12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

A Framework for Supervisory Control of Probabilistic Discrete Event Systems

Vera Pantelic, Mark Lawford and Steven Postma

McMaster Centre for Software Certification, Department of Computing and Software, Faculty of Engineering, McMaster University, 1280 Main Street West, Hamilton, ON, Canada L8S 4K1 (e-mail: pantelv at mcmaster dot ca, lawford at mcmaster dot ca, steven.m.postma at gmail dot com)

Abstract: This paper focuses on a framework for probabilistic supervisory control of probabilistic discrete event systems (PDES). PDES are modelled as generators of probabilistic languages, and the supervisors used are probabilistic. In our previous work, we presented and solved a number of supervisory control problems inside the framework. We also suggested a pseudometric to measure the behavioural similarity between PDES, and used the pseudometric in the solution of two optimal supervisory control problems defined in the framework. In this paper, we survey these results and introduce a real-world application of the framework. Further, we investigate a relationship between our framework and that of Markov Decision Processes, that could prove beneficial for both control synthesis and probabilistic model checking.

Keywords: Supervisory control, stochastic systems, discrete event systems, metrics, MDPs.

1. INTRODUCTION

A framework for supervisory control of *probabilistic discrete event systems* (PDES) was developed in our previous work (Lawford and Wonham, 1993; Postma and Lawford, 2004; Pantelic et al., 2009; Pantelic and Lawford, 2010, 2009; Pantelic, 2011; Pantelic and Lawford, 2012a,b). This paper integrates the precursory work with novel results pointing to important avenues for future research.

In the framework, PDES are modelled as *probabilistic generators*: an extension of regular generators used in supervisory control theory. More precisely, every transition in the regular generator is extended with its probability of occurrence: the probabilities of all the transitions from a state should not be greater than 1. The control used in our framework is probabilistic: instead of only being able to enable/disable a controllable event, the probabilistic supervisor enables a controllable event with a certain probability. First, Lawford and Wonham (1993), Postma and Lawford (2004) and Pantelic et al. (2009) focused on the solution of the Probabilistic Supervisory Control Problem (PSCP). The PSCP tries to find a probabilistic supervisor such that the plant's behaviour under probabilistic control satisfies a given probabilistic specification. The solution gives necessary and sufficient conditions for the existence of a probabilistic supervisor, and, if the conditions are satisfied, synthesizes the supervisor. Then, in Pantelic and Lawford (2009) and Pantelic and Lawford (2012a), an optimal supervisory control problem inside the probabilistic framework is posed and solved. As in classical supervisory theory, if there does not exist a (probabilistic) supervisor such that the controlled plant's behaviour can exactly match a prespecified probabilistic behaviour, a supervisor is synthesized such that the controlled plant's behaviour is "as close as possible" to the desired behaviour. The measure of similarity is a *pseudometric* on states of probabilistic generators. The concept of the pseudometric is useful outside the control framework: the pseudometric measures behavioural similarity between probabilistic generators, and can be used for e.g., model reduction as explored in Pantelic and Lawford (2012b).

The contributions of this paper are as following. First, we survey the previous results in the framework. Then, an example of an application of the framework is presented that has not been published anywhere except in the Ph.D. thesis of the first author (Pantelic, 2011). Then, we focus on *Markov Decision Processes* (MDPs), a widely used framework for control of probabilistic discrete event systems, with the goal of exploring the relationship between our framework and MDPs. Initial results on the relationship were presented in the Ph.D. thesis of the first author (Pantelic, 2011). We show that a probabilistic generator can be viewed as a (probabilistic) policy for MDPs (see Pantelic (2011)). On the other hand, a probabilistic supervisor as defined in our framework can be represented as an MDP. This duality between the plant to be controlled and the controller might provide interesting connections between probabilistic model checking and supervisory control theory. Therefore, the duality might pave the road for the exchange of results between the two frameworks.

The outline of the paper is as following. Section 2 presents the main results in the framework. More precisely, some previous results are summarized (solutions of probabilistic matching and optimal control problems), and a new, real-world application is presented. Then, Section 3 discusses the interplay between the framework and MDPs. Section 4 concludes the paper and sketches future work.

2. THE FRAMEWORK

This section introduces probabilistic generators as the model for PDES. Then, probabilistic control is defined, and some previous results are presented: control problems in the framework and their solutions, and a pseudometric that measures the behavioural similarity of probabilistic generators. Also, an application of the results is presented.

In the sequel, for given sets A and B, the power set of A will be denoted by $\mathcal{P}(A)$, and the set difference of A and B by $A\backslash B$. Further, B^A will be used to denote a set of functions from A to B.

2.1 Probabilistic Generators: The Model

In our framework, a PDES is modeled following Lawford and Wonham (1993) as a *probabilistic generator* $G = (Q, \Sigma, \delta, q_0, p)$, where Q is the nonempty finite set of states, Σ is a finite alphabet whose elements we will refer to as event labels, $\delta : Q \times \Sigma \to Q$ is the (partial) transition function, $q_0 \in Q$ is the initial state, and $p : Q \times \Sigma \to [0, 1]$ is the statewise event probability distribution, i.e. for any $q \in Q$, $\sum_{\sigma\in\Sigma} p(q, \sigma) \leq 1$. The probability that the event $\sigma \in \Sigma$ is going to occur at the state $q \in Q$ is $p(q, \sigma)$. For the generator G to be well-defined, $p(q, \sigma) = 0$ should hold if and only if $\delta(q, \sigma)$ is undefined. The probabilistic generator G is nonterminating if, for every reachable state $q \in Q$, $\sum_{\sigma\in\Sigma} p(q, \sigma) = 1$. Conversely, G is terminating if there is at least one reachable state $q \in Q$ such that $\sum_{\sigma\in\Sigma} p(q, \sigma) < 1$. The probability that the system terminates at state q is $1 - \sum_{\sigma\in\Sigma} p(q, \sigma)$. Throughout the sequel, unless stated otherwise, we assume nonterminating generators. If a PDES is terminating, it can easily be transformed into a nonterminating one using the technique described in Lawford and Wonham (1993).

The transition function is traditionally extended by induction on the length of strings to $\delta : Q \times \Sigma^* \to Q$ in a natural way. For a state q, and a string s, the expression $\delta(q, s)!$ will denote that δ is defined for the string s in the state q. The language $L(G)$ generated by a probabilistic DES generator $G = (Q, \Sigma, \delta, q_0, p)$ is $L(G) = \{s \in \Sigma^* \,|\, \delta(q_0, s)!\}$. The probabilistic language generated by G is defined as:

$$L_p(G)(\epsilon) = 1,$$

$$L_p(G)(s\sigma) = \begin{cases} L_p(G)(s) \cdot p(\delta(q_0, s), \sigma), & \text{if } \delta(q_0, s)! \\ 0, & \text{otherwise.} \end{cases}$$

Informally, $L_p(G)(s)$ is the probability that the string s is executed in G. Also, $L_p(G)(s) > 0$ iff $s \in L(G)$.

For each state $q \in Q$, we define the function $\rho_q : \Sigma \times Q \to [0, 1]$ such that for any $q' \in Q$, $\sigma \in \Sigma$, we have $\rho_q(\sigma, q') = p(q, \sigma)$ if $q' = \delta(q, \sigma)$, and 0 otherwise. The function ρ_q is a probability distribution on the set $\Sigma \times Q$ induced by q. Also, for a state q, we define *the set of possible events* to be $Pos(q) := \{\sigma \in \Sigma | p(q, \sigma) > 0\}$.

Next, the synchronous product of (nonprobabilistic) discrete event systems (DES) that underlie PDES is defined in a standard manner. For a probabilistic generator $G = (Q, \Sigma, \delta, q_0, p)$, the (nonprobabilistic) discrete event system (DES) that underlies G will be denoted G^{np}, i.e., $G^{np} = (Q, \Sigma, \delta, q_0)$ throughout this paper. Let G_1^{np} and G_2^{np} be the nonprobabilistic generators (DES) underlying $G_1 = (Q_1, \Sigma, \delta_1, q_{0_1}, p_1)$ and $G_2 = (Q_2, \Sigma, \delta_2, q_{0_2}, p_2)$, respectively, i.e., $G_1^{np} = (Q_1, \Sigma, \delta_1, q_{0_1})$ and $G_2^{np} = (Q_2, \Sigma, \delta_2, q_{0_2})$.

Definition 1. The synchronous product of $G_1^{np} = (Q_1, \Sigma, \delta_1, q_{0_1})$ and $G_2^{np} = (Q_2, \Sigma, \delta_2, q_{0_2})$, denoted $G_1^{np} \parallel G_2^{np}$, is the reachable sub-DES of DES $G_a = (Q_a, \Sigma, \delta, q_0)$, where $Q_a = Q_1 \times Q_2$, $q_0 = (q_{0_1}, q_{0_2})$, and, for any $\sigma \in \Sigma$, $q_i \in Q_i$, $i = 1, 2$, it holds that $\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$ whenever $\delta_1(q_1, \sigma)!$ and $\delta_2(q_2, \sigma)!$.

2.2 Probabilistic Supervisory Control of PDES

The set Σ is partitioned into the uncontrollable event set Σ_u and the controllable event set Σ_c. Deterministic supervisors for DES are generalized to probabilistic supervisors. Instead of deterministically enabling or disabling controllable events, probabilistic supervisors enable them with certain probabilities. This means that, upon reaching a certain state q, the control pattern is chosen according to supervisor's probability distributions of controllable events. Consequently, the controller does not always enable the same events when in the state q.

Definition 2. Let $x : L(G) \to [0, 1]^{\Sigma_c}$. For a PDES $G = (Q, \Sigma, \delta, q_0, p)$, a *probabilistic supervisor* is a function $V_p : L(G) \to [0, 1]^{\Sigma}$ such that

$$(\forall s \in L(G))(\forall \sigma \in \Sigma) V_p(s)(\sigma) = \begin{cases} 1, & \text{if } \sigma \in \Sigma_u \\ x(s)(\sigma), & \text{otherwise.} \end{cases}$$

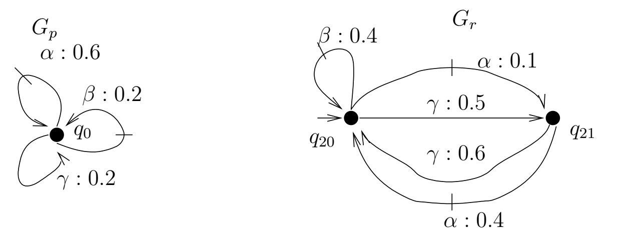

Fig. 1. Plant G_p and requirements specification G_r

Therefore, after observing a string $s \in L(G)$ (all the events are assumed to be observable), the supervisor enables event σ with probability $V_p(s)(\sigma)$. More precisely, for event σ, the supervisor performs a Bernoulli trial with possible outcomes *enable* (that has the probability $V_p(s)(\sigma)$), and *disable* (with probability $1 - V_p(s)(\sigma)$), and, depending on the outcome of the trial, decides whether to enable or disable the event. After (independent) Bernoulli trials have been performed for all controllable events, control pattern Θ is determined as a set of controllable events such that a controllable event belongs to Θ if and only if its corresponding Bernoulli trial resulted in outcome *enable*. Thus the *controllable event set probability* of Θ, i.e., the probability that V_p enables the controllable event in Θ after observing string s is given by:

$$P(V_p \text{ enables } \Theta | s) = \prod_{\sigma\in\Theta} V_p(s)(\sigma) \quad \cdot \prod_{\sigma\in(Pos(q)\cap\Sigma_c)\backslash\Theta} (1 - V_p(s)(\sigma)) \tag{1}$$

After Θ has been decided upon, the system acts as if supervised by a deterministic supervisor. Let $q \in Q$ be the state of the plant after $s \in L(G)$ has been observed. The plant G under the control of the supervisor V_p will be denoted V_p/G. The probability that the event $\alpha \in \Sigma$ will