varphi) \cup FL(\varphi)$

Proof: This proof is the same than for Petri-PDL, regarding $\mathcal{DS}_3$ FL Definition 73. ■

Definition 75 *Filtration*

Given a $\mathcal{DS}_3$ formula φ, a $\mathcal{DS}_3$ model $\mathcal{K} = \langle W, R_\eta, M, (\Pi,\Lambda), \delta, \mathbf{V}\rangle$, we define a new model

$$\mathcal{K}_3 = \langle W^\varphi, R_\eta^\varphi, M^\varphi, (\Pi^\varphi, \Lambda^\varphi), \delta^\varphi, \mathbf{V}^\varphi\rangle,$$

the filtration of $\mathcal{K}_3$ by $FL(\varphi)$ for a normalised Stochastic Petri Net program Π, as follows.

The relation $\equiv$ *over the worlds of* $\mathcal{K}_3$ *is defined as*

$$u \equiv v \leftrightarrow \forall \phi \in FL(\varphi), \Pr(\mathcal{K}_3,u \Vdash \phi \mid \delta(u,\Pi)) = \Pr(\mathcal{K}_3,v \Vdash \phi \mid \delta(v,\Pi))$$

and the relation R_η^φ *is defined as*

$$[u]R_\eta^\varphi[v] \leftrightarrow (\exists u' \in [u] \wedge \exists v' \in [v] \wedge u' R_\eta v').$$

where

(a) $[u] = \{v \mid v \equiv u\}$

(b) $W^\varphi = \{[u] \mid u \in W\}$

(c) $[u] \in \mathbf{V}^\varphi(p)$ *iff* $u \in V(p)$

(d) $M^\varphi([u]) = \langle s_1, s_2, \dots \rangle$ *where for all* $j \geq 1, v_j \in [u]$ *iff* $M(v_j) = s_j$

(e) $(\Pi^\varphi, \Lambda^\varphi) = (\Pi, \Lambda)$

(f) $\delta^\varphi([u],\pi) = \langle d_1, d_2, \dots, d_n \rangle$ *where* $\pi = \pi_1 \odot \pi_2 \odot \cdots \odot \pi_n$ *and* $d_i = \int_0^i h(\delta(u,\Pi))du$, *according to equation* (4-5) *and the stability process (Haas, 2002), where* h *is the function that decreases the firing delays.*

The process to compute δ^φ *is defined only for normalised Stochastic Petri Net programs; it is derived from the equation* (4-5) *and the stability process for Stochastic Petri Nets by the functional version of the Central Limit Theorem (Haas, 2002; James, 2006).*

All rules may be composed inductively to extend in order to compound all programs and propositions due to compositions as in definition 59.

Lemma 76 *Filtration Lemma*

$\forall u,\ v \in W,\ uR_\eta v$ *iff* $[u]R_\eta^\varphi[v]$

Proof: From Definition 75 $w \in [w]$ *iff* $\forall w' \in [w],\ w' \equiv w$ *(1)*
and $[u]R_\eta^\varphi[v]$ *for some* $u \in [u]$ *and* $v \in [v]$ *we have that* $u'R_\eta^\varphi v'$ *(2).*
So if $uR_\eta^\varphi v$ *and we do not have that* $[u]R_\eta^\varphi[v]$ *then it will contradicts (1). If* $[u]R_\eta^\varphi[v]$ *but we do not have that* $uR_\eta^\varphi v$ *then it will contradicts (2).* ■

Lemma 77 $\mathcal{K}_3^\varphi$ *is a finite* $\mathcal{DS}_3$ *model.*
Proof:

- W^φ *is a finite set of states by Definition 75 and Lemma 73.*
- $M^\varphi \colon W^\varphi \to S$ *by Definition 75.*

- $R_\eta^\varphi = \{([w],[v]) \mid$ for some η_i $\exists[u]$ such that $s_i \prec M^\varphi([u])$ and $[w]R_{\eta_i}[u]$ and $[u]R_\eta[v]\}$ *for any program* $\eta = \eta_1 \odot \cdots \odot \eta_n$, *where* $s_i = f(s, \eta_i)$ *and* $1 \leq i \leq n$.
- $\mathbf{V}^\varphi \colon \Phi \to 2^{W^\varphi}$ *by Definition 75.*
- $(\Pi^\varphi, \Lambda^\varphi)$ *is a Stochastic Petri Net program by Definition 75.*
- $\delta^\varphi \colon W^\varphi \times \Pi \to 2^{\mathbb{R}}$ *by Definition 75.*

Then $\mathcal{K}_3^\varphi$ *is a finite* $\mathcal{DS}_3$ *model.* ■

Corollary 78 *Decidability*
Proof: By Lemma 77 the number of states is finite, then there is an algorithm to check whether a formula φ *of* $\mathcal{DS}_3$ *is satisfiable.* ■

4.5 Computational complexity

Petri-PDL language is the same then the $\mathcal{DS}_3$ and its expressive power is also subsumed by $\mathcal{DS}_3$. So $\mathcal{DS}_3$ satisfiability computational complexity is stated bellow.

Lemma 79 *The satisfiability of* $\mathcal{DS}_3$ *is EXPTime-hard.*
Proof: Taking Lemma 45, we use the Petri Net which models the game stating the same firing rate to each transition. As the Petri-PDL language is the same then $\mathcal{DS}_3$ *then the reduction procedure presented in Lemma 45 and Theorem* **??** *is also valid for* $\mathcal{DS}_3$. *So* $\mathcal{DS}_3$ *SAT is EXPTime-hard.* ■

4.6 A Natural Deduction system for $\mathcal{DS}_3$

The syntax of the Natural Deduction system for Petri-PDL (Section 3.5) is same than for $\mathcal{DS}_3$. The difference is that now, regarding a model $\mathcal{M}_3 = \langle W, R_\pi, M, (\Pi, \Lambda), \delta, \mathbf{V}\rangle$, rule (3-10) has also the restriction that $\Pr(\mathcal{M}_3, w \Vdash \langle s, \pi\rangle\varphi \mid \delta(w, \Pi)) > 0$, where $w \in W$ and rule (3-12) has also the restriction that $\Pr(\mathcal{M}_3, w \Vdash \langle s, \pi\rangle\varphi \mid \delta(w, \Pi)) = 1$. The proofs of soundness and completeness of the system are also valid for $\mathcal{DS}_3$ regarding its semantical notion.

4.7 Usage examples

This section presents some usage examples for $\mathcal{DS}_3$.

4.7.1 A multi-agent system

The Petri Net in Figure 4.2 presents a scenario where four agents (A_1, A_2, A_3 and A_4) must collect and process some data from the resource centre (r), but agents A_1 and A_2 cannot make the full process and needs that A_3 or A_4 completes the computation. Another characteristic of this system is that A_3 and A_4 have a faster processor than A_1 and A_2 and that A_1 and A_2 are in a shared memory system, but the clock of the processor of A_1 is faster than A_2. As the clock of the processor of A_1 is faster than the one of A_2, the firing rates (i.e. the λ parameter of the random variable which is associated with the transitions whose preset or postset depends on A_1) is greater than the ones of A_2.

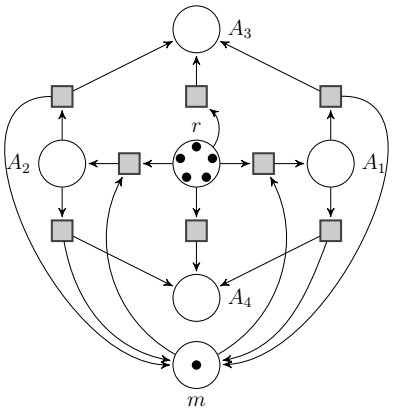


Figure 4.2: Petri Net of a four agents system

Taking a propositional formula p that means that all data was processed, the formula $\langle\{rrrrrm\}, rmt_2A_1 \odot rmt_2A_2 \odot rt_1A_3 \odot rt_1A_4 \odot A_1t_3A_3m \odot A_2t_3A_3m \odot A_1t_3A_4m \odot A_4t_3A_4m\rangle p$ says that after some running of the Petri Net of Figure 4.2, p holds, that is, all the data are processed. Verifying if this formula holds in a state w of a model $\mathcal{M}$ (i.e. verifing if it is possible that some transition fires) is equivalent to compute that the probability of some basic program fire is greater than zero, which is reduced to the equation in lemma 63. In order to verify if it is possible that A_1 and A_2 compute some data in parallel, we verify that after some of them begin to process something (i.e. rmt_2A_1 or rmt_2A_2 fires), m will not be anymore in the sequence of names,

so it is not possible that the other agent starts to compute something unless a transition that restates a token to m fires.

If it is desirable to know if, from a state w, it is possible that some agent (e.g. agent A_1) collect some data to process, then we need to compute $\Pr(\mathcal{M},w \Vdash \langle s, rmt_2A_1\rangle\top \mid \delta(w, rmt_2A_1 \odot rmt_2A_2 \odot rt_1A_3 \odot rt_1A_4 \odot A_1t_3A_3m \odot A_2t_3A_3m \odot A_1t_3A_4m \odot A_4t_3A_4m))$, where $s = M(w)$, and verify if it is greater than zero. By lemma 63 it is equivalent to verify if

$$\frac{\delta(w, rmt_2A_1)}{\sum\limits_{\pi_b \in \Pi: f(s,\pi_b)\neq\epsilon} \delta(w, \pi_b)}$$

is greater than zero, where $\Pi = rmt_2A_1 \odot rmt_2A_2 \odot rt_1A_3 \odot rt_1A_4 \odot A_1t_3A_3m \odot A_2t_3A_3m \odot A_1t_3A_4m \odot A_4t_3A_4m$.

A more sophisticated example concerns in verifying if the transmission ratings from agents A_1 and A_2 to the agents A_3 and A_4 are overheading agents A_3 and A_4. That is verify if the programs $A_1t_1A_3$, $A_1t_1A_4$, $A_2t_1A_3$ and $A_2t_1A_4$ are firing more times than rt_1A_3 and rt_1A_4. This is equivalent to verify if the probabilities of firing that first basic programs are greater than these last ones. So it is equivalent to verify if for a sequence $\Lambda(A_1t_1A_3 \odot A_1t_1A_4 \odot A_2t_1A_3 \odot A_2t_1A_4)$ from an initial state v_1 such that $v_1R_\Pi v_n = v_1Rv_2 \circ \cdots \circ v_{n-1}Rv_n$, where Π stops in state v_n,

$$\sum_{\delta(v_i, A_1t_1A_3 \odot A_1t_1A_4 \odot A_2t_1A_3 \odot A_2t_1A_4)} 1 > \sum_{\delta(v_i, rt_1A_3 \odot rt_1A_4)} 1$$

for $1 \leq i \leq n$ where all the involved basic problems are enabled. Determine a good firing rate for $A_1t_1A_3$, $A_1t_1A_4$, $A_2t_1A_3$ and $A_2t_1A_4$ is an optimisation problem for $\Lambda(A_1t_1A_3 \odot A_1t_1A_4 \odot A_2t_1A_3 \odot A_2t_1A_4)$.

4.7.2 A Kanban system

As another usage example take a Kanban system (Marsan et al., 1995), a *Just-In-Time* based flow control method. The SPN designed in Figure 4.3 represents a "cards" (the K tokens of place BB) flow of resources control with failure for a Kanban cell (a processing unit that may communicate with others). The place IB denotes the Input Buffer where the resources are stored (already with a card) before processed. If everything is OK (i.e. the place OK has a token) and the processing system is not busy (i.e. there is a token in place Id) then the resource is processed (the token goes to place B) and thereafter the resource goes to the Output Buffer (the place OB).

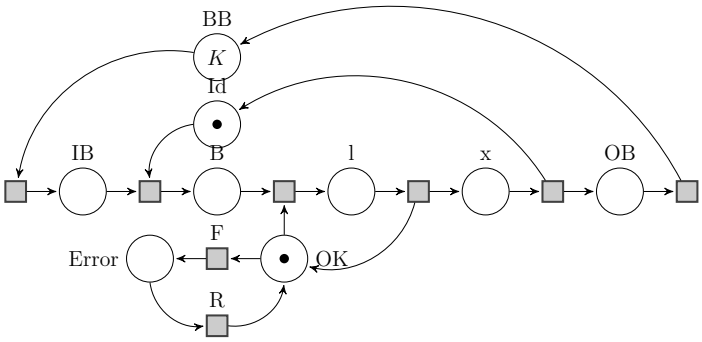


Figure 4.3: A Kanban cell with failure

When the transition F fires it denotes that some failure occurred. Similarly, when the transition R fires it denotes that the system was repaired. The failure rate and the time needed to process the resource are controlled by the parameters of the random variables associated with the respective transitions.

Modelling this scenario in a $\mathcal{DS}_3$ model $\mathcal{M} = \langle W, R_\pi, M, (\Pi, \Lambda), \delta \rangle$, we have the formula $\langle (s), Kt_1IB \odot IB, Idt_2B \odot B, OKt_2l \odot lt_3OK, x \odot xt_3Id, OB \odot OBt_1BB \odot OKt_1\text{Error} \odot \text{Error}t_1Ok \rangle \varphi$ where s is a sequence of names composed by K repetitions of "BB" and "OK" and φ is some property that holds after the running of this SPN. Verify if this formula holds in a state w of a model $\mathcal{M}$ (i.e. verify if it is possible that some transition fires) is equivalent to compute the probability of some basic program fires is greater than zero, which is reduced to the equation in lemma 63. To verify if it is possible process two resources at the same time, we verify that after some of them begin to process something (i.e. there is a token in place B), Id will not be anymore in the sequence of names, so it is not possible that other resource begins it process unless a transition that restates a token to Id fires.

Verify if from a world $w \in W$ it is possible that some resource begins its processing is equivalent to compute if $\Pr(\mathcal{M}, w \Vdash \langle r, IB, Idt_2B \odot B \rangle \top \mid \delta(w, Kt_1IB \odot IB, Idt_2B \odot B, OKt_2l \odot lt_3OK, x \odot xt_3Id, OB \odot OBt_1BB \odot OKt_1\text{Error} \odot \text{Error}t_1Ok)) > 0$ where $r = M(w)$. Using lemma 63 it is equivalent to verify if

$$\frac{\delta(Idt_2B \odot B)}{\sum_{\pi_b \in \Pi : f(r, \pi_b) \neq \epsilon} \delta(w, \pi_b)} > 0$$

where $\Pi = Kt_1IB \odot IB, Idt_2B \odot B, OKt_2l \odot lt_3OK, x \odot xt_3Id, OB \odot OBt_1BB \odot OKt_1\text{Error} \odot \text{Error}t_1Ok$ and π_b is a basic transition of Π.

4.8
A note on the length of proofs

To describe the behaviour of a program it is necessary a long sequence of states (worlds). It is needed to apply the axiom (PC) consecutively and eliminate the disjunctions which may lead to a proof difficult to read. In order to reduce the length of proofs, we will introduce another approach for $\mathcal{DS}_3$, using a transitive closure operator.