Scale-Freeness of SPA Models with Weighted Immediate Actions

Johann Schuster and Markus Siegle

University of the Federal Armed Forces Munich, Germany
{johann.schuster,markus.siegle}@unibw.de

Abstract. Whenever a process algebra uses weights for specifying probabilities, it is desirable that the rescaling of a submodel's weights by a constant factor does not affect the resulting overall model. A classical weighted approach, which is independent of rescaling weights in submodels, is the WSCCS approach by Tofts. The stochastic process algebra CASPA also uses weights, but the results are in general not independent of rescaling the submodels' weights. This paper develops necessary and sufficient criteria for CASPA models to be independent of rescalings. In addition to the general notion of scale-freeness, weaker notions that do not regard vanishing states or target on certain measures are also considered.

Keywords: stochastic process algebra, scale-free, CASPA, nondeterminism

1 Introduction

Stochastic Process Algebra (SPA) is a popular formalism used for performance and dependability modelling and evaluation. In most classical SPAs, all actions are associated with an exponentially distributed delay. However, similar to the area of GSPNs [1], immediate actions which do not consume any time can be a very useful feature, for instance for synchronising processes or for modelling probabilistic decisions.

This paper focuses on SPAs that offer both Markovian and immediate actions. Markovian actions are specified by their name and their rate, while immediate actions are specified by their name and weight. Weights are eventually transformed into probabilities in the following way: When the overall model has been constructed (in a compositional way) from subprocesses, no further synchronisation is necessary. At that point we say that the model is *closed*. On the closed model, weights are then normalised to probabilities.

In order to motivate the problem statement we use an example from [2], written in CASPA SPA (cf. [14, 17]). Suppose there are two processes:

```
A:=(*s,1*);A1+(*t,1*);A2
B:=(*s,2*);B1+(*s,3*);B2
```

(for details of the syntax and semantics cf. Sec. 2). Synchronising `A` and `B` over action `s` results in a closed model whose behaviour is shown in Fig. 1. The normalised transition probabilities in this case are (from left to right) $\frac{1}{3}$, $\frac{1}{2}$, $\frac{1}{6}$. Rescaling the weights of submodel `B` by factor 2 to `B:=(*s,4*);B1+(*s,6*);B2` would lead to transition probabilities $\frac{4}{11}$, $\frac{6}{11}$, $\frac{1}{11}$, which is clearly a different distribution. Such an effect is undesirable, since the local rescaling of a submodel's weights should not produce any effect on the overall model! In particular, this would be problematic when different modellers work on different parts of the same overall model.

$$\begin{tikzcd}
 & \mathtt{A|[s]|B} \arrow[dl, dashed, "(s,1\cdot 2)"'] \arrow[d, dashed, "(s,1\cdot 3)"] \arrow[dr, dashed, "(t,1)"] & \\
\mathtt{A1|[s]|B1} & \mathtt{A1|[s]|B2} & \mathtt{A2|[s]|B}
\end{tikzcd}$$

Fig. 1: An example that does not allow rescaling of submodel B's weights

The problem of normalising weights to probabilities in SPA is not new, as the following glance at related work shows: A classical approach, which is independent of rescaling weights in submodels is the WSCCS approach by Tofts [21], where time is discrete and processes proceed in lock-step. However, the present paper is concerned with the continuous-time scenario, where processes evolve in parallel, with the possibility of synchronising on certain events. [7] described such an approach in the context of semi-Markov PEPA, where weights are employed to determine the likelihood of occurrence of certain general distributions. The paper [11] introduces iPEPA, which is PEPA extended by weighted immediate actions. Weights are transformed into probabilities before any parallel composition takes place, and those probabilities are then recomputed at each level of composition. Both approaches just mentioned, however, are not scale-free. EMPA [5] and some versions of TIPP [16] were actually the first Markovian process algebras to offer (weighted) immediate actions, but there the scaling problem did not occur since only local normalisation was required. As investigated in [8], the problem of normalisation is also present in PTPA and some probabilistic variants of LOTOS. In these SPAs there may be competing synchronous and asynchronous transitions, and the possibly nondeterministic choices between them are converted by the semantics to a probabilistic execution fragment in the sense of [19].

Everything is fine as long as the modeller is aware of situations where potential nondeterminism could arise. However, if the modeller misses a potentially nondeterministic situation, the analysis results could be misleading, as they depend on the scaling of certain weights, i.e. their ratios, as the example in Fig. 1 has shown. Therefore it is very important to develop precise criteria for automatically deciding whether a given model is scale-free or not, and this is exactly the topic of this paper. The paper will give a necessary and sufficient criterion for CASPA models to be scale-free, and in addition a sufficient criterion which is considerably easier to prove. The results in this paper are for the stochastic process algebra CASPA [14, 17], but the same ideas apply to other SPAs that

1. have both Markovian and immediate actions,
2. use weights to specify probabilities of immediate actions, and
3. use the product of weights for the weight of a synchronised immediate action.

The remainder of the paper is organised as follows: Sec. 2 sketches the CASPA syntax by means of a small example and provides the semantic rules. Sec. 3 defines different notions of scale-freeness and establishes the relationships between them. Sec. 4 proves the main lemmata to characterise scale-freeness in the strong and weak sense, and Sec. 5 concludes the paper.

2 CASPA language

The CASPA language is a stochastic process algebra, notated in the form of a guarded command language, that is used to specify Weighted Stochastic Labelled Transition Systems:

Definition 1. *A* Weighted Stochastic Labelled Transition System *(WSLTS) is a tuple* $(S, L_M, L_I, \rightarrow, \dashrightarrow, s_0)$ *where* S *is a finite set of states,* $s_0 \in S$ *is the* initial state. *Two sets of action labels are present:* L_M *(Markovian labels) and* L_I *(immediate labels),* $L_M \cap L_I = \emptyset$. *Let* $Act = L_M \cup L_I$. *A special label* $\tau \in L_I$ *will be used for the internal tau transition. The transition relations are given as* $\rightarrow \subseteq S \times L_M \times \mathbb{R}^{>0} \times S$ *and* $\dashrightarrow \subseteq S \times L_I \times \mathbb{R}^{>0} \times S$. *An element* $(s_1, a, \lambda, s_2) \in \rightarrow$ *is called a* Markovian *transition from* s_1 *to* s_2 *with the action label* a *and the rate* λ *(describing a negative exponentially distributed random variable). Alternatively we write* $s_1 \xrightarrow{a,\lambda} s_2$. *An element* $(s_1, b, w, s_2) \in \dashrightarrow$ *is called an* immediate *transition from* s_1 *to* s_2 *with the action label* b *and the* weight w. *Alternatively we write* $s_1 \overset{b,w}{\dashrightarrow} s_2$. *A state with at least one outgoing immediate transition is called* vanishing, *otherwise it is called* tangible.

Remark 1. In some modelling frameworks, immediate transitions are given without any weight or probability. This leads to nondeterministic behaviour, as e.g. in the case of Interactive Markov Chains (IMCs) [12]. In our setup we use weights to generate transition probabilities. Both for immediate and for exponentially distributed transitions it is assumed that no "parallel" transitions with the same action label exist. They are grouped together by summing up the parallel weights/rates, i.e. $s_1 \overset{b,w_1}{\dashrightarrow} s_2$ and $s_1 \overset{b,w_2}{\dashrightarrow} s_2$ will be grouped to $s_1 \overset{b,w_1+w_2}{\dashrightarrow} s_2$.

The basic building blocks of the CASPA language are Markovian actions (a, λ) (here action a with rate λ) and immediate actions $(*b, w*)$ (here action b with weight w). Sequential processes are defined by means of the prefix operator ; and the choice operator +. The synchronised parallel composition operator is written as $|[S]|$ for a synchronisation set $S \subseteq L_I$[1]. Finally, the hiding operator `hide` turns a set of immediate labels into the internal label τ. The complete syntax can be found in [18].

The CASPA semantics in condensed form is given in Tab. 1 (notated in the style of [15]). In the table, P, P', Q and Q' are valid expressions in the language, and λ, μ, ω, σ are positive real numbers. By applying these semantic rules, for every CASPA model C a multi-transition system, and from this by standard flattening [13] a unique WSLTS W is generated. We denote by $\mathcal{C}_{cl}$ the set of *closed* CASPA models, i.e. models which will not be used in further parallel composition. In a closed model

– all immediate action labels can safely be ignored
– time can only evolve in tangible states, (maximal progress assumption [12])
– weights can be normalised to probabilities
– a tangible initial state is required

A CASPA state measure characterises a subset of states. It can be used in connection with transient or steady-state analysis. For a more detailed explanation cf. [2].

$$\frac{}{(a,\lambda);P \xrightarrow{(a,\lambda)} P}\; a \in L_M \qquad \frac{}{(*b,\omega*);P \overset{(b,\omega)}{\dashrightarrow} P}\; b \in L_I$$

$$\frac{P \xrightarrow{(a,\lambda)} P'}{P+Q \xrightarrow{(a,\lambda)} P'}\; a \in L_M \qquad \frac{Q \xrightarrow{(a,\lambda)} Q'}{P+Q \xrightarrow{(a,\lambda)} Q'}\; a \in L_M$$

$$\frac{P \overset{(b,\omega)}{\dashrightarrow} P'}{P+Q \overset{(b,\omega)}{\dashrightarrow} P'}\; b \in L_I \qquad \frac{Q \overset{(b,\omega)}{\dashrightarrow} Q'}{P+Q \overset{(b,\omega)}{\dashrightarrow} Q'}\; b \in L_I$$

$$\frac{P \xrightarrow{(a,\lambda)} P'}{P|[S]|Q \xrightarrow{(a,\lambda)} P'|[S]|Q}\; a \in L_M \qquad \frac{Q \xrightarrow{(a,\lambda)} Q'}{P|[S]|Q \xrightarrow{(a,\lambda)} P|[S]|Q'}\; a \in L_M$$

$$\frac{P \overset{(b,\omega)}{\dashrightarrow} P'}{P|[S]|Q \overset{(b,\omega)}{\dashrightarrow} P'|[S]|Q}\; b \notin S \qquad \frac{Q \overset{(b,\omega)}{\dashrightarrow} Q'}{P|[S]|Q \overset{(b,\omega)}{\dashrightarrow} P|[S]|Q'}\; b \notin S$$

$$\frac{P \overset{(s_b,\omega)}{\dashrightarrow} P' \qquad Q \overset{(s_b,\sigma)}{\dashrightarrow} Q'}{P|[S]|Q \overset{(s_b,\omega\cdot\sigma)}{\dashrightarrow} P'|[S]|Q'}\; s_b \in S \subseteq L_I$$

$$\frac{P \overset{(s_b,\omega)}{\dashrightarrow} P'}{\texttt{hide } S \texttt{ in } P \overset{(\tau,\omega)}{\dashrightarrow} \texttt{hide } S \texttt{ in } P'}\; s_b \in S \qquad \frac{P \overset{(s_b,\omega)}{\dashrightarrow} P'}{\texttt{hide } S \texttt{ in } P \overset{(s_b,\omega)}{\dashrightarrow} \texttt{hide } S \texttt{ in } P'}\; s_b \notin S$$

$$\frac{P \xrightarrow{(a,\lambda)} P'}{\texttt{hide } S \texttt{ in } P \xrightarrow{(a,\lambda)} \texttt{hide } S \texttt{ in } P'}\; a \in L_M$$

Table 1: CASPA language semantics

[1] Other versions of CASPA also allow the synchronisation over Markovian labels, but since this is irrelevant for the considerations of this paper we decided to omit this feature.

2.1 A small running example

Consider a production system with two machines – represented by their input buffers – that are loaded by a shared robot (an extremely simplified version of the model in [20]). The CASPA model of the system is given in Fig. 2. Lines 1-4 describe one input buffer (X is to be substituted by 1 or 2), the robot that loads the buffers is described in lines 5-7 and the system definition is given in lines 8-10 (synchronisations over `start_enqueue`, `stop_enqueue`). The state measures of interest are defined in lines 11-13.

The possibly nondeterministic choice which buffer should be loaded next is resolved by CASPA semantics to a probabilistic execution fragment (cf. [19]) according to the ratios of weights. Suppose that all weights of immediate actions in Buffer2 are changed by the same constant factor α_2. From a subprocess-perspective, this should lead to the *same* resulting probabilities. But this is not the case in this example, as we shall see later.

```
(1)  BufferX(state [2]) :=
(2)       [state = IDLE] -> (*start_enqueue,1*); BufferX(WORKING)
(3)       [state = WORKING] -> (enqueue,5); BufferX(FINISHED)
(4)       [state = FINISHED] -> (*stop_enqueue,1*); BufferX(IDLE)

(5)  Robot(state [1]) :=
(6)       [state = IDLE] -> (*start_enqueue,1*); Robot(BUSY)
(7)       [state = BUSY] -> (*stop_enqueue,1*); Robot(IDLE)

(8)  Sys := ( Buffer1(WORKING)|[]|Buffer2(IDLE) )
(9)           |[start_enqueue, stop_enqueue]|
(10)        Robot(BUSY)

(11) statemeasure buf1working Buffer1(state=WORKING)
(12) statemeasure buf2working Buffer2(state=WORKING)
(13) statemeasure any_working Buffer1(state=WORKING) | Buffer2(state=WORKING)
```

Fig. 2: CASPA model and measure definitions

3 Scale-freeness

The example from the previous section shows that it is natural to ask for CASPA models that lead to the same results for certain measures, even if weights of immediate transitions are rescaled for any submodel. To make this statement more precise, we will generalise CASPA models to accept also variables (in addition to constants) for their weights. Before doing this, we introduce a powerful tool from algebra, the quotient field.

Definition 2. *For a CASPA model C with N sequential submodels we define*

1. *a* generalised weight *as an element of the polynomial ring* $\mathbb{R}[\alpha_1, \dots, \alpha_N]$ *(note that this is a unique factorisation domain (UFD) [10]),*
2. *the quotient field* $\mathbb{R}(\alpha_1, \dots, \alpha_N) := Quot(\mathbb{R}[\alpha_1, \dots, \alpha_N])$ *is called the* field of rational functions *in N variables,*

3. *two quotients $\frac{p}{q}$, $\frac{r}{s}$ are equivalent in $\mathbb{R}(\alpha_1,\ldots,\alpha_N)$, i.e. $\frac{p}{q}=\frac{r}{s}$, if and only if $ps=rq$. We say that a fraction $\frac{p}{q}$ "is in $\mathbb{R}$" or "constant" if $\frac{p}{q}=\frac{r}{1}$ for some $r\in\mathbb{R}$.*

For generalised weights w_i, $i\in\{1,\ldots,m\}$ we have $\frac{w_i}{\sum_{j=1}^{m} w_j}\in\mathbb{R}(\alpha_1,\ldots,\alpha_n)$. So we see that normalising a WSLTS (including generalised weights) leads in general to probabilities in the field of rational functions. In some lucky cases the probabilities are constant. We define:

Definition 3. *For a CASPA model C consisting of sequential submodels $\mathcal{S}_i$, $i\in\{1,\ldots,N\}$ we define its* relaxation *$\widetilde{C}$ as the model where every weight in a submodel $\mathcal{S}_i$ is multiplied by variable $\alpha_i>0$ ($i\in\{1,\ldots,N\}$). A CASPA model is called* scale-free *if after applying the maximal progress assumption all probabilities in the reachable part of the normalised WSLTS corresponding to $\widetilde{C}$ do not change regardless which actual values for α_i are chosen, i.e. the corresponding probabilities are constant in $\mathbb{R}(\alpha_1,\ldots,\alpha_N)$.*

The next definition introduces balancedness that – while on first glance a very different approach – will prove to be equivalent to the scale-free property.

Definition 4. *A CASPA model consisting of sequential submodels $\mathcal{S}_i$, $i\in\{1,\ldots,N\}$, is called* balanced *if, after applying the maximal progress assumption, for every reachable state s there is a set $syn(s)\subseteq\{\mathcal{S}_1,\ldots,\mathcal{S}_N\}$ such that for every transition emanating from s the set of participating submodels is equal to $syn(s)$.*

Lemma 1. *For a CASPA model C it holds that: C scale-free $\Leftrightarrow$ C balanced .*

Proof. Let $\mathcal{S}_i$, $i\in\{1,\ldots,N\}$ be the sequential submodels of C. Assume without loss of generality that the maximal progress assumption has been applied and the set of transitions has been restricted to the reachable subset. A general transition in the WSLTS of $\widetilde{C}$ emanating from s has the form $s\overset{a_I,c_I\cdot\alpha_I}{\dashrightarrow}s'$ where I is to be understood as the multi-index $(i_1,\ldots,i_M)$, $\forall k\in\{1,\ldots,M\}:i_k\in\{1,\ldots,N\}$ (M denotes the number of synchronising submodels). Furthermore, we have $a_I\in Act$, $c_I=\prod_{k=1}^{M}c_{i_k}\in\mathbb{R}$ and $\alpha_I:=\alpha_{i_1}\cdot\ldots\cdot\alpha_{i_M}$. Now the normalisation leads to the following fraction for the corresponding probability: $\frac{c_I\cdot\alpha_I}{c_I\cdot\alpha_I+\ldots+c_K\cdot\alpha_K}$. We see that the α product can only be cancelled out if and only if the multi-indices are the *same* for every summand in the denominator.

Remark 2. The idea of balanced models is quite classic. Without calling it this way, [21] uses a semantics that implies balancedness of all weighted decisions for the resulting models. There, only discrete time steps are considered.

For an even stronger formulation we use

Definition 5. *A transition is called* synchronised *if more than one submodel participate in it, otherwise it is called* unsynchronised*. A CASPA model is called* simple *if after maximal progress and reachability for every state s it holds that*

1. *there is at most one synchronised transition emanating from s and no unsynchronised transition, or*
2. *the transitions emanating from s are all are unsynchronised and belong to the* same *submodel.*

Lemma 2. *For a CASPA model C it holds that: C simple $\Rightarrow C$ balanced .*

Proof. Clear by definition.

Sometimes this definition of scale-freeness is too restrictive. Note that – similar to the elimination of vanishing states in Generalised Stochastic Petri Nets (GSPNs) – all vanishing states in a CASPA model may be eliminated in the case that no timeless trap is present[2] (cf. [18, 4]). We denote this process by $el(.)$ (note that this is well-defined up to bisimilarity [18]).

Definition 6. *A CASPA model C is called* quasi scale-free*, if in $el(\widetilde{C})$ all probabilities and rates are constant.*

Example 1. Let C be the example presented in Fig. 2. We see in Fig. 3a the corresponding WLSTS of the relaxation. It can immediately be seen that it is not balanced (syn(`(IDLE,IDLE,IDLE)`)), or equivalently multi-index α_I, is not constant) and therefore, by Lemma 1 not scale-free. The same can be seen in Fig. 3b as in the fractions emanating from `(IDLE,IDLE,IDLE)` only α_3 cancelled out, but nothing more. Fig. 3c shows the that the model is not even quasi scale-free, as the rates of the CTMC depend on α_1 and α_2.

Sometimes even not quasi scale-free models can be useful – if the measure that is to be computed does not depend on the weights chosen.

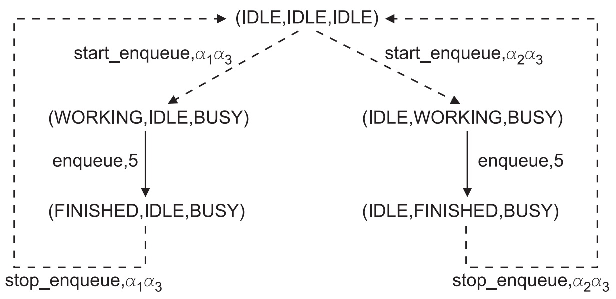
Definition 7. *Let $\mathcal{M}$ be a set of CASPA state measures. A CASPA model C without timeless traps is called* scale-free with respect to $\mathcal{M}$*, if the analysis results of $el(\widetilde{C})$ for every measure in $\mathcal{M}$ are constant in $\mathbb{R}(\alpha_1, \ldots, \alpha_N)$ (i.e. they do not depend on the scaling parameters α_i, $i \in \{1, \ldots, N\}$).*

Example 2. Let C be the model given in Fig. 2. Doing CTMC analysis on $el(\widetilde{C})$ we see that the results are `buf1working`$= \frac{\alpha_1}{\alpha_1+\alpha_2}$, `buf2working`$= \frac{\alpha_2}{\alpha_1+\alpha_2}$ and `any_working`$= \frac{\alpha_1+\alpha_2}{\alpha_1+\alpha_2} = 1$. So the model is only scale-free with respect to the trivial measure `any_working`.

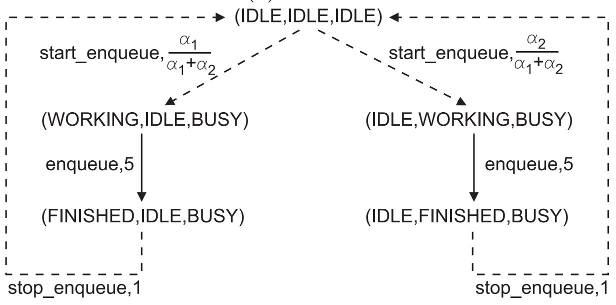
Lemma 3. *The following implications hold*

1. *"scale-free" $\Rightarrow$ "quasi scale-free"*
2. *"quasi scale-free" without timeless trap $\Rightarrow$ "scale-free with respect to all state measures"*

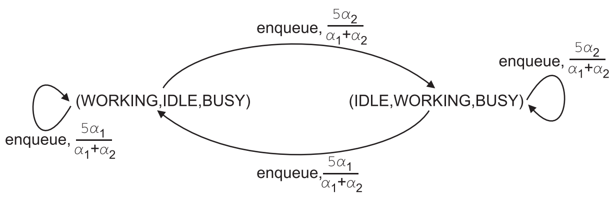
[2] In the case that timeless traps are present, we leave vanishing states with timeless self-loops with loop-probability 1 unchanged.


(IDLE,IDLE,IDLE)
start_enqueue,$\alpha_1\alpha_3$
start_enqueue,$\alpha_2\alpha_3$
(WORKING,IDLE,BUSY)
(IDLE,WORKING,BUSY)
enqueue,5
enqueue,5
(FINISHED,IDLE,BUSY)
(IDLE,FINISHED,BUSY)
stop_enqueue,$\alpha_1\alpha_3$
stop_enqueue,$\alpha_2\alpha_3$

(a) WSLTS



(b) WSLTS after normalisation


enqueue, $\frac{5\alpha_2}{\alpha_1+\alpha_2}$
enqueue, $\frac{5\alpha_2}{\alpha_1+\alpha_2}$
(WORKING,IDLE,BUSY)
(IDLE,WORKING,BUSY)
enqueue, $\frac{5\alpha_1}{\alpha_1+\alpha_2}$
enqueue,$\frac{5\alpha_1}{\alpha_1+\alpha_2}$

(c) CTMC

Fig. 3: State graphs for the example

$$\begin{tikzcd}
\text{“scale-free”} \arrow[r, Leftrightarrow, "\text{Lemma 1}"] \arrow[d, Rightarrow, "\text{Lemma 3}"'] & \text{“balanced”} & \text{“simple”} \arrow[l, Rightarrow, "\text{Lemma 2}"'] \\
\text{“quasi scale-free”} \arrow[d, Rightarrow, "\text{Lemma 3}"', "\text{no timeless trap}"] & & \\
\text{“scale-free w.r.t. } \mathcal{M}\text{”} & &
\end{tikzcd}$$

Fig. 4: Different notions of scale-freeness and their relations

Proof. The first part is clear by definition: If the normalised WSLTS does not depend on rescaling, also the result after elimination does not. Second part: If no timeless trap is present, all vanishing states can be eliminated (cf. [18, 4]) and so the result is a CTMC. If a CTMC does not depend on rescaling, also the analysis results will not.

Fig. 4 includes the implications and corresponding lemmata that we have shown.

4 Detecting different types of CASPA models

In this section we present methods to detect the different types of CASPA models defined in Sec. 3.

4.1 Detecting simple CASPA models

In general it can be demanding to verify that a CASPA model is indeed scale-free. It cannot be shown using only the syntax of the model, but we have to examine the state graph. The following notations will be useful:

Definition 8 (Local and global transitions). *A CASPA model C is called* locally/globally annotated*, if every immediate action label $a \in Act \setminus \{\tau\}$ has the suffix* `_loc` *for local transitions, i.e. transitions that only affect the local submodel or* `_glob`*, i.e. the transition is used for synchronisation with other components. The internal action τ is always local (τ can always be read as τ*`_loc`*). We define Act_{loc} (Act_{glob}) as the set of local (global) immediate actions defined in C — with suffix* `_loc` *(*`_glob`*) omitted. Immediate transitions that perform local (global) actions are called* local *(*global*)* transitions*.*

The local/global annotation can be used to check condition 1 of Definition 5. For condition 2 more information is needed. In order to distinguish local transitions according to the submodels they belong to, we define:

Definition 9. *A locally/globally annotated CASPA model is called* partitioned*, if every local action has an additional suffix that determines the submodel a local action acts in.*

Example 3. The partitioned transition system from the example in Fig. 1 is shown in Fig. 5. Clearly the model is not simple as there are two concurring global transitions and a local transition concurring with global transitions.

Example 4. We give also an annotated version of our running example in Fig. 6. The annotated version reveals that there are only global actions in the model. But now the simple condition is violated as state `(IDLE,IDLE,IDLE)` has two outgoing global transitions.

We already saw in Example 1 that our running example is not scale-free. But there – even if it was easy to see – we used the fact that one `start_enqueue` transition has participants `Robot` and `Buffer1`, while the other has participants `Robot` and `Buffer2`, which is hard to detect automatically. We will formalise the detection in the next section.

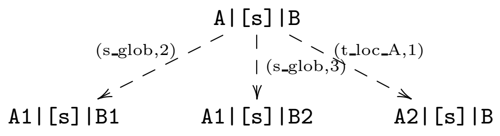


Fig. 5: Partitioned version of example from the introduction

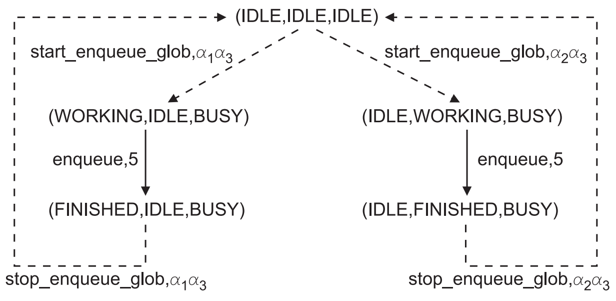

(IDLE,IDLE,IDLE)
start_enqueue_glob,$\alpha_1\alpha_3$
start_enqueue_glob,$\alpha_2\alpha_3$
(WORKING,IDLE,BUSY)
(IDLE,WORKING,BUSY)
enqueue,5
enqueue,5
(FINISHED,IDLE,BUSY)
(IDLE,FINISHED,BUSY)
stop_enqueue_glob,$\alpha_1\alpha_3$
stop_enqueue_glob,$\alpha_2\alpha_3$

Fig. 6: annotated WSLTS

4.2 Dectecting balanced CASPA models

For detecting balanced CASPA models, we introduce an additional signature parameter for every transition that can be derived directly from the syntax of a CASPA model. We need the following notation

Definition 10. *Let N be fixed and $\mathfrak{B}(N) := \cup_{i=1}^{N}\mathbb{B}^i$ the set of boolean strings of length up to N. There is a canonical outer product $\mathbb{B}^i \times \mathbb{B}^j \to \mathbb{B}^{i+j}$,*

$$((a_1, \ldots, a_i), (b_1, \ldots, b_j)) \mapsto (a_1, \ldots, a_i).(b_1, \ldots, b_j) := (a_1, \ldots, a_i, b_1, \ldots, b_j).$$

The set $\mathfrak{B}(N)$ will be called the set of patterns *for CASPA models with N submodels. For any CASPA process Q we define $|Q|$ as the number of sequential submodels contained in Q.*

The transition relation for a CASPA model with N submodels will be altered to $\dashrightarrow \subseteq S \times L_I \times \mathbb{R}^{>0} \times \mathfrak{B}(N) \times S$, where $\mathfrak{B}(N)$ describes the submodels that participate at a transition. Every participating submodel will be denoted by 1, while every non-participating submodel is set to 0. We will use 0^n to denote the boolean string $\underbrace{(0, \ldots, 0)}_{n \text{ times}}$.

We change the semantics according to Tab. 2 (only the modified semantic rules are depicted, $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}$ denote patterns). With this semantics at hand, it remains to check the closed model (after maximal progress and reachability) whether every vanishing state has its unique pattern of outgoing transitions.

$$\frac{}{(*b,\omega*);P \overset{(b,\omega,(1))}{\dashrightarrow} P}\; b \in L_I$$

$$\frac{P \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} P'}{P+Q \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} P'}\; b \in L_I \qquad \frac{Q \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} Q'}{P+Q \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} Q'}\; b \in L_I$$

$$\frac{P \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} P'}{P|[S]|Q \overset{(b,\omega,\mathcal{P}.0^{|Q|})}{\dashrightarrow} P'|[S]|Q}\; b \notin S \qquad \frac{Q \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} Q'}{P|[S]|Q \overset{(b,\omega,0^{|P|}.\mathcal{P})}{\dashrightarrow} P|[S]|Q'}\; b \notin S$$

$$\frac{P \overset{(s_b,\omega,\mathcal{P}_1)}{\dashrightarrow} P' \qquad Q \overset{(s_b,\sigma,\mathcal{P}_2)}{\dashrightarrow} Q'}{P|[S]|Q \overset{(s_b,\omega\cdot\sigma,\mathcal{P}_1.\mathcal{P}_2)}{\dashrightarrow} P'|[S]|Q'}\; s_b \in S$$

$$\frac{P \overset{(s_b,\omega,\mathcal{P})}{\dashrightarrow} P'}{\texttt{hide } S \texttt{ in } P \overset{(\tau,\omega,\mathcal{P})}{\dashrightarrow} \texttt{hide } S \texttt{ in } P'}\; s_b \in S \qquad \frac{P \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} P'}{\texttt{hide } S \texttt{ in } P \overset{(b,\omega,\mathcal{P})}{\dashrightarrow} \texttt{hide } S \texttt{ in } P'}\; b \notin S$$

Table 2: Changes to CASPA semantics for using patterns

Example 5. In our running example from Fig. 2 after maximal progress assumption there are two patterns for the reachable state `(IDLE,IDLE,IDLE)`, namely $(1,0,1)$ and $(0,1,1)$, so it is clear that it is not balanced.

Remark 3. In the pattern-based setting we can also characterise simple models quite easily if we define $|\ | : \mathfrak{B}(N) \to \{1,\ldots,N\}$, $(a_1,\ldots,a_j) \mapsto \sum_{i=1}^{j} a_i$ $(1 \leq j \leq N)$. We consider the reachable part of the state graph after maximal progress assumption (S being the set of reachable states). A model is simple if

– there exists a mapping $\mathcal{P} : S \to \mathfrak{B}(N)$ (i.e. for every state s there is a unique pattern $\mathcal{P}(s)$ such that all transitions emanating from s have this pattern), and
– for every state s with $|\mathcal{P}(s)| > 1$ there is only one transition emanating from s.

4.3 The quasi scale-free case

So far we have concentrated on the strong form of scale-freeness. For the quasi scale-free case we present two approaches: The first one, which is straightforward, is to carry out the elimination process with generalised probabilities calculated from generalised weights in $\mathbb{R}(\alpha_1,\ldots,\alpha_n)$ (again after maximal progress and reachability analysis). If after the elimination process all remaining rates and probabilities are constant, the model is quasi scale-free by definition.

The second approach uses the patterns introduced in Sec. 4.2. The idea is to find equivalent models under some appropriate equivalence relation that only have emanating transitions with one single pattern per state. We will now construct Markov Automata (MA, cf. [9]) that reflect the problem of unbalancedness by introducing nondeterminism. Informally, an MA is a 5-tuple $(S, \mathit{Act}, \twoheadrightarrow, \twoheadrightarrow\!\!\!\!\rightarrow, s_0)$ with state space S, action set Act and inital state s_0. MAs can express both, nondeterminism between action (immediate) transitions ($\twoheadrightarrow$), and races between Markovian (delay) transitions ($\twoheadrightarrow\!\!\!\!\rightarrow$).

Remember from Sec. 2 that for closed CASPA models all immediate action labels can safely be ignored. So we would like to stress that the assumption in the following definition poses no restriction for closed CASPA models. For the rest of this section we assume that whenever $C \in \mathcal{C}_{cl}$ then

1. all immediate action labels are equal to τ
2. the maximal progress assumption has been applied.

Definition 11. *Let $C \in \mathcal{C}_{cl}$ be a CASPA model (with N submodels) and S be its set of reachable states, $s_0 \in S$ the initial state, $\rightarrow$ the set of Markovian transitions and $\dashrightarrow$ the set of immediate transitions (annotated by patterns). Let $\mathfrak{P}(s) \subseteq \mathfrak{B}(N)$ be the set of patterns emanating from s (remember that for balanced models $|\mathfrak{P}(s)| = 1$ for every s). Then a corresponding MA can be generated in the following way:*

1. *For every $\mathcal{P} \in \mathfrak{P}(s)$ we calculate $P(s, \mathcal{P}, s') := \frac{\sum_{(s,\tau,\omega,\mathcal{P},s') \in \dashrightarrow} \omega}{\sum_{\exists t:(s,\tau,\omega,\mathcal{P},t) \in \dashrightarrow} \omega}$*
2. *$\twoheadrightarrow := \{(s, \tau, \mu) | s \in S, \mathcal{P} \in \mathfrak{P}(s), \mu = P(s, \mathcal{P}, .)\}$*
3. *$\twoheadrightarrow\!\!\!\!\!\rightarrow := \{(s, \lambda, s') | \exists a \in Act : (s, a, \lambda, s') \in \rightarrow\}$*

The corresponding MA $MA(C)$ is then defined as $(S, \{\tau\}, \twoheadrightarrow, \twoheadrightarrow\!\!\!\!\!\rightarrow, s_0)$.

Note that the sum in the numerator of item 1 has either one or zero elements. The combined effect of item 1 and item 2 is a pattern-wise normalisation of weights. This construction has interesting properties, as the following lemmata show.

Lemma 4. *For a CASPA model C the MA $MA(C)$ is always scale-free in the sense that it does not change when submodels are rescaled.*

Proof. This is clear by definition: Only transitions with the *same* pattern are merged into a distribution.

Lemma 5. *When Markovian action labels are ignored, the standard semantics of a CASPA model $C \in \mathcal{C}_{cl}$ describes a probabilistic execution fragment of $MA(C)$ in the sense of [19].*

Proof. We construct a probabilistic scheduler that yields the desired behaviour. We have to show that for every state s every distribution on successor states of s can be realised by some convex combination of the transitions in $\twoheadrightarrow$ emanating from s. Suppose s has the outgoing transitions $s \overset{(\tau,\omega_i,\mathcal{P}_i)}{\dashrightarrow} s_i$, $i \in \{1, \ldots, I(s)\}$, then the probability distribution on the successor states s_i is given as $P(s_i) := \frac{\omega_i}{\sum_{j=1}^{I(s)} \omega_j}$ by standard semantics. Assume without loss of generality (otherwise reorder) that $|\mathfrak{P}(s)| = K$ and that transitions with index in $\{k_m, \ldots, k_{m+1} - 1\}$ belong to pattern $\mathcal{P}_m$, $k_1 = 1$, $k_{K+1} = I(s) + 1$, $k_h < k_j$ for $h < j$. Now the probabilities – in $MA(C)$ – when using the m-th pattern are $P(s, \mathcal{P}_m, s_i) = \frac{\omega_i}{\sum_{j=k_m}^{k_{m+1}-1} \omega_j}$, so it is easy to see that the coefficients c_m for the convex combination have to be chosen as $\frac{\sum_{j=k_m}^{k_{m+1}-1} \omega_j}{\sum_{j=1}^{I(s)} \omega_j}$ to get the distribution that is determined by standard CASPA semantics.

One could now reduce $MA(C)$ modulo weak bisimulation (a relation on distributions) and check whether there is a *deterministic* representative in the equivalence class modulo weak MA bisimilarity, because this would mean that in a nondeterministic choice it does not matter which action is taken. The problem is that this is not the whole story. Looking at our running example, a possible result would be an MA with just one state with a Markovian loop with rate 5, which is clearly deterministic. But we saw earlier that our example is not scale-free!

The solution is that for getting results on the state measures (which only can be specified for stable states) we are not allowed to lump together different stable states (which would be done to get the Markovian loop above). This can be realised by a sort of AP (atomic proposition) bisimulation in the spirit of F-bisimulation [3] or Markov-AP bisimulation [6]: If all stable states have different APs, they can never be grouped together! We recall from [9] that weak MA bisimulation $\approx$ induces an equivalence relation $\approx_\Delta$ on states by defining $x \approx_\Delta y :\Leftrightarrow \Delta_x \approx \Delta_y$ (where Δ_x denotes the Dirac distribution at state x). Therefore, with an appropriate extension of $\approx_\Delta$ to an AP bisimulation $\approx_\Delta^{AP}$ we can conclude that if our MA is weakly AP bisimilar to a deterministic MA, the corresponding CASPA model is quasi scale-free, i.e. this is a sufficient criterion. Note that weak AP bisimulation also generalises the elimination of vanishing states (cf. Sec. V in [9]). The drawback of this approach is that no weak MA bisimulation algorithm has been published so far, but at least we can state:

Lemma 6. *If for a CASPA model $C \in \mathcal{C}_{cl}$ the resulting MA $MA(C) = (S, \{\tau\}, \multimap, \twoheadrightarrow, s_0)$ is weakly AP bisimilar to a deterministic MA $M' = (S, \{\tau\}, \multimap', \twoheadrightarrow, s_0)$, where $\multimap' \subseteq \multimap$, then C is quasi scale-free.*

Proof. Assume that all stable states in S are equipped with different atomic propositions. Weak MA AP bisimilarity ensures that every nondeterministic choice in $MA(C)$ is equivalent to a deterministic one in M'. So each decision "does not matter" with respect to our equivalence relation. Especially it is not important which probabilistic execution fragment of $MA(C)$ is chosen by the probabilistic scheduler induced by CASPA semantics according to Lemma 5. As by Lemma 4 the distributions in $MA(C)$ do not depend on rescaling, also the distributions in M' cannot depend on rescaling of weights of submodels.

We use the diagram in Fig. 7 to prove the claim. Firstly we show that commutativity of the diagram implies our claim. Secondly we prove the commutativity. The diagram is explained as follows: $el(.)$ denotes the elimination of vanishing states (this is also defined for deterministic MA, cf. Sec. V in [9]), $det(.)$ denotes the mapping to a weakly AP bisimilar deterministic representative M'. The canonical mapping from CASPA models to MA by transforming the reachable state graph to MA (disregarding labels of Markovian actions) is denoted by $canon(.)$. Now we observe that quasi scale-freeness can be identified as $canon \circ el(C)$ having constant rates and probabilities – no matter, which rescaling factors for submodels are chosen. But the mapping $el \circ det \circ MA(C)$ surely produces constant rates and probabilities. Commutativity of the diagram follows from the above observation that the actual probabilistic execution fragment of $MA(C)$ does not play any role and by Theorem 7 in [9].

$$\begin{tikzcd}
CASPA \arrow[r, "MA(.)"] \arrow[d, "el(.)"] & MA \arrow[r, "\exists det(.)", "\approx_{\Delta}^{AP}"'] & MA \arrow[d, "el(.)", "\approx_{\Delta}^{AP}"'] \\
CASPA \arrow[rr, "canon(.)"] & & MA/\approx_{\Delta}^{AP}
\end{tikzcd}$$

Fig. 7: Elimination and weak AP bisimulation

4.4 Scale-freeness with respect to some measure

The MA-based approach sketched in Sec. 4.3 can also be adapted to the "with respect to some measure" case. Using different state labels (characterising some given measures) for the AP bisimulation, one can use this machinery to check for the weaker case of scale-freeness with respect to some measure $\mathcal{M}$. The basic difference between the two settings is that in the quasi scale-free case *all* stable states get different labels, while in the weaker setting some stable states might share the same label.

5 Conclusion

We have shown necessary and sufficient criteria for scale-free CASPA models and related different kinds of scale-freeness. Patterns have been introduced to easily recognise participating submodels. From the state graph annotated with patterns one can check scale-freeness. Patterns have also been used to transform the problem to the domain of Markov Automata in order to check the quasi scale-free case.

Acknowledgements: We would like to thank the anonymous reviewers for their valuable comments, and DFG who supported this work under grant SI 710/7-1 and for partial support by DFG/NWO Bilateral Research Programme ROCKS.

References

1. M. Ajmone Marsan, G. Balbo, G. Conte, Donatelli S., and Franceschinis G. *Modelling with Generalized Stochastic Petri Nets.* Wiley Series in Parallel Computing, 1995.
2. J. Bachmann, M. Riedl, J. Schuster, and M. Siegle. An Efficient Symbolic Elimination Algorithm for the Stochastic Process Algebra tool CASPA. In *SOFSEM 2009: Theory and Practice of Computer Science*, pages 485–496. Springer, LNCS 5404, 2009.

3. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(7), 2003.
4. F. Bause. No Way Out ∞ The Timeless Trap. *Petri Net Newsletters*, 37:4–8, 1990.
5. M. Bernardo and R. Gorrieri. A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time. *Theoretical Computer Science*, 202(1-2):1–54, 1998.
6. S. Blom, B. Haverkort, M. Kuntz, and J. van de Pol. Distributed Markovian Bisimulation Reduction aimed at CSL Model Checking. *Electron. Notes Theor. Comput. Sci.*, 220:35–50, December 2008.
7. J. Bradley. Semi-Markov PEPA: Modelling with Generally Distributed Actions. *International Journal of Simulation*, 6(3–4):43–51, February 2005.
8. P. D'Argenio, H. Hermanns, and J.-P. Katoen. On generative parallel composition. *Electr. Notes Theor. Comput. Sci.*, 22:30–54, 1999.
9. C. Eisentraut, H. Hermanns, and L. Zhang. On probabilistic automata in continuous time. In *Proceedings of the 2010 25th Annual IEEE Symposium on Logic in Computer Science*, LICS '10, pages 342–351. IEEE Computer Society, 2010.
10. W. Fulton. *Algebraic Curves.* Benjamin, 1969.
11. R. Hayden, J. Bradley, and A. Clark. Performance specification and evaluation with Unified Stochastic Probes and fluid analysis. *IEEE Transactions on Software Engineering*, December 2011. In press.
12. H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *Lecture Notes in Computer Science.* Springer, 2002.
13. J. Hillston. *A Compositional Approach to Performance Modelling.* Cambridge University Press, 1996.
14. M. Kuntz, M. Siegle, and E. Werner. Symbolic Performance and Dependability Evaluation with the Tool CASPA. In *Applying Formal Methods: Testing, Performance and M/E Commerce: FORTE 2004 Workshops, European Performance Engineering Workshop*, pages 293–307. Springer, LNCS 3236, 2004.
15. G. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.
16. M. Rettelbach. Probabilistic Branching in Markovian Process Algebras. *The Computer Journal*, 38(7):590–599, 1995.
17. M. Riedl, J. Schuster, and M. Siegle. Recent extensions to the stochastic process algebra tool CASPA. In *5th International Conference on the Quantitative Evaluation of SysTems (QEST'08)*, pages 113–114. IEEE Computer Society, 2008.
18. J. Schuster. *Towards faster numerical solution of Continuous Time Markov Chains stored by symbolic data structures.* PhD thesis, Universität der Bundeswehr München, 2011.
19. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1995.
20. E. Teruel, G. Franceschinis, and M. De Pierro. Clarifying the priority specification of GSPN: Detached priorities. In *Proceedings of the 8th International Workshop on Petri Nets and Performance Models (PNPM '99)*, pages 114 –123. IEEE Computer Society, 1999.
21. C. Tofts. Processes with probabilities, priority and time. *Formal Aspects of Computing*, 6:536–564, 1994. 10.1007/BF01211867.