Revisiting Generalised Stochastic Petri Nets

New Semantics and Analysis Algorithms

Joost-Pieter Katoen

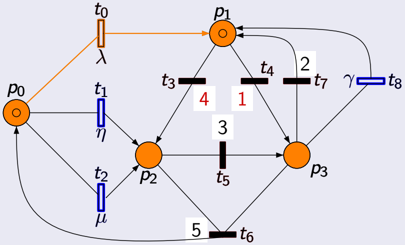
RWTH Aachen University, Germany
University of Twente, The Netherlands

IFIP WG 2.2 Meeting, Amsterdam, The Netherlands

This talk in a nutshell

Generalised stochastic Petri nets (GSPNs) [Ajmone Marsan et al, 1984]

- Places
- Timed transitions (rates)
- Immediate transitions
- Input, output, inhibitor arcs
- Tokens •



Semantics: play token game

Resolving the conflict?

- So far: use weightsweights
- Drawbacks: Which weights? Strange effects! Trustworthy analysis?
- New: Don't care. Keep it as is. (without abandoning weights)

Outline of the talk

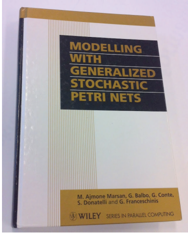
Overview

Historical perspective

1973	Timed Petri Nets	[Noe & Nutt]
1980	Stochastic Petri Nets	[Molloy, Natkin, Symons]
1984	Generalized Stochastic Petri Nets	[Ajmone Marsan, Conte & Balbo]
1991	GSPN Reward Nets	[Ciardo, Muppala & Trivedi]
1994	Non-Markovian Stochastic Petri Nets	[Bobbio, German et al.]
1995	Modeling with Generalized Stochastic Petri Nets	[Ajmone Marsan et al.]

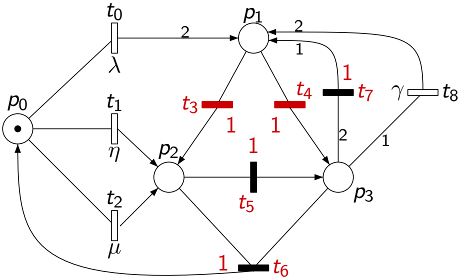
A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems

MARCO AJMONE MARSAN and GIANNI CONTE
Politecnico di Torino, Turin, Italy
and
GIANFRANCO BALBO
Universita' di Torino, Turin, Italy



Generalised stochastic Petri nets

- Places
- Timed transitions
- Immediate transitions
- Weights
- Input, output, inhibitor arcs
- Tokens •



Maximal progress: immediate transitions have priority over timed ones.

Removal of reachable vanishing markings in marking graph yields a continuous-time Markov chain.

Applicability

Quantitative measures

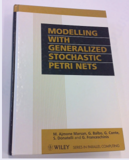
1. the reachability probability of a given marking
2. the probability to be in a marking after t time units — transient
3. the probability to be in a marking on the long run — stationary
4. the probability to satisfy a temporal logic formula — CSL model checking

All these quantities can be computed efficiently and are tool-supported.

A caveat

The presence of confused subnets
of immediate transitions within a GSPN
is an undesirable property of the model.

Ajmone Marsan et al. (1995)

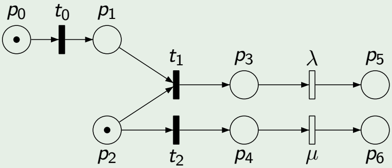


Confusion

What is confusion?

Confusion arises if there is a reachable marking in which multiple non-conflicting immediate transitions are simultaneously enabled.

A simple confused GSPN



- Transitions t_0 and t_2 are concurrent
- If t_2 fires first, no conflict arises
- If t_0 fires first, a conflict $t_1 \frown t_2$ arises

In marking $p_1 + p_6$ one cannot conclude whether a conflict had to be resolved.

This situation is called confusion.

Why is confusion problematic?

No stochastic process

The reachability graph of a confused net is not a continuous-time Markov chain but a stochastic decision process. Standard CTMC analysis is not possible.

It is meaningless to consider

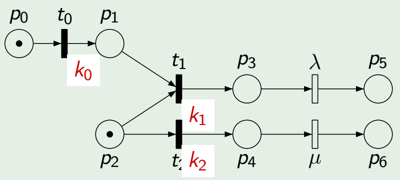
1. the reachability probability of a given marking
2. the probability to be in a marking after t time units
3. the probability to be in a marking on the long run

These quantities are all subject to the resolution of nondeterminism.

Classical GSPN approach: resolve nondeterminism by using weights.

Weighted immediate transitions

A simple weighted GSPN



- Transition t_i has weight $k_i \in \mathbb{N}_{>0}$
- t_2 fires first with probability $\frac{k_2}{k_0+k_2}$
- t_0 fires first with probability $\frac{k_0}{k_0+k_2}$
- Concurrency is thus resolved probabilistically

$$\Pr\{\Diamond(p_1+p_6)\} = \underbrace{\frac{k_2}{k_0+k_2}}_{t_2 \text{ before } t_0} + \underbrace{\frac{k_0}{k_0+k_2} \cdot \frac{k_2}{k_1+k_2}}_{\substack{t_0 \text{ before } t_2 \\ \text{and } t_2 \text{ before } t_1}}$$

Note the influence of k_0 on this quantity.

Drawbacks of weights

How to get adequate weights?

For conflicting transitions this is mostly simple, but not for confused ones.

But: weight values are fundamental for the quantitative evaluation.

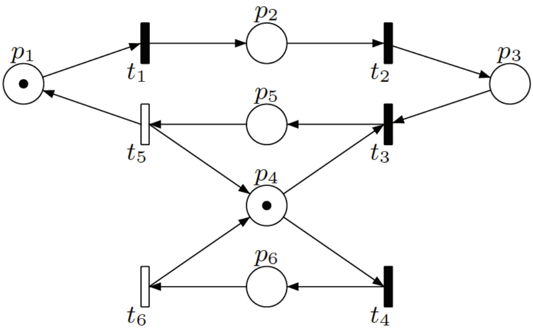
Biased analysis

Quantitative results are subject to a specific weight assignment. This bias is often neglected. (see later case study)

Unexpected effects

Splitting or deleting an immediate transition "has drastic effects on the values of the results obtained from the quantitative evaluation".

Weights are not so innocent



Assume all rates and weights equal one. Then $LRA(\ldots + p_5) = \frac{4}{11}$.

Deleting p_2 and immediate transition t_2 yields $LRA(\ldots + p_5) = \frac{4}{10}$.

Workarounds

Some approaches

1. Net-level reasoning [Chiola et al., 1993]
   - signals which immediate transitions may become in conflict
   - simple and efficient, but incomplete
   - translational semantics rely on this: no confusion-free proof
2. State-space reasoning [Ciardo et al., 1996]
   - exact but computationally involved
3. Net-level restrictions [Teruel et al., 2003]
   - ensure different priorities for conflicting transitions
   - but can provide false ("spurious") alarms

"Well-specified" checks exist for SANs [Deavours & Sanders, 1999]

Our approach: no checks. No restrictions. All nets are well-defined.

Overview

Recent developments

2010	Markov Automata	[Eisentraut et al.]
2011	Semantics of Markov Automata	[Deng & Hennessy]
2012	Quantitative Analysis of Simple MA	[Katoen et al.]
2012	Weak Bisimulation Minimisation of MA	[Turrini & Hermanns]
2012	Quantitative Analysis of MA	[Guck & Katoen]
2012	Efficient Generation of MA	[Timmer et al.]
today	New GSPN Semantics, Analysis Algorithms, and Tool	

2010 25th Annual IEEE Symposium on Logic in Computer Science

On Probabilistic Automata in Continuous Time

Christian Eisentraut, *Saarland University –*
Holger Hermanns, *Saarland University – Computer Science,*
Lijun Zhang, *DTU Informatics*

```
NetOne(P0 : Nat, P1 : Nat, P2 : Nat, P3 : Nat) =
      ((P0 = 0) & (P1 = 0) & (P2 = 0) & (P3 = 1) => reach(
    + (P0 >= 1 => (1.0) . NetOne[P0 := P0 - 1, P1 := P1 +
    + (P0 >= 1 => (1.0) . NetOne[P0 := P0 - 1, P2 := P2 +
    + (P0 >= 1 => (1.0) . NetOne[P0 := P0 - 1, P2 := P2 +
    + (P3 >= 1 => (1.0) . NetOne[P1 := P1 + 2, P3 := P3 -
    + ((P1 >= 1) | (P1 >= 1) | (P2 >= 1) | ((P2 >= 1) & (P

Initial state: NetOne(1, 0, 0, 0)
Number of states: 10
Number of transitions: 12
```

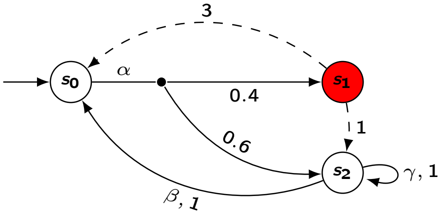
Markov automata

[Eisentraut et al., LICS 2010]

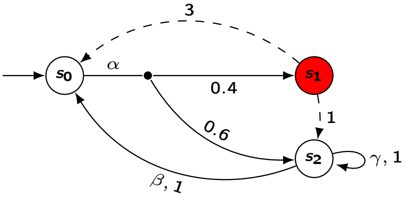
A Markov automaton M is a tuple $(S, Act, \longrightarrow, \Longrightarrow, s_0)$ where

- S is a nonempty set of states with initial state $s_0 \in S$
- Act is a finite set of actions; τ is an internal action
- $\longrightarrow \subseteq S \times Act \times Dist(S)$ is a set of action transitions, and
- $\Longrightarrow \subseteq S \times \mathbb{R}_{>0} \times S$ is a set of Markovian transitions

  such that there is at most one $r \in \mathbb{R}_{>0}$ such that $s \overset{r}{\Longrightarrow} s'$



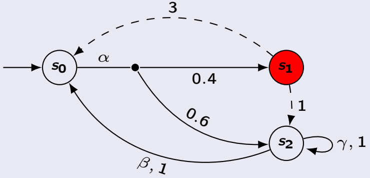
Déjà vu?



- $\Longrightarrow\ = \emptyset$ yields probabilistic automata.
- $\Longrightarrow\ = \emptyset$ and $\longrightarrow$ is deterministic yields Markov decision processes.
- $\Longrightarrow\ = \emptyset$, $\longrightarrow$ is deterministic, and $|Act| = 1$ yields Markov chains.
- $\Longrightarrow\ = \emptyset$ and $\longrightarrow$ is Dirac yields labeled transition systems.
- $\longrightarrow$ is Dirac yields interactive Markov chains.
- $\longrightarrow\ = \emptyset$ yields continuous-time Markov chains.

Semantics

- s is Markovian if $s \Longrightarrow$ and $s \not\longrightarrow$
- s is probabilistic if $s \not\Longrightarrow$ and $s \longrightarrow$
- s is hybrid if $s \Longrightarrow$ and $s \longrightarrow$
- s is timelock if $s \not\Longrightarrow$ and $s \not\longrightarrow$



For Markovian s, let:

- $\mathbf{r}(s, s')$ be the rate to move from s to s',
- $E(s) = \sum_{s' \in S} \mathbf{r}(s, s')$ be the exit rate of s
- $\mathbf{p}(s, s') = \frac{\mathbf{r}(s,s')}{E(s)}$ is the probability to move from s to s'

$\mathbf{r}(s_1, s_0) = 3,\ E(s_1) = 1 + 3 = 4$ and $\mathbf{p}(s_1, s_0) = \frac{3}{4}$ and $\mathbf{p}(s_1, s_2) = \frac{1}{4}$.

Maximal progress assumption

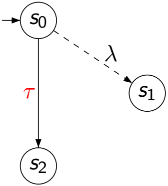
Justification

1. Internal (action) transitions are labeled with the action τ.
2. These transitions will not be subject to interaction.
3. They cannot be delayed by other components.
4. Thus, internal interactive transitions can trigger immediately.
5. But, almost surely no Markovian transition occurs immediately.

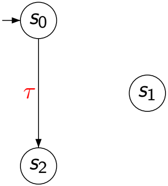
Maximal progress assumption

Internal action transitions take precedence over Markovian ones.

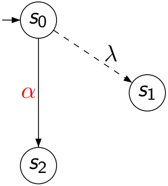
Maximal progress assumption



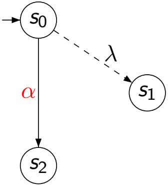
reduces to



But as visible actions may be subject to delaying by other components:



remains



Parallel composition

The *composition* of M_1 and M_2 wrt. $A \subseteq (Act_1 \cup Act_2) \setminus \{ \tau \}$ is:

$$M_1 \,||_A\, M_2 = (S_1 \times S_2, Act_1 \cup Act_2, \longrightarrow, \Longrightarrow, (s_{0,1}, s_{0,2}))$$

where $\longrightarrow$ and $\Longrightarrow$ are defined as the smallest relations satisfying:

$$\text{(SYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } s_2 \xrightarrow{\alpha}_2 \mu_2 \text{ and } \alpha \in A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \mu_2}$$

$$\text{(ASYNC)}\ \frac{s_1 \xrightarrow{\alpha}_1 \mu_1 \text{ and } \alpha \notin A}{(s_1, s_2) \xrightarrow{\alpha} \mu_1 \cdot \Delta_{s_2}}$$

$$\text{(DELAY)}\ \frac{s_1 \xRightarrow{\lambda}_1 s_1'}{(s_1, s_2) \xRightarrow{\lambda} (s_1', s_2)} \quad \text{and} \quad \frac{s_1 \xRightarrow{\lambda}_1 s_1 \text{ and } s_2 \xRightarrow{\lambda'}_2 s_2}{(s_1, s_2) \xRightarrow{\lambda+\lambda'} (s_1, s_2)}$$

Déjà vu?

- if M_1 and M_2 are LTSs, $||_A$ is TCSP-composition
- if M_1 and M_2 are PA, $||_A$ is PA-composition
- if M_1 and M_2 are MCs over Act, $||_{Act}$ is PCCS-composition
- if M_1 and M_2 are CTMCs, $||_\emptyset$ is independent parallelism
- if M_1 and M_2 are IMCs, $||_A$ is IMC-composition

Thus:

Parallel composition of MA is backward compatible with well-understood composition operators.

Hiding

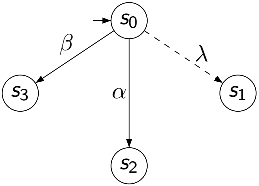
What is hiding?

Hiding the actions $A \subseteq Act \setminus \{ \tau \}$ in $M = (S, Act, \longrightarrow, \Longrightarrow, s_0)$ yields $M \setminus A = (S, Act \setminus A, \longrightarrow', \Longrightarrow, s_0)$ where $\longrightarrow'$ is defined by:

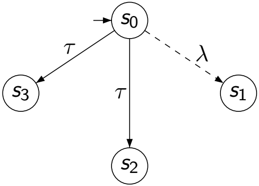
1. $s \xrightarrow{\alpha} \mu$ and $\alpha \notin A$ implies $s \xrightarrow{\alpha}' \mu$, and
2. $s \xrightarrow{\alpha} \mu$ and $\alpha \in A$ implies $s \xrightarrow{\tau}' \mu$.

- Hiding transforms α-transitions with $\alpha \in A$ into τ-transitions.
- This may enable maximal progress reduction.

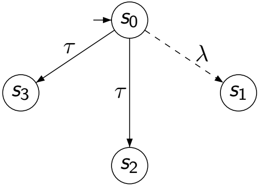
Hiding and maximal progress



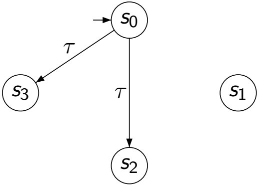
hiding $\{ \alpha, \beta \}$ yields



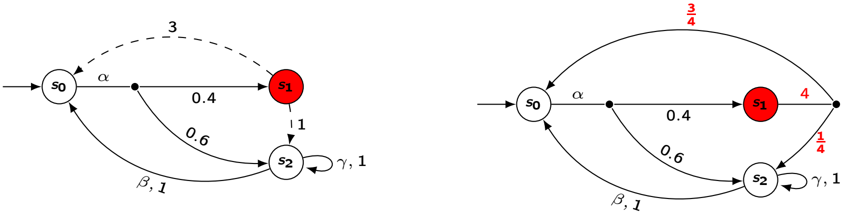
Applying maximal progress reduction yields:



reduces to



Bisimulation



Bisimulation

Equivalence $R \subseteq S \times S$ is a *bisimulation* if for all $(s, t) \in R$:

$$\forall \delta \in Act \cup \mathbb{R}_{>0}: s \xrightarrow{\delta} \mu \text{ implies } t \xrightarrow{\delta} \nu \text{ with } \forall C \in S/R : \mu(C) = \nu(C).$$

Let $\sim$ be the largest bisimulation relation.

Congruence

$\sim$ is a congruence wrt. parallel composition and hiding.

Déjà vu?

- if M is an LTS, $\sim$ is Milner's bisimulation
- if M is a PA, $\sim$ is Segala's bisimulation
- if M is an MCs, $\sim$ is lumpability
- if M is a CTMC, $\sim$ is lumping equivalence
- if M is an IMC, $\sim$ is Hermanns' bisimulation

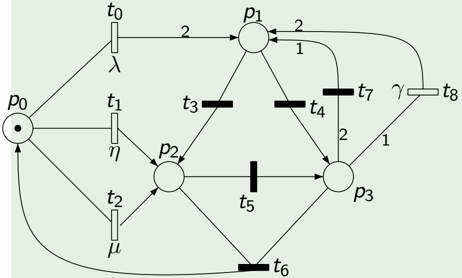
Thus:

$\sim$ on MA is backward compatible with well-understood bisimulations.

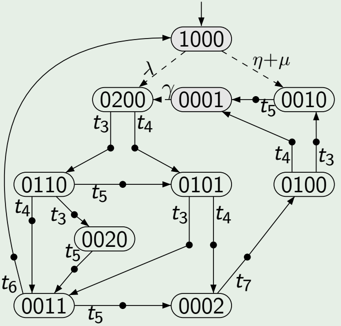
Variants like weak bismulation, simulation pre-orders can also be defined.

GSPN marking graphs are Markov automata!

A confused GSPN:

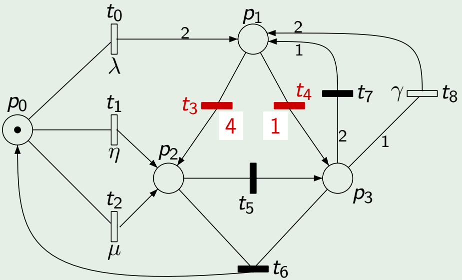


Its semantics:

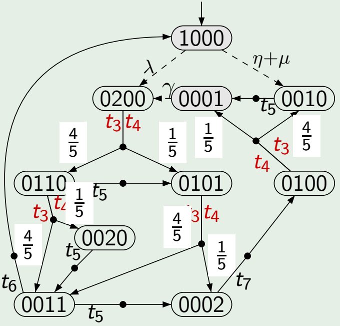


Adding some weights

A confused GSPN with weights:



Its semantics:



Claim:

This yields a truly simple GSPN semantics.

Backward compatibility

New GSPN semantics

Advantages

- It is truly simple
- It is intuitive
- It is compositional
- It is backward compatible
- Allows compositional reduction
- No restrictions on net level

But:

How to quantitatively analyse these stochastic decision processes?
Steady-state? Transient? Expected time?

Overview

Closed Markov automata

Model to be analysed

Typical structure:

$$\left(M_1 \,||_{A_1}\, M_2 \,||_{A_2}\, \ldots \,||_{A_{n-1}}\, M_n\right) \setminus A$$

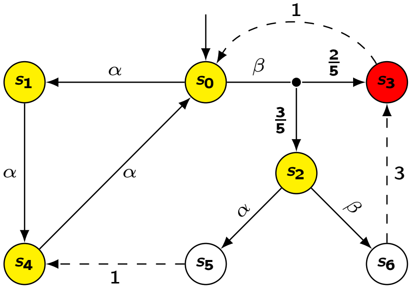
where A is the union of all visible actions, i.e., $A = \bigcup_{i=1}^{n-1} Act_i \setminus \{\, \tau \,\}$.

It is closed, as no action is subject to further interaction.

States have either only Markovian or only action transitions.

Every GSPN yields a closed Markov automaton.

Expected time



- Start state s_0
- Goal states $G = \{ s_3 \}$
- Expected time from s_0 to G?
- $eT^{\max}(s_0, \Diamond G) = \infty$
- $eT^{\min}(s_0, \Diamond G) = \frac{2}{5} \cdot 0 + \frac{3}{5} \cdot \frac{1}{3}$

Nondeterminism

Due to nondeterminism, the expected time to reach G is not uniquely defined. It depends on the choices in states s_0 and s_2. Approach: consider expected time under all policies! This yields bounds. Adding weights yields tighter bounds.

Policies

- A policy describes how all nondeterminism is resolved.
- It maps any finite path onto an enabled transition in its last state.
- A policy may make a choice on the basis of all information in a path: the visited states, their order, the state delays, and so on.
- We use deterministic positional policies.
- They always take the same decision in a state.

Expected time

For path π let the random variable V_G be the first hitting time of G:

$$V_G(\pi) = \min\{\, t \in \mathbb{R}_{\geqslant 0} \mid G \cap \pi@t \neq \emptyset \,\}$$

Expected time

The expected time to reach G from s for policy P is:

$$eT_P(s, \Diamond G) \;=\; \mathbb{E}_{s,P}(V_G) \;=\; \int_{Paths(s)} V_G(\pi) \Pr_{s,P}(d\pi)$$

The minimal expected time to reach G from s is:

$$eT^{\min}(s, \Diamond G) \;=\; \inf_P eT_P(s, \Diamond G)$$

Fixpoint theorem

$$eT^{\min}(s, \Diamond G) \;=\; \inf_P eT_P(s, \Diamond G) \;=\; \inf_P \int_{Paths(s)} V_G(\pi) \Pr_{s,P}(d\pi)$$

Theorem

$eT^{\min}(s, \Diamond G)$ is the unique fixpoint of the Bellman operator:

$$[L(v)](s) \;=\; \begin{cases} \dfrac{1}{E(s)} + \sum_{s' \in S} \mathbf{p}(s, s') \cdot v(s') & \text{if } s \in MS - G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_\alpha(s') \cdot v(s') & \text{if } s \in PS - G \\ 0 & \text{if } s \in G \end{cases}$$

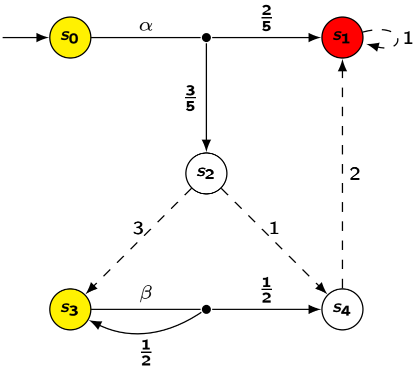
Exceptions

States on Zeno cycles and states that cannot reach G yield value ∞.

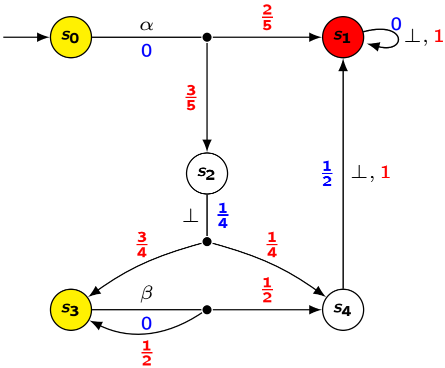
,

Reduction to SSP problem

Example Markov automaton:



Its induced SSP instance:



$$\mathbf{p}(s,\sigma,s') = \begin{cases} \frac{\mathbf{r}(s,s')}{E(s)} & \text{if } \sigma = \bot \\ \mu(s') & \text{if } s \xrightarrow{\sigma} \mu \\ 0 & \text{otherwise} \end{cases}$$

$$c(s,\sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \notin G, \sigma = \bot \\ 0 & \text{otherwise} \end{cases}$$

Solving SSP

$$[L(v)](s) = \begin{cases} \frac{1}{E(s)} + \sum_{s' \in S} \mathbf{p}(s,s') \cdot v(s') & \text{if } s \in MS - G \\ \min_{\alpha \in Act(s)} \sum_{s' \in S} \mu_\alpha(s) \cdot v(s') & \text{if } s \in PS - G \\ 0 & \text{if } s \in G \end{cases}$$

LP problem [Bertsekas & Tsitsiklis, 1991]

$eT^{\min}(s, \Diamond G)$ is the solution of the following LP problem:

$$\max \sum_{s \in S} x_s$$

$$\begin{aligned} x_{s_i} &\leqslant \frac{1}{E(s_i)} + \sum_{s' \in S} \mathbf{p}(s_i, \perp, s') \cdot x_{s'} && \text{if } s_i \in MS - G \\ x_{s_i} &\leqslant \min_{\alpha \in Act(s_i)} \sum_{s' \in S} \mathbf{p}(s_i, \alpha, s') \cdot x_{s'} && \text{if } s_i \in PS - G \\ x_{s_i} &= 0 && \text{if } s_i \in G \end{aligned}$$

Expected time analysis: synopsis

Minimal and maximal expected time

1. Make all states in G absorbing
2. Transform the Markov automaton to an SSP problem
3. Solve the SSP problem by linear programming

Positional policies suffice

There is a positional policy that yields $eT^{\min}(s, \Diamond G)$.

Long run average

$A_{G,t}$ is the fraction of time spent in $G \subseteq MS$ up to time t along path π:

$$A_{G,t}(\pi) = \frac{1}{t} \int_0^t \mathbf{1}_G(\pi@u)\, du \quad \text{and} \quad A_G(\pi) = \lim_{t \to \infty} A_{G,t}(\pi)$$

Long run average

The long-run average time spent in G starting from s under policy P:

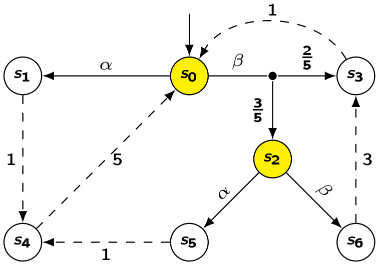
$$LRA_P(s, G) \;=\; \mathbb{E}_{s,P}(A_G) \;=\; \int_{Paths(s)} A_G(\pi) \Pr_{s,P}(d\pi)$$

The minimal long-run average time spent in G from s is:

$$LRA^{\min}(s, G) \;=\; \inf_P \, LRA_P(s, G)$$

Unichain Markov automata

MA M is unichain iff for all positional policies, M is strongly connected.



- $P(s_0) = \alpha$ yields $\{ s_0, s_1, s_4 \}$
- $P(s_0) = \beta$
  - $P(s_2) = \alpha$ yields $S \setminus \{ s_6 \}$
  - $P(s_2) = \beta$ yields $\{ s_0, s_2, s_3, s_6 \}$

$\Rightarrow$ this is a unichain

Reduction to long-run ratio objectives

Recall the Markov decision process, and consider the two cost functions:

$$c_G(s,\sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS \cap G, \sigma = \bot \\ 0 & \text{otherwise} \end{cases} \quad c(s,\sigma) = \begin{cases} \frac{1}{E(s)} & \text{if } s \in MS, \sigma = \bot \\ 0 & \text{otherwise} \end{cases}$$

Reduction to long-run ratio objective

For unichain MA M, $LRA^{\min}(s, G)$ equals the minimal long-run ratio between the accumulated costs over c_G and c.

Corollary

For every unichain MA, there is a positional policy that yields $LRA^{\min}(G)$.

Long-run ratio objectives

Long-run ratio

The long-run ratio between costs k and c along path π is:

$$R(\pi) \;=\; \lim_{n \to \infty} \frac{\sum_{i=0}^{n-1} k(s_i, \sigma_i)}{\sum_{i=0}^{n-1} c(s_i, \sigma_i)}$$

where $k(s_i, \sigma_i)$ is the cost k in state s_i on selecting σ_i (and similar for c).

Example

steps n	0	1	2	3	4	5	......
costs k	0	2	4	4	10	2	......
costs c	0	4	2	2	0	3	......
ratio R up to n		$\frac{1}{2}$	1	$\frac{5}{4}$	$\frac{5}{2}$	2	......

If it is not a unichain

Theorem

Let $M_1, \ldots, M_k$ be the maximal end components of MA M with state space $S_1, \ldots, S_k$. Then:

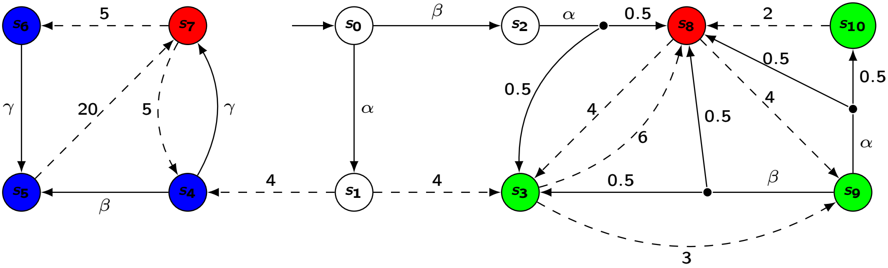
$$\text{LRA}^{\min}(s, G) \;=\; inf_P \sum_{i=1}^{k} LRA_i^{\min}(G) \cdot \Pr_P(s \models \Diamond\Box S_i)$$

Algorithm

1. Determine the maximal end components $\{M_1, \ldots, M_k\}$ of the MA M.
2. Determine $LRA^{\min}(G)$ for each M_j.
3. Solve a stochastic shortest path problem.

Corollary

Example



- $LRA_1^{\text{min}}(G) = \frac{2}{3}$
- $LRA_2^{\text{min}}(G) = \frac{9}{25}$
- $LRA^{\text{min}}(s_0, G) = \frac{9}{25}$
- $P(s_0) = \beta$ and $P(s_9) = \alpha$
- $LRA_1^{\text{max}}(G) = \frac{4}{5}$
- $LRA_2^{\text{max}}(G) = \frac{5}{9}$
- $LRA^{\text{max}}(s_0, G) = \frac{1}{2} \cdot \frac{4}{5} + \frac{1}{2} \cdot \frac{5}{9} = \frac{61}{90}$
- $P(s_0) = \alpha$, $P(s_9) = \beta$ and $P(s_4) = \gamma$

Timed reachability

What is timed reachability?

Given a set G of goal states, start state s and time interval I, what is the minimal (or maximal) probability to reach G at some time point in I?

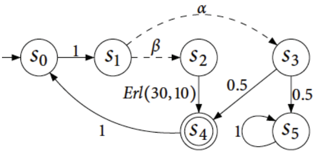
Remark

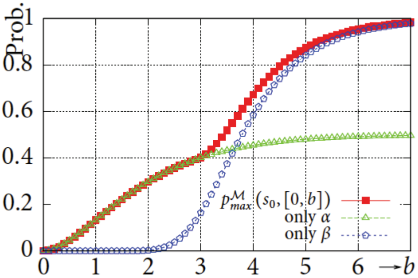
$LRA^{\min}(G)$ is the counterpart of stationary probabilities in CTMCs; timed reachability for $I = [t, t]$ corresponds to transient probabilities in CTMCs.

Policies [Miller 1969]

Positional policies are insufficient; (total) timed policies are optimal.

Why timed policies are essential





Discretisation: bounding the imprecision

Approximation theorem: [Zhang and Neuhäusser, 2010]

Let $I = [k_a{\cdot}\tau, k_b{\cdot}\tau]$ for some $k_a, k_b \in \mathbb{N}$. Then:

$$k_a{\cdot}\frac{(\lambda\tau)^2}{2} \;\leq\; |p^*(s, I) - \tilde{p}^*(\cdot)| \;\leq\; k_b{\cdot}\frac{(\lambda\tau)^2}{2} + \lambda\tau$$

where λ is the maximal rate in simple MA M,

Time complexity

Value $\tilde{p}^*(\cdot)$ can be obtained by value iteration for $\varepsilon > 0$ and $\sup I = b$ in

$$\mathcal{O}\Big(n^{2.376} + (m{+}n^2) \cdot \frac{(\lambda b)^2}{\varepsilon}\Big)$$

Bisimulation

Property preservation

For two bisimilar states s and t:

1. the minimal (and maximal) expected reachability times coincide
2. the long-run average times coincide
3. timed reachability probabilities coincide.

This holds for bisimulation and weak bisimulation.

Time complexity

The quotient under $\sim$ can be computed in $\mathcal{O}(m \log n)$.

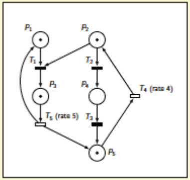
Deciding $\approx$ can be done in polynomial time.

Given that $\sim$ and $\approx$ are congruences wrt. parallel composition, this allows for compositional state-space reduction.

Overview

Tool support

GSPN
(.pnml)

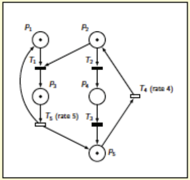


`reach P1 = 1 & P5 = 2`

http://wwwhome.cs.utwente.nl/~timmer/scoop/

Tool support

GSPN (.pnml)



reach P1 = 1 & P5 = 2

GEMMA

MAPA

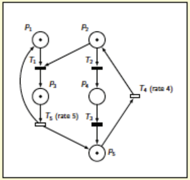
```
GSPN(P1:ℕ,P2:ℕ,P3:ℕ,
P4:ℕ,P5:ℕ) =

P2 >= 1 => T2 .
    GSPN[P2--, P4++]
+ P5 >= 1 => (4.0) .
    GSPN[P2++, P5--]
+ ...

init GSPN(1,1,1,0,1)
```

reach P1 = 1 & P5 = 2

http://wwwhome.cs.utwente.nl/~timmer/scoop/

Tool support

GSPN (.pnml)



`reach P1 = 1 & P5 = 2`

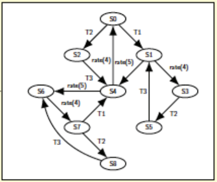
GEMMA

MAPA

```
GSPN(P1:ℕ,P2:ℕ,P3:ℕ,
P4:ℕ,P5:ℕ) =

P2 >= 1 => T2 .
    GSPN[P2--, P4++]
+ P5 >= 1 => (4.0) .
    GSPN[P2++, P5--]
+ ...

init GSPN(1,1,1,0,1)
```

`reach P1 = 1 & P5 = 2`

SCOOP



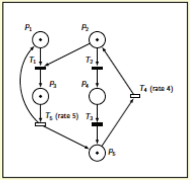
`#GOALS S4`

MA

http://wwwhome.cs.utwente.nl/~timmer/scoop/

Tool support

GSPN
(.pnml)




```
reach P1 = 1 & P5 = 2
```

GEMMA

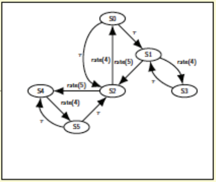
MAPA

```
GSPN(P1:ℕ,P2:ℕ,P3:ℕ,
P4:ℕ,P5:ℕ) =

P2 >= 1 => T2 .
    GSPN[P2--, P4++]
+ P5 >= 1 => (4.0) .
    GSPN[P2++, P5--]
+ ...

init GSPN(1,1,1,0,1)
```

```
reach P1 = 1 & P5 = 2
```

SCOOP
(optimised)

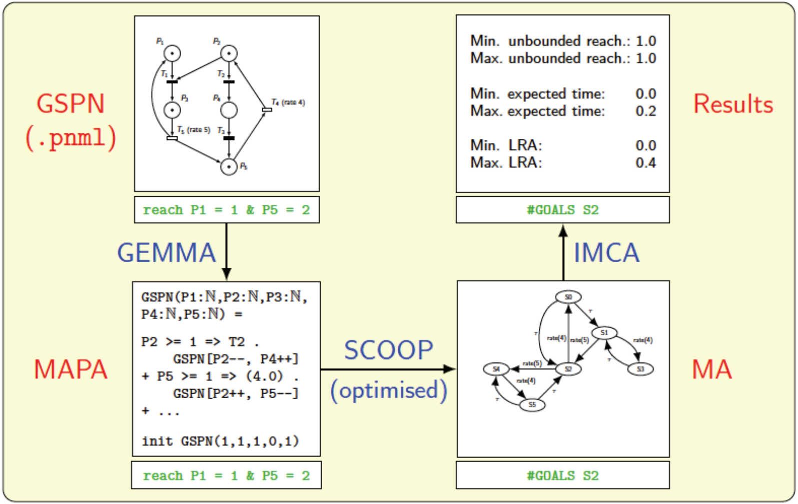
MA




```
#GOALS S2
```

http://wwwhome.cs.utwente.nl/~timmer/scoop/

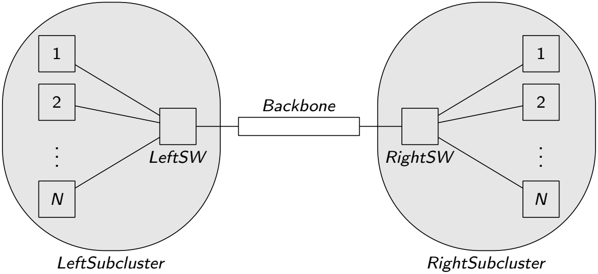
Tool support



http://wwwhome.cs.utwente.nl/~timmer/scoop/

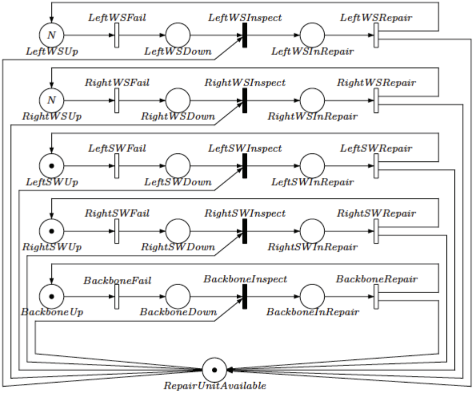
Workstation cluster

[Haverkort et al., SRDS 2000]



event	duration	event	duration
LeftWSFail	500h	*LeftWSRepair*	0.5h
RightWSFail	500h	*RightWSRepair*	0.5h
LeftSWFail	4000h	*LeftSWRepair*	4h
RightSWFail	4000h	*RightSWRepair*	4h
BackboneFail	5000h	*BackboneRepair*	8h

GSPN model



Due to the presence of confusion, this net is only analysable by classical GSPN analysis techniques when introducing a repair strategy.

Expected time analysis

N	k	$\|S\|$	$\|G\|$	$eT^{\max}(s_0, \Diamond G)$	$eT^{\min}(s_0, \Diamond G)$	time (seconds) max	time (seconds) min
4	3	819	566	1125179.46	1122465.40	0.014	0.023
4	4	819	692	51704.89	51699.58	0.002	0.002
4	6	819	807	1427.22	1427.22	0.007	0.011
4	8	819	818	59.88	59.88	0.002	0.002
8	6	2771	2009	1724343.30	1719447.05	0.119	0.267
8	8	2771	2482	25610.45	25604.95	0.039	0.044
8	12	2771	2736	1428.57	1428.57	0.017	0.016
8	16	2771	2770	30.58	30.58	0.008	0.007
16	12	10131	7544	1966511.37	1963868.17	1.243	3.505
16	16	10131	9374	12751.19	12745.39	0.241	0.311
16	24	10131	10014	1428.57	1428.57	0.071	0.076
16	32	10131	10130	15.46	15.46	0.053	0.052
32	24	38675	29210	1982468.90	1978880.69	20.716	114.111
32	32	38675	36406	6642.52	6636.17	2.141	3.256
32	48	38675	38250	1428.57	1428.57	0.791	0.909
32	64	38675	38674	7.77	7.77	0.643	0.671

Long-run analysis

N	k	$\|S\|$	$\|G\|$	$\mathrm{LRA}^{\max}(s_0, G)$	$\mathrm{LRA}^{\min}(s_0, G)$	time (seconds) max	time (seconds) min
4	3	819	253	0.999996	0.999996	0.323	0.284
4	4	819	127	0.999924	0.999923	0.239	0.299
4	6	819	12	0.996401	0.996401	0.321	0.319
4	8	819	1	0.988413	0.988413	0.321	0.269
8	6	2771	762	0.999998	0.999998	3.838	3.264
8	8	2771	289	0.999838	0.999836	2.607	3.418
8	12	2771	35	0.996399	0.996398	3.948	3.539
8	16	2771	1	0.980421	0.980421	3.360	3.227
16	12	10131	2587	0.999998	0.999998	86.148	49.547
16	16	10131	757	0.999655	0.999652	50.691	302.064
16	24	10131	117	0.996393	0.996392	47.571	260.782
16	32	10131	1	0.964439	0.964439	45.891	337.817
32	24	38675	9465	0.999998	0.999998	1913.229	24342.275
32	32	38675	2269	0.999306	0.999271	1345.557	7267.627
32	48	38675	425	0.996382	0.996378	1440.494	4964.068
32	64	38675	1	0.932476	0.932476	1104.782	13232.694

These results cover all possible weight assignments.

Timed reachability analysis

N	k	$\|S\|$	quot. states	t	results		time	
					MA	weights	MA	weights
4	3	820	424	20	0.3797	0.3038	304s	4s
4	5	820	164	20	0.4219	0.3717	90s	4s
4	8	820	164	20	0.4278	0.4250	15m	4s
8	3	2772	1412	10	0.9319	0.7457	277s	6s
8	10	2772	316	10	0.9805	0.9178	45s	7s
8	16	2772	316	20	0.6147	0.6089	36m	123s

Property

If QoS is too low, the maximal probability to violate QoS again within t time units.

System reliability is 18% less than predicted by standard GSPN analysis.

Explanation

The chance to go from a degraded to a degraded mode of operation is high
⇒ the worst case repair strategy becomes relevant

Overview

Concluding remarks

GSPNs := Markov Automata

- It is truly simple
- It is intuitive
- It is compositional
- It is backward compatible
- Allows compositional reduction
- No restrictions on net level

Analysis algorithms

- Expected time = SSP problem
- Long-run average
  1. Graph analysis
  2. Long-run ratio problem
  3. SSP problem
- Timed reachability
  - Discretisation

Confused GSPNs pose no problems. Neither semantically nor analysability.

Future work: efficiency gains, symbolic, abstraction, confluence reduction, applications (railway systems, aerospace systems, cloud computing) . . .

Thanks

Co-workers

Rob Bamberg (Twente), Dennis Guck (Aachen), Tingting Han (Oxford), Holger Hermanns (Saarbrücken), Martin Neuhäusser (Siemens), Mark Timmer (Twente) and Lijun Zhang (DTU Lyngby)

Literature

LICS'10, ICALP'11, NFM'12, CONCUR'12, ACSD'12, TACAS'13?

Tool download at:

`http://wwwhome.cs.utwente.nl/~timmer/scoop/`