Model Checking Nondeterministic and Randomly Timed Systems

Martin R. Neuhäußer[1,2]

[1] RWTH Aachen University, Germany
[2] University of Twente, The Netherlands

Oberseminar on January 25, 2010, RWTH Aachen University

The challenge

Software in safety critical systems becomes more and more complex.

The challenge

Software in safety critical systems becomes more and more complex.

Erroneous behavior of complex hardware & software systems

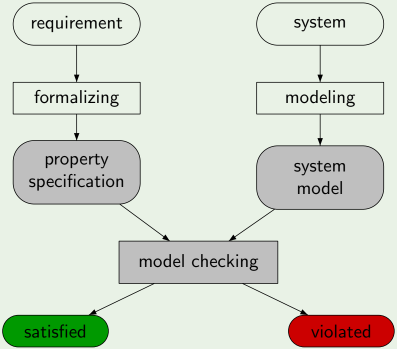
The challenge

Software in safety critical systems becomes more and more complex.

Erroneous behavior of complex hardware & software systems

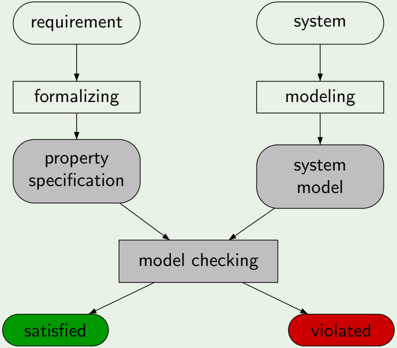
Formal methods in computer science

The model checking approach



Formal methods in computer science

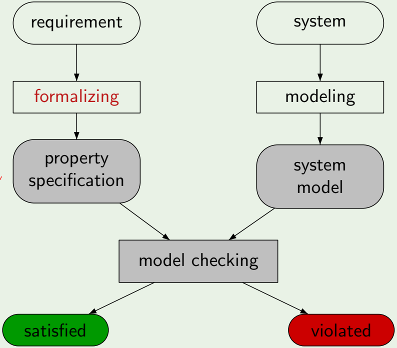
The model checking approach



Formal methods in computer science

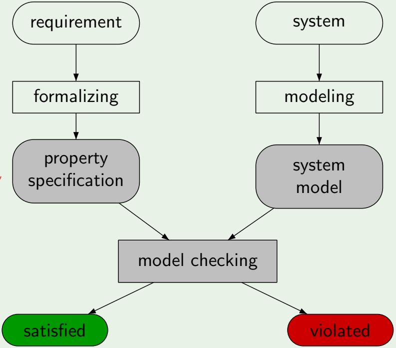
The model checking approach

$$\Phi = \forall \Box \neg \textit{collision}$$



Formal methods in computer science

The model checking approach



$\Phi = \forall \Box \neg \textit{collision}$

Formal methods in computer science

The model checking approach

$\Phi = \forall \Box \neg \textit{collision}$

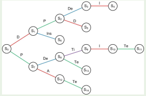
requirement → formalizing → property specification

system → modeling → system model

property specification, system model → model checking

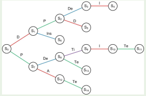
model checking → satisfied

model checking → violated



Formal methods in computer science

The model checking approach



requirement → formalizing → property specification

system → modeling → system model

$\Phi = \forall \square \neg \textit{collision}$

property specification, system model → model checking

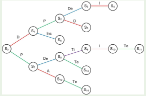
model checking → satisfied

model checking → violated

The model checking approach

requirement → formalizing → property specification

$\Phi = \forall \, \Box \, \neg \textit{collision}$



system → modeling → system model

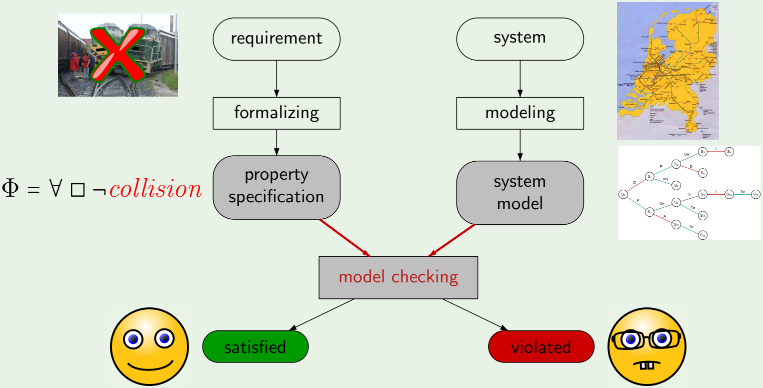
property specification, system model → **model checking**

model checking → satisfied

model checking → violated

Formal methods in computer science

The model checking approach



Model checking: Does a system model satisfy its specification?

Quantitative system analysis

Classical model checking

Model checking yields **YES** or **NO**.
But: Absolute correctness unrealistic:

- Systems are subject to random phenomena
- Environment behaves randomly
- Imprecisions in the model

Quantitative model checking

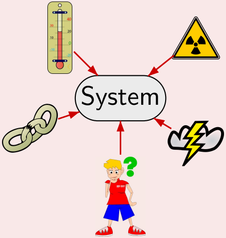
Extend models with probabilities to describe real-world systems
$\Rightarrow$ Performance & dependability evaluation

Quantitative system analysis

Classical model checking

Model checking yields **YES** or **NO**.
But: Absolute correctness unrealistic:

1. Systems are subject to random phenomena
2. Environment behaves randomly
3. Imprecisions in the model

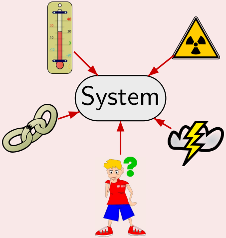


Quantitative system analysis

Classical model checking

Model checking yields **YES** or **NO**.
But: Absolute correctness unrealistic:

1. Systems are subject to random phenomena
2. Environment behaves randomly
3. Imprecisions in the model



Quantitative model checking

Extend models with probabilities to describe real-world systems
$\Rightarrow$ **Performance & dependability evaluation**

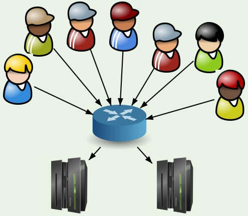
Outline of the talk

1. Introduction
2. Continuous-time Markov decision processes (CTMDPs)
   - Motivation
   - Preliminaries
   - Resolving nondeterministic choices
3. Time-bounded reachability analysis in CTMDPs
   - The approximation algorithm
   - Solving the sJSP
4. Further results in the thesis
   - Model checking interactive Markov chains
   - Model checking generalized stochastic Petri nets
5. Conclusion

Motivation: The stochastic job scheduling problem

Application: Load balancing of a bank's website

1. Customer request ≡ job.
2. Distribute request to multiple servers.
3. Classify jobs according to exp. duration:
   - Online Banking: long job
   - Serving ticker: short job



Clever way to distribute jobs to servers?

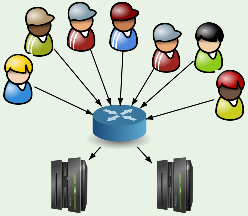
The problem statement [Bruno,Downey,Frederickson '81]

- Four jobs $\{1, 2, 3, 4\}$
- Expected duration of job k is $\frac{1}{\lambda_k}$ time units.
- Two identical processors.

Motivation: The stochastic job scheduling problem

Application: Load balancing of a bank's website

1. Customer request ≡ job.
2. Distribute request to multiple servers.
3. Classify jobs according to exp. duration:
   - Online Banking: long job
   - Serving ticker: short job



Clever way to distribute jobs to servers?

The problem statement [Bruno,Downey,Frederickson '81]

- Four jobs $\{1, 2, 3, 4\}$
- Expected duration of job k is $\frac{1}{\lambda_k}$ time units.
- Two identical processors.

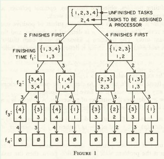


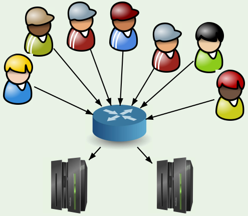
Figure 1

Motivation: The stochastic job scheduling problem

Application: Load balancing of a bank's website

1. Customer request ≡ job.
2. Distribute request to multiple servers.
3. Classify jobs according to exp. duration:
   - Online Banking: long job
   - Serving ticker: short job

Clever way to distribute jobs to servers?



The problem statement [Bruno,Downey,Frederickson '81]

- Four jobs $\{1, 2, 3, 4\}$
- Expected duration of job k is $\frac{1}{\lambda_k}$ time units.
- Two identical processors.
- **1981:** Minimize expected makespan.

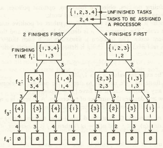
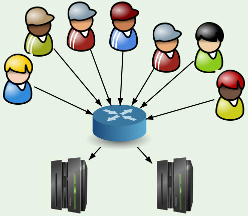


FIGURE 1

Motivation: The stochastic job scheduling problem

Application: Load balancing of a bank's website

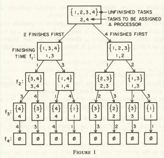
1. Customer request ≡ job.
2. Distribute request to multiple servers.
3. Classify jobs according to exp. duration:
    - Online Banking: long job
    - Serving ticker: short job



Clever way to distribute jobs to servers?

The problem statement [Bruno,Downey,Frederickson '81]

- Four jobs $\{1, 2, 3, 4\}$
- Expected duration of job k is $\frac{1}{\lambda_k}$ time units.
- Two identical processors.
- **1981:** Minimize expected makespan.



Today: Compute maximum probability to finish all jobs within time z!

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

$$\longrightarrow \boxed{1,2,3,4}$$

In this talk:

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

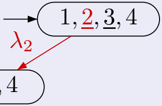
$$\longrightarrow \boxed{1, \underline{2}, \underline{3}, 4}$$

In this talk

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

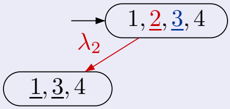


In this talk

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

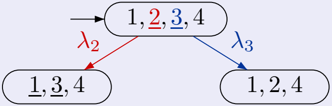


In this talk

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability.

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

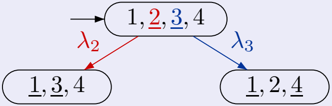


In this talk:

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

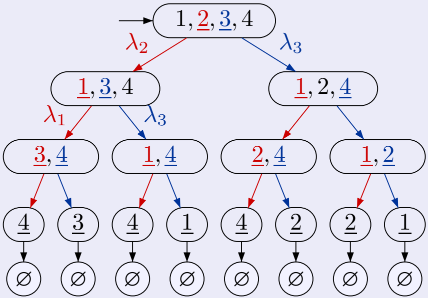


In this talk

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability.

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

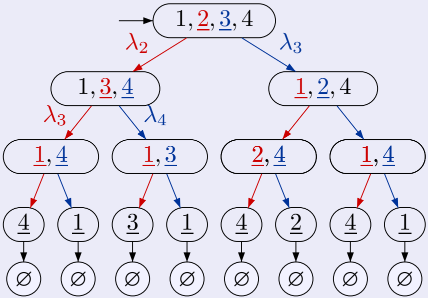


In this talk:

- Compute the maximum probability to finish all jobs before time …
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors



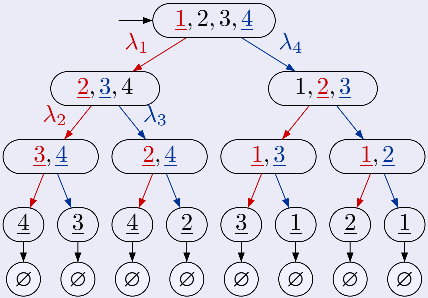
Many schedules possible!

In this talk

- Compute the maximum probability to finish all jobs before time ...
- Synthesize optimal schedule to achieve this probability

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors



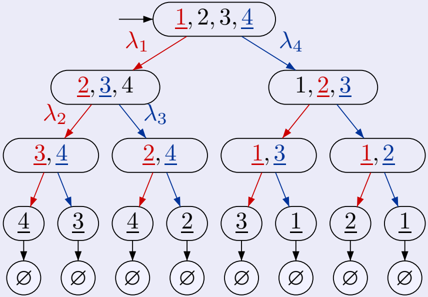
Many schedules possible!

In this talk:

- Compute the maximum probability to finish all jobs before time t.
- Synthesize optimal schedule to achieve this probability.

Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors



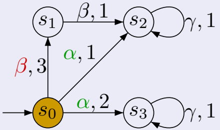
Many schedules possible!

In this talk:

1. Compute the maximum probability to finish all jobs before time z.
2. Synthesize optimal schedule to achieve this probability.

Continuous-time Markov decision processes (CTMDPs)

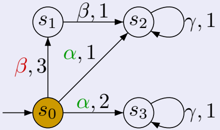
Random timing and nondeterminism in CTMDPs



Initial state: s_0
Actions: $Act(s_0) = \{\alpha, \beta\}$
Choice is nondeterministic!
Transition rates: $\mathbf{R}(s_0, \alpha, s_3) = 2$

Continuous-time Markov decision processes (CTMDPs)

Random timing and nondeterminism in CTMDPs



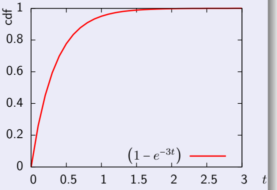
Initial state: s_0
Actions: $Act(s_0) = \{\alpha, \beta\}$
Choice is nondeterministic!
Transition rates: $\mathbf{R}(s_0, \alpha, s_3) = 2$

If action β is chosen: Only one transition available

$s_0 \xrightarrow{\beta, 3} s_1$ executes after $X_1 \sim Exp(3)$ time units.

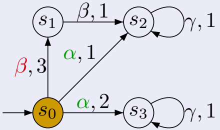
Probability to move before t time units:

$$P(X_1 \leq t) = \int_0^t 3 \cdot e^{-3x}\, dx = \left(1 - e^{-3t}\right)$$



Continuous-time Markov decision processes (CTMDPs)

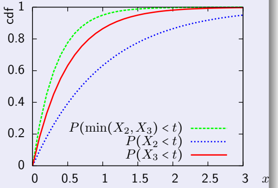
Random timing and nondeterminism in CTMDPs



Initial state: s_0
Actions: $Act(s_0) = \{\alpha, \beta\}$
Choice is nondeterministic!
Transition rates: $\mathbf{R}(s_0, \alpha, s_3) = 2$

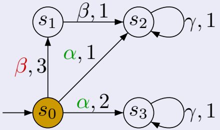
If action α is chosen: Race condition

$s_0 \xrightarrow{\alpha, 1} s_2$ executes after $X_2 \sim Exp(1)$ time units.
$s_0 \xrightarrow{\alpha, 2} s_3$ executes after $X_3 \sim Exp(2)$ time units.



Continuous-time Markov decision processes (CTMDPs)

Random timing and nondeterminism in CTMDPs



Initial state: s_0

Actions: $Act(s_0) = \{\alpha, \beta\}$

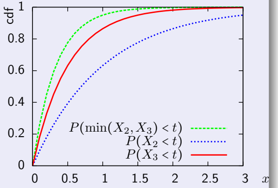
Choice is nondeterministic!

Transition rates: $\mathbf{R}(s_0, \alpha, s_3) = 2$

If action α is chosen: Race condition

$s_0 \xrightarrow{\alpha,1} s_2$ executes after $X_2 \sim Exp(1)$ time units.

$s_0 \xrightarrow{\alpha,2} s_3$ executes after $X_3 \sim Exp(2)$ time units.



The transition that executes first, wins:

1. Time spent in s_0: $\min(X_2, X_3) \sim Exp(1 + 2)$
   Exit rate: $E(s, \alpha) = \sum_{s' \in \mathcal{S}} \mathbf{R}(s, \alpha, s') = 1 + 2$
2. Prob. to move to $s_2 = P(X_2 < X_3)$.

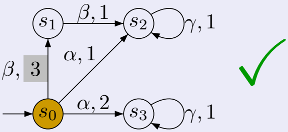
The subclass of locally uniform CTMDPs

Restriction to local uniformity

A CTMDP is locally uniform iff

$$\forall s \in \mathcal{S}.\ \forall \alpha, \beta \in Act(s). \quad E(s,\alpha) = E(s,\beta).$$

Exit rate $E(s)$ independent of action!



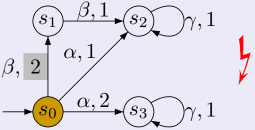
The subclass of locally uniform CTMDPs

Restriction to local uniformity

A CTMDP is locally uniform iff

$$\forall s \in \mathcal{S}.\ \forall \alpha, \beta \in Act(s). \quad E(s,\alpha) = E(s,\beta).$$

Exit rate $E(s)$ independent of action!



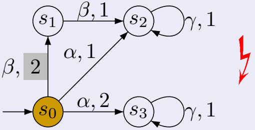
The subclass of locally uniform CTMDPs

Restriction to local uniformity

A CTMDP is locally uniform iff

$$\forall s \in \mathcal{S}.\ \forall \alpha, \beta \in Act(s). \quad E(s,\alpha) = E(s,\beta).$$

Exit rate $E(s)$ independent of action!



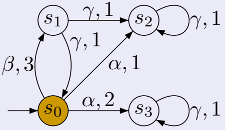
Why this restriction?

Sojourn time in a state does not depend on action!

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$\pi = s_0$



Time & history dependent schedulers

A scheduler is a measurable mapping

$$D : Paths^{\star} \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

The probability measure

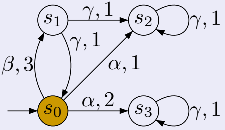
Scheduler D & initial state s induce unique probability measure

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$\pi = s_0$

wait in s_0 for $X_0 \sim Exp(3)$ time units

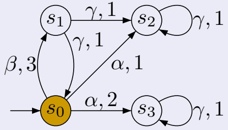


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{t_0} ?$$

upon leaving s_0: nondeterministic choice!

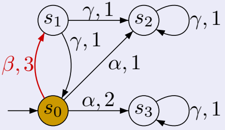


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} ?$$

$$D\left(s_0, t_0\right) = \beta$$



Time & history dependent schedulers

A scheduler is a measurable mapping

$$D : Paths^{*} \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

The probability measure

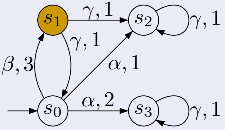
Scheduler D & initial state ν induce unique probability measure

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1$$

wait in s_1 for $X_1 \sim Exp(2)$ time units

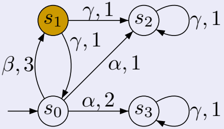


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{t_1} ?$$

upon leaving s_1: only γ available.

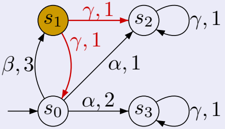


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} ?$$

Race: $\frac{\mathbf{R}(s_1,\gamma,s_2)}{E(s_1)} = \frac{1}{2}$ chance to move to s_0

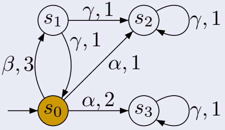


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0$$

wait in s_0 for $X_0 \sim Exp(3)$ time units



Time & history dependent schedulers

A scheduler is a measurable mapping

$$D : Paths^* \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

The probability measure

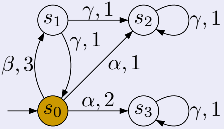
Scheduler D & initial state s induce unique probability measure $Pr^{\omega}_{D,s}$

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{t_2} ?$$

nondeterministic choice between α and β

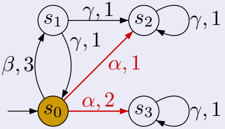


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{\alpha, t_2} ?$$

$$D\left(s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0, t_2\right) = \alpha$$

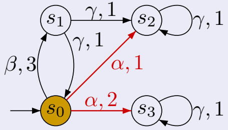


Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{\alpha, t_2} ?$$

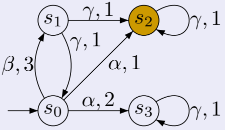
Race: $\frac{\mathbf{R}(s_0,\alpha,s_2)}{E(s_0)} = \frac{1}{3}$ chance to move to s_2



Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{\alpha, t_2} s_2$$



Time & history dependent schedulers

A scheduler is a measurable mapping

$$D : Paths^* \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

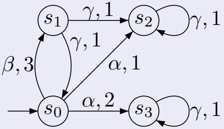
The probability measure

Scheduler D & initial state s induce unique probability measure Pr^{D}_{s}

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{\alpha, t_2} s_2$$



Time & history dependent schedulers [Neuhäußer,Stoelinga,Katoen '09]

A scheduler is a measurable mapping

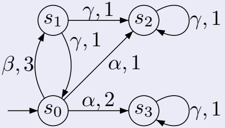
$$D : Paths^{\star} \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

$$\pi = s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \xrightarrow{\alpha, t_2} s_2$$



Time & history dependent schedulers [Neuhäußer,Stoelinga,Katoen '09]

A scheduler is a measurable mapping

$$D : Paths^{\star} \times \mathbb{R}_{\geq 0} \to Distr(Act)$$

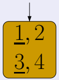
such that $D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(last(\pi))$.

The probability measure

Scheduler D & initial state s induce unique probability measure $Pr^{\omega}_{s,D}$!

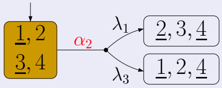
Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



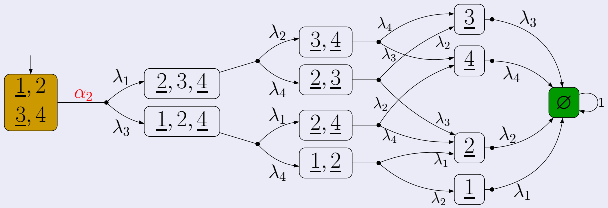
Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



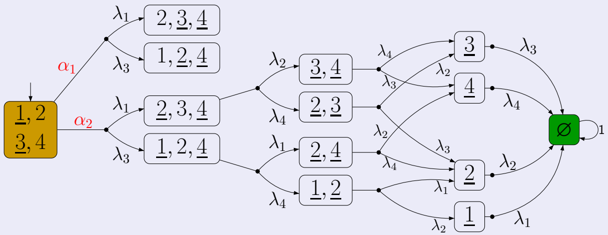
Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



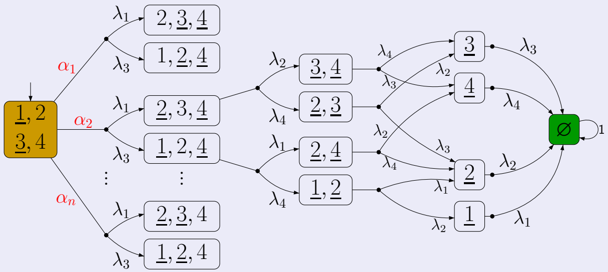
Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



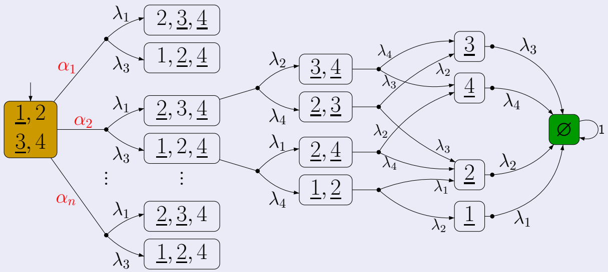
Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:

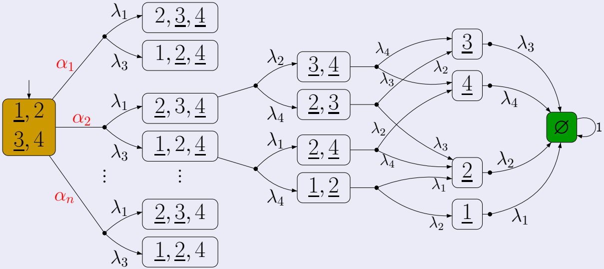


All scheduling strategies represented in the CTMDP.

$\alpha_1 : (1 \mapsto \{3,4\}, 3 \mapsto \{2,4\})$ $\alpha_2 : (1 \mapsto \{2,4\}, 3 \mapsto \{1,4\})$ ...

Modeling the sJSP as a CTMDP

Finishing all jobs within z time units:



All scheduling strategies represented in the CTMDP.

$\alpha_1 : (1 \mapsto \{3,4\}, 3 \mapsto \{2,4\})$ $\alpha_2 : (1 \mapsto \{2,4\}, 3 \mapsto \{1,4\})$ ...

Properties:

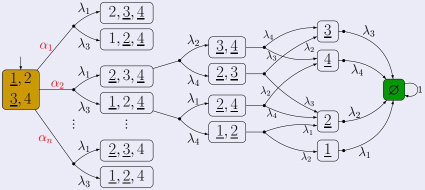
1. CTMDP combines nondeterministic choices and stochastic timing.
2. The CTMDP model is locally uniform.

Outline of the talk

Time-bounded reachability in the sJSP

Time-bounded reachability probabilities

- CTMDP model $\mathcal{C}$.
- Initial state: $s \in \mathcal{S}$
- Goal states: $G \subseteq \mathcal{S}$
- Time-bound: $z \in \mathbb{R}_{\geq 0}$



The time-bounded reachability event:

$$\Diamond^{[0,z]} G = \left\{ \pi \in Paths^{\omega} \mid \exists t \in [0,z].\ \pi@t \in G \right\}$$

Maximum time-bounded reachability probability

$$p^{\mathcal{C}}_{max}(s,z) = \sup_{D} Pr^{\omega}_{s,D}\left(\Diamond^{[0,z]} G\right)$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in S} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in S} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Computing the maximum time-bounded reachability probability

How to compute $p^{\mathcal{C}}_{max}$?

Idea: Characterize $p^{\mathcal{C}}_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

Define $\Omega : (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1]) \to (\mathcal{S} \times \mathbb{R}_{\geq 0} \to [0,1])$ on measurable functions:

- If $s \in G$ then $\Omega(F)(s,z) = 1$.
- If $s \notin G$ then

$$\Omega(F)(s,z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Fixed point characterization

The function $p^{\mathcal{C}}_{max}(s,z)$ is the least fixed point of Ω.

Applying the fixed point characterization directly

Fixed point characterization

If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Applying the fixed point characterization directly

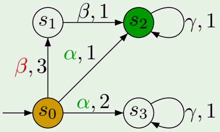
Fixed point characterization

If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$



Applying the fixed point characterization directly

Fixed point characterization

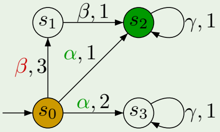
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_1(s_0,z) = \Omega(F_0)(s_0,z) = 0.$
$F_1(s_1,z) = \Omega(F_0)(s_1,z) = 0.$
$F_1(s_2,z) = \Omega(F_0)(s_2,z) = 1.$
$F_1(s_3,z) = \Omega(F_0)(s_0,z) = 0.$



Applying the fixed point characterization directly

Fixed point characterization

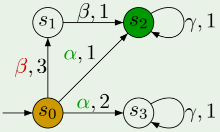
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(, \right)\ dt.$



Applying the fixed point characterization directly

Fixed point characterization

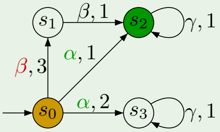
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\tfrac{1}{3} \cdot F_1(s_2, z-t),\right)\, dt.$$



Applying the fixed point characterization directly

Fixed point characterization

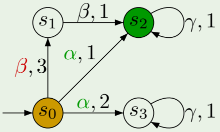
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, \right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

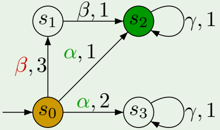
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, F_1(s_1,z-t)\right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

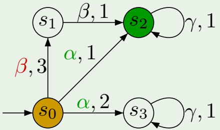
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 0\right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

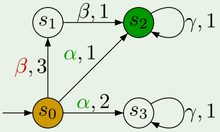
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \frac{1}{3}\left(1 - e^{-3z}\right).$



Applying the fixed point characterization directly

Fixed point characterization

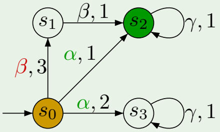
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \frac{1}{3}\left(1 - e^{-3z}\right).$
$F_2(s_1,z) = \Omega(F_1)(s_1,z) = \int_0^z e^{-t} \cdot F_1(s_2, z-t)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

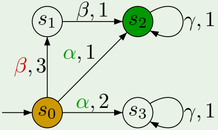
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \frac{1}{3}\left(1 - e^{-3z}\right).$
$F_2(s_1,z) = \Omega(F_1)(s_1,z) = \int_0^z e^{-t} \cdot F_1(s_2, z-t)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

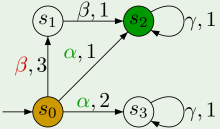
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \frac{1}{3}\left(1 - e^{-3z}\right).$
$F_2(s_1,z) = \Omega(F_1)(s_1,z) = 1 - e^{-z}.$



Applying the fixed point characterization directly

Fixed point characterization

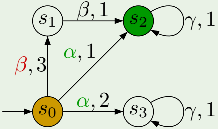
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_2(s_0,z) = \Omega(F_1)(s_0,z) = \frac{1}{3}\left(1 - e^{-3z}\right).$
$F_2(s_1,z) = \Omega(F_1)(s_1,z) = 1 - e^{-z}.$
$F_2(s_2,z) = \Omega(F_1)(s_2,z) = 1.$
$F_2(s_3,z) = \Omega(F_1)(s_3,z) = 0.$



Applying the fixed point characterization directly

Fixed point characterization

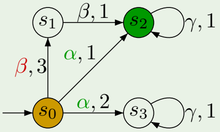
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max(,)\ dt.$



Applying the fixed point characterization directly

Fixed point characterization

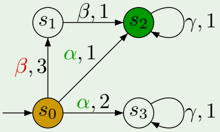
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3} \cdot F_2(s_2, z-t),\right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

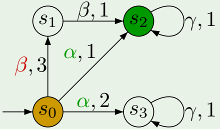
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3},\right)\ dt.$



Applying the fixed point characterization directly

Fixed point characterization

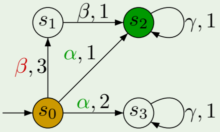
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, F_2(s_1, z-t)\right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

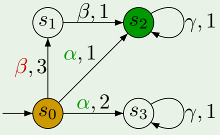
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

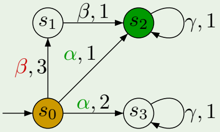
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = \int_0^z e^{-t} \cdot F_2(s_2, z-t)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

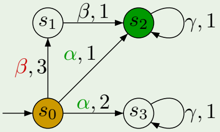
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = \int_0^z e^{-t} \cdot F_2(s_2, z-t)\, dt.$



Applying the fixed point characterization directly

Fixed point characterization

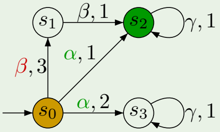
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = 1 - e^{-z}.$



Applying the fixed point characterization directly

Fixed point characterization

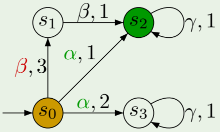
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = 1 - e^{-z}.$
$F_3(s_2,z) = \Omega(F_2)(s_2,z) = 1.$
$F_3(s_3,z) = \Omega(F_2)(s_3,z) = 0.$



Applying the fixed point characterization directly

Fixed point characterization

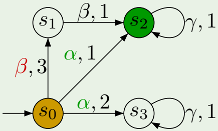
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = 1 - e^{-z}.$
$F_3(s_2,z) = \Omega(F_2)(s_2,z) = 1.$
$F_3(s_3,z) = \Omega(F_2)(s_3,z) = 0.$



Result: $F_3 = \Omega(F_3) \Rightarrow F_3$ is least fixed-point.

Applying the fixed point characterization directly

Fixed point characterization

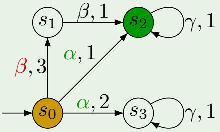
If $s \in G$: $\Omega(F)(s,z) = 1$. Otherwise:

$$\Omega(F)(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot F(s', z-t)\, dt.$$

Solving the reachability problem analytically

Fixed-point computation: $\forall s \in \mathcal{S}.\ F_0(s,z) = 0.$

$F_3(s_0,z) = \Omega(F_2)(s_0,z) = \int_0^z 3e^{-3t} \cdot \max\left(\frac{1}{3}, 1 - e^{-z}\right)\, dt.$
$F_3(s_1,z) = \Omega(F_2)(s_1,z) = 1 - e^{-z}.$
$F_3(s_2,z) = \Omega(F_2)(s_2,z) = 1.$
$F_3(s_3,z) = \Omega(F_2)(s_3,z) = 0.$



Result: $F_3 = \Omega(F_3) \Rightarrow F_3$ is least fixed-point.
For $z = 1$: $p^{\mathcal{C}}_{max}(s_0, 1) = 1 + \frac{19}{24}e^{-3} - \frac{3}{2}e^{-1} \simeq 0.48759.$

What is achieved so far:

Analytical solution

Allows to compute $p^{\mathcal{C}}_{max}(s, z)$ for small problem instances.

What is achieved so far:

Analytical solution

Allows to compute $p_{max}^{\mathcal{C}}(s, z)$ for small problem instances.

Disadvantages:

1. Numerical instabilities due to nested integrals.
2. Integration over the maximum of functions.

$\Rightarrow$ Fixed-point characterization not suitable for an algorithmic solution.

What is achieved so far:

Analytical solution

Allows to compute $p_{max}^{\mathcal{C}}(s, z)$ for small problem instances.

Disadvantages:

1. Numerical instabilities due to nested integrals.
2. Integration over the maximum of functions.

$\Rightarrow$ Fixed-point characterization not suitable for an algorithmic solution.

Instead:

Use the discretization technique that comes next!

A discretization that computes $p_{max}(s, z)$

Reduce $p^{\mathcal{C}}_{max}$ to step-bounded reachability $p^{\mathcal{C}_\tau}_{max}$ in MDPs.

Each discrete step corresponds to a time-interval of length τ.

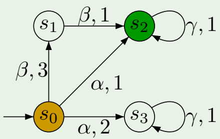
A discretization that computes $p_{max}(s,z)$

Reduce $p^{\mathcal{C}}_{max}$ to step-bounded reachability $p^{\mathcal{C}_\tau}_{max}$ in MDPs.

Each discrete step corresponds to a time-interval of length τ.

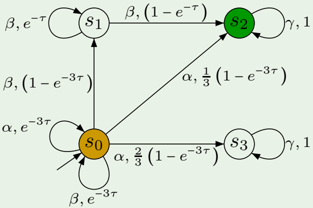
Continuous-time vs. discrete-time Markov decision processes

Continuous-time MDP $\mathcal{C}$



Exponential distributions

Discrete-time MDP $\mathcal{C}_\tau$



Discrete probability distributions

Reachability within time z $\equiv$ **Reachability in $\frac{z}{\tau}$ steps!**

$p^{\mathcal{C}}_{max}(s, z)$ $\qquad$ $p^{\mathcal{C}_\tau}_{max}(s, \frac{z}{\tau})$

Discretization I

Recall the fixed-point characterization:

The function $p^{\mathcal{C}}_{max}(s, z)$ is the least fixed point of Ω: If $s \notin G$, then

$$p^{\mathcal{C}}_{max}(s, z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') \cdot p^{\mathcal{C}}_{max}(s', z - t) \, dt.$$

Discretization I

Recall the fixed-point characterization:

The function $p^{\mathcal{C}}_{max}(s, z)$ is the least fixed point of Ω: If $s \notin G$, then

$$p^{\mathcal{C}}_{max}(s, z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') \cdot p^{\mathcal{C}}_{max}(s', z - t)\, dt.$$

The idea for a discretization to compute $p^{\mathcal{C}}_{max}(s, z)$:

Choose $\tau \ll z$ and split $p^{\mathcal{C}}_{max}(s, z)$ accordingly:

Discretization I

Recall the fixed-point characterization:

The function $p^{\mathcal{C}}_{max}(s, z)$ is the least fixed point of Ω: If $s \notin G$, then

$$p^{\mathcal{C}}_{max}(s, z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') \cdot p^{\mathcal{C}}_{max}(s', z - t)\, dt.$$

The idea for a discretization to compute $p^{\mathcal{C}}_{max}(s, z)$:

Choose $\tau \ll z$ and split $p^{\mathcal{C}}_{max}(s, z)$ accordingly:

$$A(s, z) = \int_0^\tau E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s, \alpha, s') \cdot p^{\mathcal{C}}_{max}(s', z - t)\, dt$$

Discretization I

Recall the fixed-point characterization:

The function $p^{\mathcal{C}}_{max}(s,z)$ is the least fixed point of Ω: If $s \notin G$, then

$$p^{\mathcal{C}}_{max}(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt.$$

The idea for a discretization to compute $p^{\mathcal{C}}_{max}(s,z)$:

Choose $\tau \ll z$ and split $p^{\mathcal{C}}_{max}(s,z)$ accordingly:

$$\begin{aligned}
A(s,z) &= \int_0^\tau E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt \\
B(s,z) &= \int_\tau^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt \\
&= e^{-E(s)\tau} \cdot p^{\mathcal{C}}_{max}(s, z-\tau).
\end{aligned}$$

Discretization I

Recall the fixed-point characterization:

The function $p^{\mathcal{C}}_{max}(s,z)$ is the least fixed point of Ω: If $s \notin G$, then

$$p^{\mathcal{C}}_{max}(s,z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt.$$

The idea for a discretization to compute $p^{\mathcal{C}}_{max}(s,z)$:

Choose $\tau \ll z$ and split $p^{\mathcal{C}}_{max}(s,z)$ accordingly:

$$A(s,z) = \int_0^\tau E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt$$

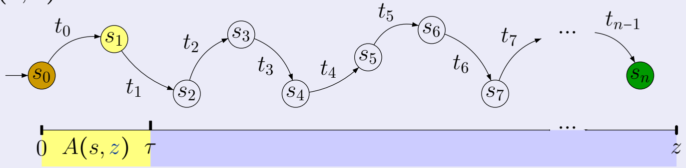
$$\begin{aligned} B(s,z) &= \int_\tau^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot p^{\mathcal{C}}_{max}(s', z-t)\, dt \\ &= e^{-E(s)\tau} \cdot p^{\mathcal{C}}_{max}(s, z-\tau). \end{aligned}$$

Relation to $p^{\mathcal{C}}_{max}(s,z)$: $p^{\mathcal{C}}_{max}(s,z) = A(s,z) + B(s,z)$

Discretization II

Intuition behind $A(s,z)$ and $B(s,z)$

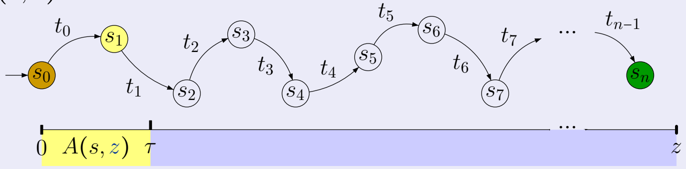
- $A(s,z)$ = Prob. to reach G within time z with ≥ 1 **transitions in** $[0,\tau]$.



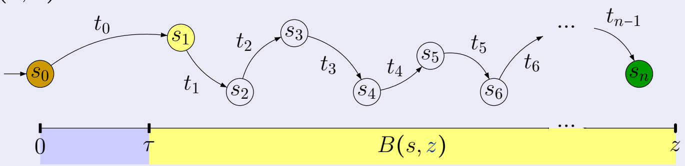
Discretization II

Intuition behind $A(s, z)$ and $B(s, z)$

- $A(s, z)$ = Prob. to reach G within time z with ≥ 1 **transitions in** $[0, \tau]$.



- $B(s, z)$ = Prob. to reach G within time z with **no transition in** $[0, \tau]$.



A step-wise approximation of $p^{\mathcal{C}}_{max}(s, z)$

A single step in the discretized MDP

CTMDP $\mathcal{C}$ and step duration $\tau < z$ induce the discretized MDP $\mathcal{C}_\tau$:

$$\mathbf{P}_\tau(s, \alpha, s') = \begin{cases} \left(1 - e^{-E(s)\tau}\right) \cdot \mathbf{P}(s, \alpha, s') & \text{if } s \neq s' \\ \left(1 - e^{-E(s)\tau}\right) \cdot \mathbf{P}(s, \alpha, s) + e^{-\lambda(s)\tau} & \text{if } s = s'. \end{cases}$$

A step-wise approximation of $p^{\mathcal{C}}_{max}(s,z)$

A single step in the discretized MDP

CTMDP $\mathcal{C}$ and step duration $\tau < z$ induce the discretized MDP $\mathcal{C}_\tau$:

$$\mathbf{P}_\tau(s,\alpha,s') = \begin{cases} \left(1 - e^{-E(s)\tau}\right) \cdot \mathbf{P}(s,\alpha,s') & \text{if } s \neq s' \\ \left(1 - e^{-E(s)\tau}\right) \cdot \mathbf{P}(s,\alpha,s) + e^{-\lambda(s)\tau} & \text{if } s = s'. \end{cases}$$

Theorem (Correctness of our reduction)

Let $\mathcal{C}$ be a CTMDP, G a set of goal states and z a time bound. Choose some $k \in \mathbb{N}_{>0}$ and set $\tau = \frac{z}{k}$. Then

$$p^{\mathcal{C}_\tau}_{max}(s,k) \leq p^{\mathcal{C}}_{max}(s,z) \leq p^{\mathcal{C}_\tau}_{max}(s,k) + \frac{(\lambda z)^2}{2k}.$$

1. *$p^{\mathcal{C}_\tau}_{max}(s,k)$ is the probability to reach G in at most k steps in $\mathcal{C}_\tau$,*
2. *$\lambda = max_{s\in\mathcal{S}} E(s)$ is the maximum exit rate in $\mathcal{C}$ and*
3. *k is the number of discretization steps.*

Solving the step-bounded reachability problem in $\mathcal{C}_\tau$

Value iteration for discrete-time MDPs [Bellman '57]

Let $G \subseteq \mathcal{S}$ be a set of goal states and $\vec{v}_n \in [0,1]^{|\mathcal{S}|}$ such that

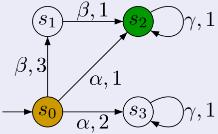
$$\vec{v}_0(s) = \begin{cases} 1 & \text{if } s \in G \\ 0 & \text{if } s \notin G \end{cases} \qquad \vec{v}_{n+1}(s) = \begin{cases} 1 & \text{if } s \in G \\ \max\limits_{\alpha \in Act} \sum\limits_{s' \in \mathcal{S}} \mathbf{P}(s,\alpha,s') \cdot v_n(s') & \text{if } s \notin G \end{cases}$$

Then $p_{max}^{\mathcal{C}_\tau}(s,k) = \vec{v}_k(s)$.

Summarizing our time-bounded reachability analysis

Input

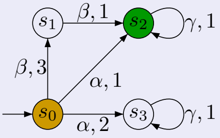
1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$



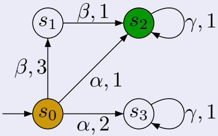
Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

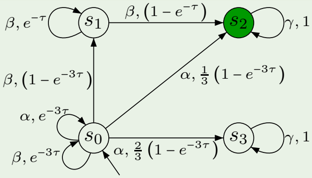


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

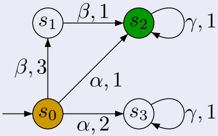
$\vec{v}_0 = (0, 0, 1, 0)$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

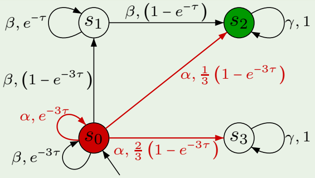


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p^{\mathcal{C}_\tau}_{max}(s, k)$

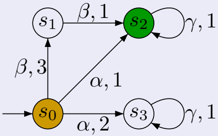
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \big(\max\big(\qquad\qquad , \quad \big), \qquad\qquad , \quad , \quad \big)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

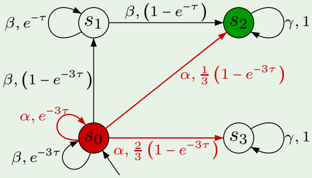


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

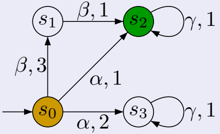
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\max\left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \quad\right), \qquad\qquad , \quad , \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

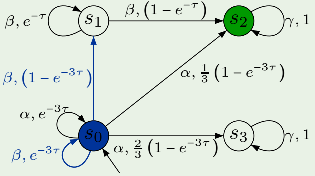


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

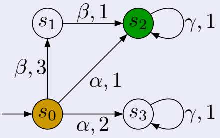
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\max\left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \quad \right), \qquad\qquad , \quad , \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

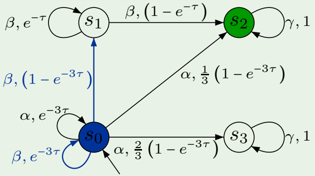


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s,k)$

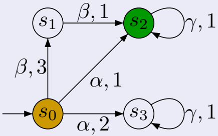
$$\vec{v}_0 = (0,0,1,0)$$
$$\vec{v}_1 = \left(\max\left(\tfrac{1}{3}\left(1-e^{-3\tau}\right), 0\right), \quad , \quad , \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$



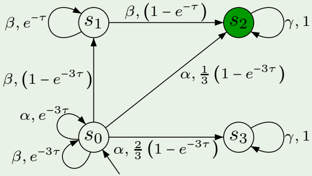
Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \le \varepsilon \quad \Rightarrow \quad k \ge 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s,k)$

$\vec{v}_0 = (0, 0, 1, 0)$

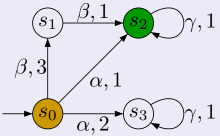
$\vec{v}_1 = \left(\frac{1}{3}\left(1 - e^{-3\tau}\right), \quad , \quad , \quad \right)$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

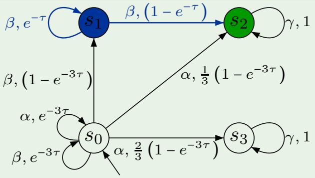


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p^{\mathcal{C}_\tau}_{max}(s,k)$

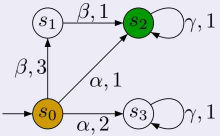
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \qquad , \quad , \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

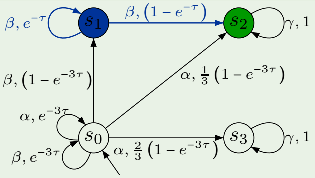


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p^{\mathcal{C}_\tau}_{max}(s,k)$

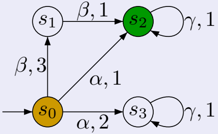
$$\vec{v}_0 = (0,0,1,0)$$
$$\vec{v}_1 = \left(\tfrac{1}{3}\left(1-e^{-3\tau}\right), \left(1-e^{-\tau}\right), \quad , \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

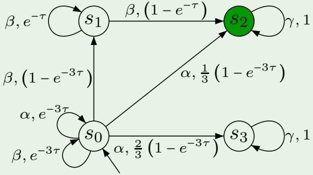


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p^{\mathcal{C}_\tau}_{max}(s, k)$

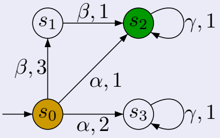
$\vec{v}_0 = (0, 0, 1, 0)$
$\vec{v}_1 = \left(\frac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), \quad , \quad \right)$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

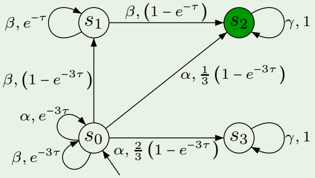


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

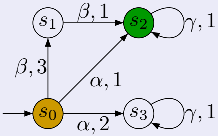
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), 1, \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

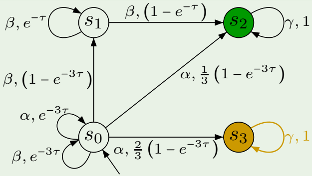


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

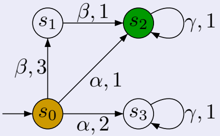
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), 1, \quad \right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

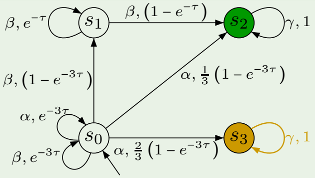


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \le \varepsilon \quad \Rightarrow \quad k \ge 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s,k)$

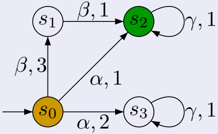
$$\vec{v}_0 = (0,0,1,0)$$
$$\vec{v}_1 = \left(\tfrac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), 1, 0\right)$$



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$

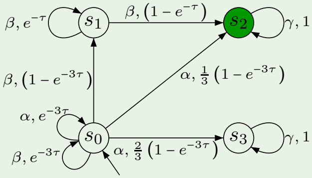


Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \le \varepsilon \quad \Rightarrow \quad k \ge 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

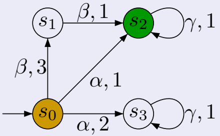
$$\vec{v}_0 = (0, 0, 1, 0)$$
$$\vec{v}_1 = \left(\frac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), 1, 0\right)$$
...



Summarizing our time-bounded reachability analysis

Input

1. locally uniform CTMDP
2. Goal states: $G = \{s_2\}$
3. Time bound: $z = 1$
4. Maximum allowed error: $\varepsilon = 10^{-3}$



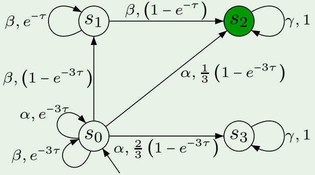
Compute the number of discretization steps k

$$\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.000\overline{2}.$$

Value iteration to compute $p_{max}^{\mathcal{C}_\tau}(s, k)$

$\vec{v}_0 = (0, 0, 1, 0)$
$\vec{v}_1 = \left(\frac{1}{3}\left(1 - e^{-3\tau}\right), \left(1 - e^{-\tau}\right), 1, 0\right)$
...

Result: $p_{max}^{\mathcal{C}_\tau}(s_0, 4500) = \vec{v}_{4500}(s_0) \simeq 0.487$



Complexity of the discretization approach

Complexity

For CTMDP $\mathcal{C}$, time bound z and error bound ε:

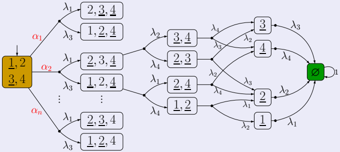
- Number of iteration steps: $\mathcal{O}\big((\lambda z)^2/\varepsilon\big)$.
- Each value iteration step: linear in the size of $\mathcal{C}$ (transitions + states)

Overall complexity: $\mathcal{O}\big(|\mathcal{C}| \cdot (\lambda z)^2/\varepsilon\big)$.

Outlook: Computing optimal solutions for the sJSP

Analysis of the sJSP

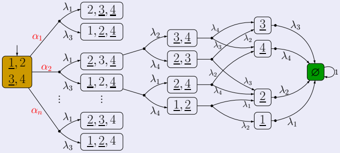
- Different rates ⇒ schedule important
- Synthesis of best and worst schedules



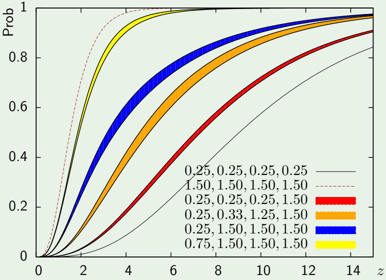
Outlook: Computing optimal solutions for the sJSP

Analysis of the sJSP

- Different rates ⇒ schedule important
- Synthesis of best and worst schedules



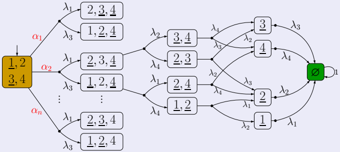
Numerical results: Maximum and minimum probabilities



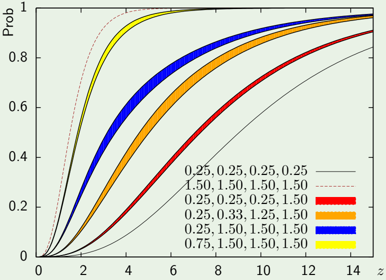
Outlook: Computing optimal solutions for the sJSP

Analysis of the sJSP

1. Different rates ⇒ schedule important.
2. Synthesis of best and worst schedules



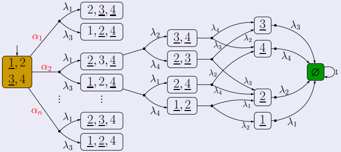
Numerical results: Maximum and minimum probabilities



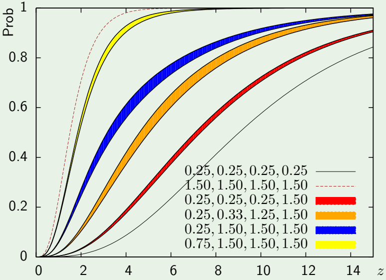
Outlook: Computing optimal solutions for the sJSP

Analysis of the sJSP

1. Different rates ⇒ schedule important.
2. Synthesis of best and worst schedules.



Numerical results: Maximum and minimum probabilities



Optimal schedule for $(0.25, 0.33, 1.25, 1.5)$

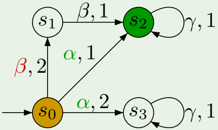
$\underline{1}, \underline{2}, 3, 4$	$1 \mapsto \{2,3\}$ $2 \mapsto \{1,3\}$
$1, \underline{2}, \underline{3}, 4$	$2 \mapsto \{1,3\}$ $3 \mapsto \{1,2\}$
$1, 2, \underline{3}, \underline{4}$	$3 \mapsto \{1,2\}$ $4 \mapsto \{1,2\}$

Outline of the talk

Interactive Markov chains

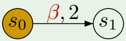
Model Checking Interactive Markov Chains [Zhang, Neuhäußer '10]

Continuous-time MDP



Combines actions and rates.

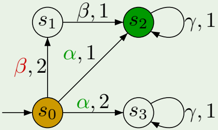
Only one type of transitions:



Interactive Markov chains

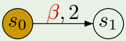
Model Checking Interactive Markov Chains [Zhang, Neuhäußer '10]

Continuous-time MDP

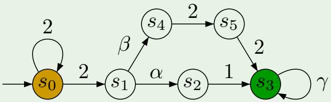


Combines actions and rates.

Only one type of transitions:

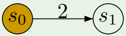


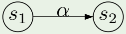
Interactive Markov Chain [Hermanns'02]



Separates actions and rates.

Two types of transitions:

- Markovian 

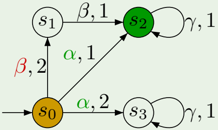
- Interactive 

Maximal progress assumption!

Interactive Markov chains

Model Checking Interactive Markov Chains [Zhang, Neuhäußer '10]

Continuous-time MDP

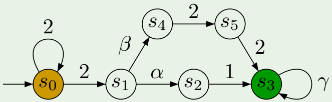


Combines actions and rates.

Only one type of transitions:

$$s_0 \xrightarrow{\beta, 2} s_1$$

Interactive Markov Chain [Hermanns'02]



Separates actions and rates.

Two types of transitions:

- Markovian $s_0 \xrightarrow{2} s_1$
- Interactive $s_1 \xrightarrow{\alpha} s_2$

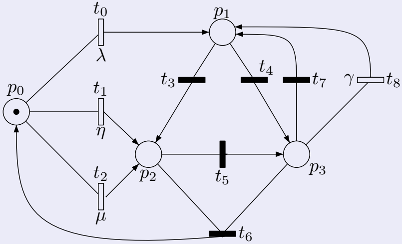
Maximal progress assumption!

Extending the discretization from CTMDPs to IMCs

Model checking the continuous stochastic logic (CSL) on IMCs!

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

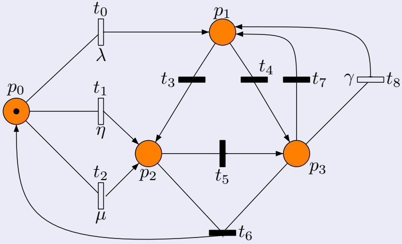


- Input, output and inhibitor arcs
- Tokens

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

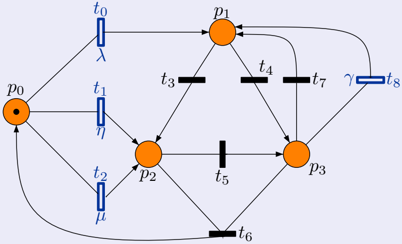
- Places P
- Input, output and inhibitor arcs
- Tokens



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

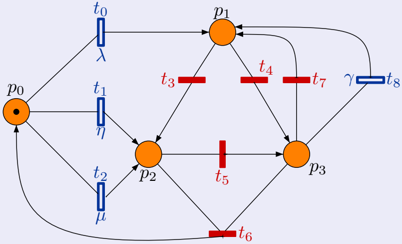
- Places P
- Timed transitions (exp. rate)
- Input, output and inhibitor arcs
- Tokens



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

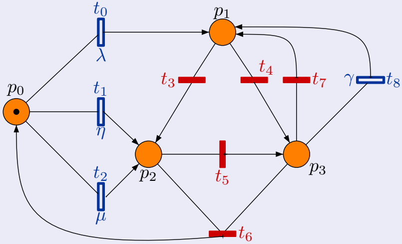
- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

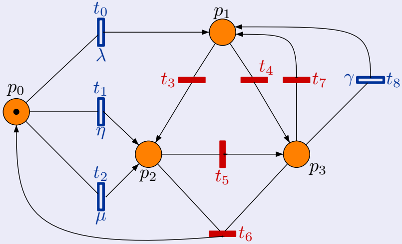
- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

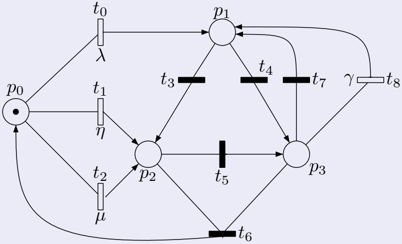


Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

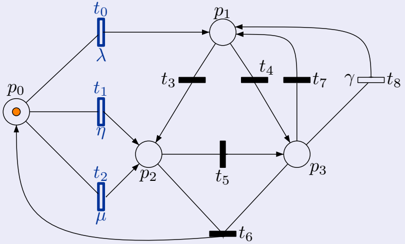
Semantics: Reachability graph



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens



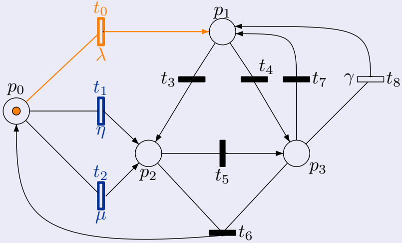
Semantics: Reachability graph
$\Rightarrow$ token game!

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

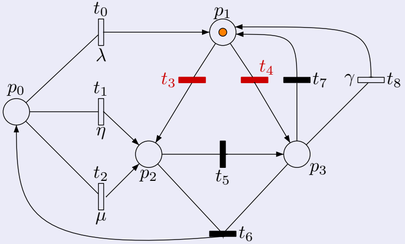
Semantics: Reachability graph
⇒ token game!



Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens



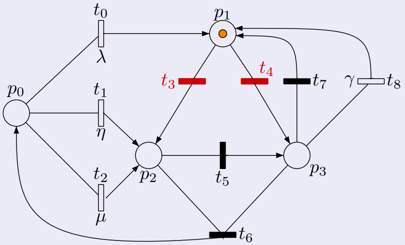
Semantics: Reachability graph
$\Rightarrow$ token game!

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

Semantics: Reachability graph
⇒ token game!



Overcome confusion in GSPNs [Hermanns,Katoen,Neuhäußer,Zhang '10]

Multiple conflicting immediate transitions enabled.
Which one executes first?

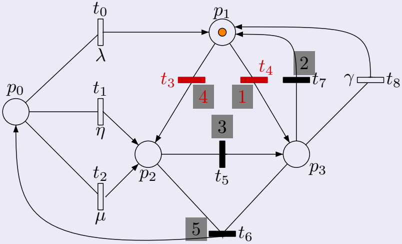
- Classical answer: Avoid this case by using weights!
- New approach: Nondeterminism!
  Interpret reachability graph of a GSPN as an IMC!

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

Semantics: Reachability graph
$\Rightarrow$ token game!



Overcome confusion in GSPNs [Hermanns,Katoen,Neuhäußer,Zhang '10]

Multiple conflicting immediate transitions enabled.
Which one executes first?

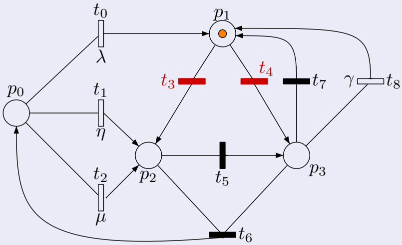
1. Classical answer: Avoid this case by using weights!
2. New approach: Nondeterminism!
   Interpret reachability graph of a GSPN as an IMC!

Model checking generalized stochastic Petri nets

Generalized stochastic Petri nets (GSPNs) [Marsan,Conte,Balbo '84]

- Places P
- Timed transitions (exp. rate)
- Immediate transitions
- Input, output and inhibitor arcs
- Tokens

Semantics: Reachability graph
$\Rightarrow$ token game!



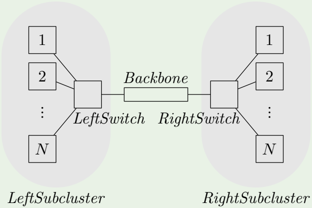
Overcome confusion in GSPNs [Hermanns,Katoen,Neuhäußer,Zhang '10]

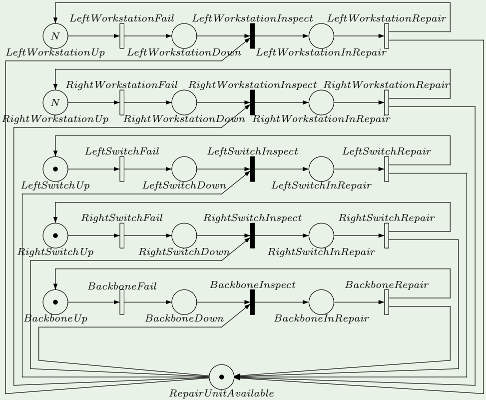
Multiple conflicting immediate transitions enabled.
Which one executes first?

1. Classical answer: Avoid this case by using weights!
2. New approach: Nondeterminism!
   Interpret reachability graph of a GSPN as an IMC!

Dependability Analysis of a Workstation Cluster Haverkort'00

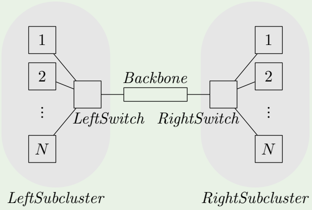
A GSPN model for the dependable workstation cluster

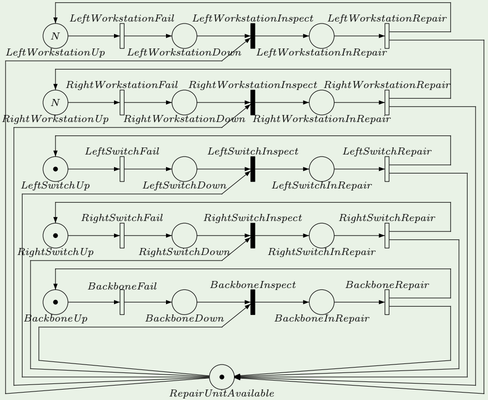




Dependability Analysis of a Workstation Cluster Haverkort'00

A GSPN model for the dependable workstation cluster





Result of the nondeterministic analysis

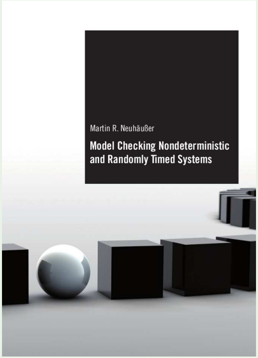
System is 18% less reliable than predicted by earlier analysis!

Outline of the talk

Results of the thesis

What can be found in there?

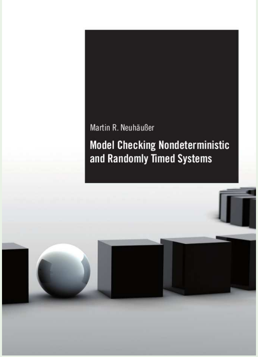
1. Continuous-time Markov decision processes
   - A new class of time-dependent schedulers
   - Time-bounded reachability analysis
   - Strong bisimulation minimization for CTMDPs.
2. Interactive Markov chains
   - Extension of CTMDP analysis to IMCs
   - CSL model checking algorithm
3. Generalized stochastic Petri nets
   - A simple and concise semantics for GSPNs
   - Model checking CSL formulas on GSPNs
4. Case studies
   - Solving the stochastic job scheduling problem
   - Dependability analysis of a workstation cluster



Results of the thesis

What can be found in there?

1. Continuous-time Markov decision processes
   - A new class of time-dependent schedulers
   - **Time-bounded reachability analysis**
   - Strong bisimulation minimization for CTMDPs.
2. Interactive Markov chains
   - Extension of CTMDP analysis to IMCs
   - **CSL model checking algorithm**
3. Generalized stochastic Petri nets
   - **A simple and concise semantics for GSPNs**
   - Model checking CSL formulas on GSPNs
4. Case studies
   - Solving the stochastic job scheduling problem
   - Dependability analysis of a workstation cluster



Open problems and future research directions

The future...

1. Continuous-time Markov decision processes
   - Restriction to local uniformity?
   - Uniformization for time-dependent schedulers?
2. Interactive Markov chains
   - Computing long run average measures?
   - Support for reward extensions of CSL?
3. Generalized stochastic Petri nets
   - Allow for partial weight specifications?
   - Extension towards stochastic activity networks?


THANK
YOU

List of Publications

Published papers

1. *Model Checking Interactive Markov Chains.*
   Zhang, Neuhäußer. TACAS 2010
2. *Delayed Nondeterminism in Continuous-Time Markov Decision Processes.*
   Neuhäußer, Stoelinga, Katoen. FoSSaCS 2009
3. *Compositional Abstraction for Stochastic Systems.*
   Katoen, Klink, Neuhäußer. FORMATS 2009
4. *Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes.*
   Neuhäußer, Katoen. CONCUR 2007
5. *Abstraction and Model Checking of Core Erlang Programs in Maude.*
   Neuhäußer, Noll. WRLA 2007

The pipeline

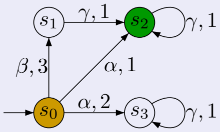
1. *Time-Bounded Reachability in Continuous-Time Markov Decision Processes.*
   Neuhäußer, Zhang. submitted to LICS 2010
2. *GSPN model checking despite confusion.*
   Hermanns, Katoen, Neuhäußer, Zhang. submitted to ICATPN 2010

Are time-dependent schedulers necessary?

Time-abstract vs. time dependent schedulers

- Why not positional schedulers?
- ... or time-abstract schedulers?

Are our schedulers really better?

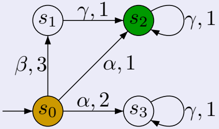


Are time-dependent schedulers necessary?

Time-abstract vs. time dependent schedulers

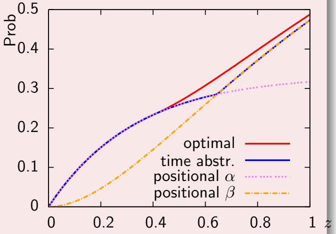
- Why not positional schedulers?
- ... or time-abstract schedulers?

Are our schedulers really better?



Maximum probability to reach state s_2 in ≤ 1 time unit

Yes, they are!

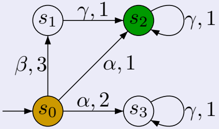


Are time-dependent schedulers necessary?

Time-abstract vs. time dependent schedulers

- Why not positional schedulers?
- ... or time-abstract schedulers?

Are our schedulers really better?

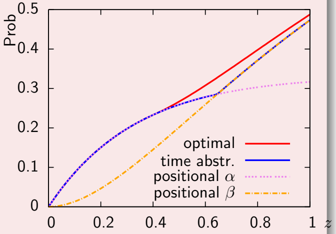


Maximum probability to reach state s_2 in ≤ 1 time unit

Why is this:
Generic scheduler decides upon leaving s_0:

- If long time remains: choose β
- If few time remains: choose α.

Time-abstract schedulers cannot do this!

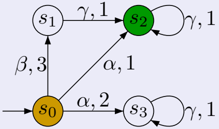


Are time-dependent schedulers necessary?

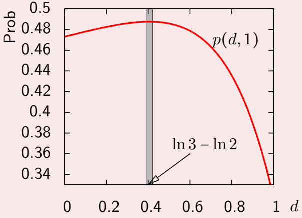
Time-abstract vs. time dependent schedulers

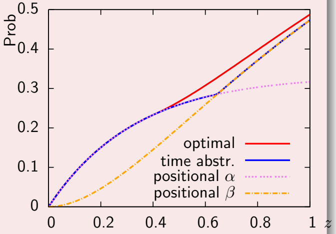
- Why not positional schedulers?
- ... or time-abstract schedulers?

Are our schedulers really better?



Maximum probability to reach state s_2 in ≤ 1 time unit



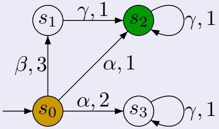


Are time-dependent schedulers necessary?

Time-abstract vs. time dependent schedulers

- Why not positional schedulers?
- ... or time-abstract schedulers?

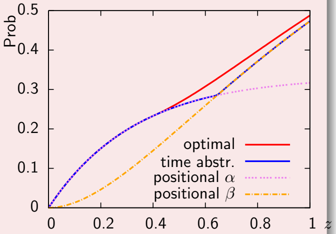
Are our schedulers really better?



Maximum probability to reach state s_2 in ≤ 1 time unit

Optimal scheduler for time-bound $z = 1$:

$$D(s_0, t_0) = \begin{cases} \{\alpha \mapsto 1\} & \text{if } (1 - t_0) \leq \ln 3 - \ln 2 \\ \{\beta \mapsto 1\} & \text{otherwise.} \end{cases}$$



A simpler class of optimal schedulers

Total time positional schedulers

A scheduler $D : Paths^\star \times \mathbb{R}_{\geq 0} \to Distr(Act)$ is total time positional iff
$\forall \pi, \pi' \in Paths^\star .\ \forall t, t' \in \mathbb{R}_{\geq 0}$.

$$\Big(last(\pi) = last(\pi') \wedge \Delta(\pi) + t = \Delta(\pi') + t' \Big) \Rightarrow D(\pi, t) = D(\pi', t').$$

$\Delta(\pi)$ is the total time spent on π.

Intuition

Total time positional schedulers only depend on

- the current state $last(\pi)$
- the total amount of time $\Delta(\pi) + t$ that has passed.

Optimality of TTPD schedulers

There exists $D \in TTPD$ such that $Pr^{\omega}_{s,D^*}(\Diamond^{[0,z]} G) = p^{G}_{max}(s, z)$.

A simpler class of optimal schedulers

Total time positional schedulers

A scheduler $D : Paths^{\star} \times \mathbb{R}_{\geq 0} \rightarrow Distr(Act)$ is total time positional iff $\forall \pi, \pi' \in Paths^{\star}.\ \forall t, t' \in \mathbb{R}_{\geq 0}$.

$$\Big(last(\pi) = last(\pi') \wedge \Delta(\pi) + t = \Delta(\pi') + t' \Big) \Rightarrow D(\pi, t) = D(\pi', t').$$

$\Delta(\pi)$ is the total time spent on π.

Intuition:

Total time positional schedulers only depend on

1. the current state $last(\pi)$
2. the total amount of time $\Delta(\pi) + t)$ that has passed.

A simpler class of optimal schedulers

Total time positional schedulers

A scheduler $D : Paths^\star \times \mathbb{R}_{\geq 0} \to Distr(Act)$ is total time positional iff $\forall \pi, \pi' \in Paths^\star.\ \forall t, t' \in \mathbb{R}_{\geq 0}.$

$$\Big(last(\pi) = last(\pi') \wedge \Delta(\pi) + t = \Delta(\pi') + t'\Big) \Rightarrow D(\pi, t) = D(\pi', t').$$

$\Delta(\pi)$ is the total time spent on π.

Intuition:

Total time positional schedulers only depend on

1. the current state $last(\pi)$
2. the total amount of time $\Delta(\pi) + t)$ that has passed.

Optimality of TTPD schedulers

There exists $D \in TTPD$ such that $Pr^\omega_{s, D^z}\big(\Diamond^{[0,z]} G\big) = p^G_{max}(s, z)$.