Model Checking Nondeterministic and Randomly Timed Systems

Martin R. Neuhäußer¹,²

¹RWTH Aachen University, Germany
²University 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

requirement

formalizing

property specification

model checking

satisfied

system

modeling

system model

violated
The model checking approach

- requirement
  - formalizing
  - property specification
- system
  - modeling
  - system model

model checking

- satisfied
- violated
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \Box \neg \text{collision} \]
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \square \neg \text{collision} \]
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \Box \neg \text{collision} \]
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \square \neg \text{collision} \]
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \Box \neg collision \]
Formal methods in computer science

The model checking approach

\[ \Phi = \forall \square \neg \text{collision} \]

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

⇒ 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 model checking

Extend models with probabilities to describe real-world systems

⇒ 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 model checking

Extend models with probabilities to describe real-world systems

⇒ **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 $\equiv$ 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 \equiv 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 \( \equiv \) 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.
Motivation: The stochastic job scheduling problem

Application: Load balancing of a bank’s website

1. Customer request $\equiv$ 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 $z$!
Preemptive scheduling of 4 jobs onto 2 processors

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

\[ 1, 2, 3, 4 \]

In this talk:
1. Compute the maximum probability to finish all jobs before time \( z \).
2. Synthesize optimal schedule to achieve this probability.
Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

1, 2, 3, 4

1, 3, 4

In this talk:
- Compute the maximum probability to finish all jobs before time $z$.
- Synthesize optimal schedule to achieve this probability.
Preemptive scheduling of 4 jobs onto 2 processors

In this talk:

1. Compute the maximum probability to finish all jobs before time $t$.
2. Synthesize optimal schedule to achieve this probability.
In this talk:

- Compute the maximum probability to finish all jobs before time $z$.
- Synthesize optimal schedule to achieve this probability.
In this talk:

1. Compute the maximum probability to finish all jobs before time $z$.
2. Synthesize optimal schedule to achieve this probability.
Formalizing the stochastic job scheduling problem

Preemptive scheduling of 4 jobs onto 2 processors

In this talk:
1. Compute the maximum probability to finish all jobs before time $t$.
2. Synthesize optimal schedule to achieve this probability.
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:

- 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.
Random timing and nondeterminism in CTMDPs

Initial state: $s_0$
Actions: $Act(s_0) = \{\alpha, \beta\}$
Choice is nondeterministic!
Transition rates: $R(s_0, \alpha, s_3) = 2$
Continuous-time Markov decision processes (CTMDPs)

Random timing and nondeterminism in CTMDPs

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

If action $\beta$ is chosen: Only one transition available

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

Probability to move before $t$ time units:

$$P(X_1 \leq t) = \int_0^t 3 \cdot e^{-3x} \, dx = (1 - e^{-3t})$$
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: $R(s_0, \alpha, s_3) = 2$

If action $\alpha$ 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: \(\text{Act}(s_0) = \{\alpha, \beta\}\)

Choice is nondeterministic!

Transition rates: \(R(s_0, \alpha, s_3) = 2\)

If action \(\alpha\) is chosen: Race condition

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

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

The transition that executes first, wins:

1. Time spent in \(s_0\): \(\min(X_2, X_3) \sim \text{Exp}(1 + 2)\)

Exit rate: \(E(s, \alpha) = \sum_{s' \in S} R(s, \alpha, s') = 1 + 2\)

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

\[ \forall s \in S. \forall \alpha, \beta \in \text{Act}(s). \ E(s, \alpha) = E(s, \beta). \]

Exit rate \( E(s) \) independent of action!
The subclass of locally uniform CTMDPs

Restriction to local uniformity

A CTMDP is locally uniform iff

\[ \forall s \in S. \forall \alpha, \beta \in \text{Act}(s). \quad E(s, \alpha) = E(s, \beta). \]

Exit rate \( E(s) \) independent of action!
The subclass of locally uniform CTMDPs

Restriction to local uniformity

A CTMDP is locally uniform iff

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

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 [Neuhaüßer, Stoelinga, Katoen '09]

A scheduler is a measurable mapping

\[ D : \textit{Paths} \times \mathbb{R}_{\geq 0} \rightarrow \textit{Distr}(\textit{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \textit{Act}(\textit{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^{\pi, s} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = s_0 \]

wait in \( s_0 \) for \( X_0 \sim \text{Exp}(3) \) time units

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

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( Pr^{\pi, s}_D \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = s_0 \xrightarrow{t_0} ? \]

upon leaving \( s_0 \): nondeterministic choice!

Time & history dependent schedulers [Neuhausser, Stoelinga, Katoen ’09]

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^{D,s} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = s_0 \xrightarrow{\beta, t_0} \]?

\[ D(s_0, t_0) = \beta \]

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

A scheduler is a measurable mapping

\[ D : Paths^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(Act) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( Pr_D^s(\cdot) \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = (s_0 \xrightarrow{\beta, t_0} s_1) \]

wait in \( s_1 \) for \( X_1 \sim \text{Exp}(2) \) time units

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

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_\geq 0 \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^D_{s, \omega} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = (s_0, \beta, t_0) \xrightarrow{t_1} \gamma \]

upon leaving \( s_1 \): only \( \gamma \) available.

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

A scheduler is a measurable mapping

\[ D : \text{Paths} \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr(Act)} \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act(last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \text{Pr}^{\pi,s} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = \begin{align*}
  &s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} ? \end{align*} \]

Race:

\[ \frac{R(s_1, \gamma, s_2)}{E(s_1)} = \frac{1}{2} \text{ chance to move to } s_0 \]

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

A scheduler is a measurable mapping

\[ D : Paths \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(Act) \]

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

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( Pr^D_{\pi}(s) \).
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = \begin{array}{c} s_0 \xrightarrow{\beta, t_0} s_1 \xrightarrow{\gamma, t_1} s_0 \end{array} \]

wait in \( s_0 \) for \( X_0 \sim \text{Exp}(3) \) time units

Time & history dependent schedulers [Neuhauser Stoelinga Katoen '09]

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \text{Pr}^{\pi} \)
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 \( \alpha \) and \( \beta \)

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

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^{s}_{\pi, D} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = (s_0, \beta, t_0) \xrightarrow{\gamma, t_1} s_0, \alpha, t_2 \]

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

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

A scheduler is a measurable mapping

\[ D : Paths^\ast \times \mathbb{R}_{\geq 0} \to \text{Distr}(Act) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in Act(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( Pr^D_{\omega, s} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = (s_0, \beta, t_0) \xrightarrow{\gamma, t_1} (s_0, \alpha, t_2) \xrightarrow{?} \]

Race:

\[ \frac{R(s_0, \alpha, s_2)}{E(s_0)} = \frac{1}{3} \] chance to move to \( s_2 \)

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

A scheduler is a measurable mapping

\[ D : \text{Paths} \times \mathbb{R}_{\geq 0} \to \text{Distr}(\text{Act}) \]

such that \( D(\pi, t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^{\pi, s} \)
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 [Neuhaußer, Stoelinga, Katoen '09]

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

such that \( D(\pi,t)(\alpha) > 0 \Rightarrow \alpha \in \text{Act}(\text{last}(\pi)) \)

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( \Pr^{\pi,s} \)
Resolving nondeterministic choices

Resolving nondeterminism by schedulers:

\[ \pi = (s_0, \beta, t_0, s_1, \gamma, t_1, s_0, \alpha, t_2, s_2) \]

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

A scheduler is a measurable mapping

\[ D : Paths^* \times \mathbb{R}_{\geq 0} \rightarrow Distr(Act) \]

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

Resolving nondeterminism by schedulers:

\[ \pi = (s_0, \beta, t_0, s_1, \gamma, t_1, s_0, \alpha, t_2, s_2) \]

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

A scheduler is a measurable mapping

\[ D : \text{Paths}^* \times \mathbb{R}_{\geq 0} \rightarrow \text{Distr}(\text{Act}) \]

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

The probability measure

Scheduler \( D \) & initial state \( s \) induce unique probability measure \( P_{\pi, s}^{\omega, D} \).
Finishing all jobs within \( \approx \) time units:
Modeling the sJSP as a CTMDP

Finishing all jobs within $\tau$ time units:

\[ \begin{align*} 
1, 2 & \quad \lambda_2 \\
3, 4 & \quad \lambda_3 \\
\end{align*} \]

\[ \begin{align*} 
2, 3, 4 & \quad \lambda_1 \\
1, 2, 4 & \\
\end{align*} \]
Modeling the sJSP as a CTMDP

Finishing all jobs within $z$ time units:

\[\begin{array}{c}
\frac{1,2}{3,4} \\
\alpha_2 \\
\lambda_2 \\
\lambda_3 \\
\lambda_4 \\
\lambda_1 \\
\lambda_4 \\
\lambda_3 \\
\lambda_2 \\
\lambda_1 \\
\lambda_2 \\
\lambda_1 \\
\emptyset
\end{array}\]
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:

![Diagram showing the CTMDP model for the sJSP](image-url)
Modeling the sJSP as a CTMDP

Finishing all jobs within $z$ time units:

All scheduling strategies represented in the CTMDP.

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

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

$\ldots$
Modeling the sJSP as a CTMDP

Finishing all jobs within \( z \) time units:

All scheduling strategies represented in the CTMDP.

\[
\alpha_1 : (1 \mapsto \{3, 4\}, 3 \mapsto \{2, 4\}) \quad \alpha_2 : (1 \mapsto \{2, 4\}, 3 \mapsto \{1, 4\}) \quad \ldots
\]

Properties:

1. CTMDP combines nondeterministic choices and stochastic timing.
2. The CTMDP model is locally uniform.
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
Time-bounded reachability in the sJSP

Time-bounded reachability probabilities

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

The time-bounded reachability event:

$$\diamondsuit^{[0,z]} G = \{ \pi \in \text{Paths}^\omega | \exists t \in [0,z]. \pi @ t \in G \}$$

Maximum time-bounded reachability probability

$$p^C_{\text{max}}(s, z) = \sup_D Pr^D_s(\diamondsuit^{[0,z]} G)$$
Computing the maximum time-bounded reachability probability

How to compute $p_{\text{max}}^c$?

Idea: Characterize $p_{\text{max}}^c$ as a fixed-point!
Computing the maximum time-bounded reachability probability

How to compute $p_{max}^C$?

Idea: Characterize $p_{max}^C$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

$$
\Omega(F)(s, z) = \int_0^\infty E(s)e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
$$
Computing the maximum time-bounded reachability probability

**How to compute \( p_{max}^C \)?**

**Idea:** Characterize \( p_{max}^C \) as a fixed-point!

**A higher operator for maximum time-bounded reachability**

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

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

\[
\Omega(F)(s, z) = \int_0^\infty E(s) e^{-(s, z)} \cdot \max_{\alpha \in act} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
\]
Computing the maximum time-bounded reachability probability

How to compute $p_{\text{max}}^C$?

Idea: Characterize $p_{\text{max}}^C$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

$$
\Omega(F)(s, z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
$$
Computing the maximum time-bounded reachability probability

How to compute $p_{\text{max}}^C$?

Idea: Characterize $p_{\text{max}}^C$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

$$
\Omega(F)(s, z) = \int_0^z E(s) e^{-E(s) t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
$$
Computing the maximum time-bounded reachability probability

How to compute $p^C_{max}$?

Idea: Characterize $p^C_{max}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

$$
\Omega(F)(s, z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z-t) \, dt.
$$
Computing the maximum time-bounded reachability probability

How to compute $p_{\text{max}}^{C}$?

Idea: Characterize $p_{\text{max}}^{C}$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

$$
\Omega(F)(s, z) = \int_{0}^{z} E(s)e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
$$
Computing the maximum time-bounded reachability probability

How to compute $p_{\text{max}}^C$?

Idea: Characterize $p_{\text{max}}^C$ as a fixed-point!

A higher operator for maximum time-bounded reachability

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

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

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

Fixed point characterization

The function $p_{\text{max}}^C(s, z)$ is the least fixed point of $\Omega$. 

Martin R. Neuhäußer (RWTH Aachen) 
Nondeterministic & Stochastic Model Checking
January 25, 2010 14 / 35
Applying the fixed point characterization directly

Fixed point characterization

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

\[
\Omega(F)(s, z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot F(s', z - t) \, dt.
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

---

[Diagram showing a state transition diagram with states $s_0, s_1, s_2, s_3$ and transitions labeled with $\alpha, \beta, \gamma$ and numbers 1, 2, 3.]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

\[
\begin{align*}
F_1(s_0, z) &= \Omega(F_0)(s_0, z) = 0. \\
F_1(s_1, z) &= \Omega(F_0)(s_1, z) = 0. \\
F_1(s_2, z) &= \Omega(F_0)(s_2, z) = 1. \\
F_1(s_3, z) &= \Omega(F_0)(s_0, z) = 0.
\end{align*}
\]
Applying the fixed point characterization directly

Fixed point characterization

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

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

Solving the reachability problem analytically

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

$$F_2(s_0, z) = \Omega(F_1)(s_0, z) = \int_0^z 3e^{-3t} \cdot \max (, ) \, dt.$$
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

$$
F_2(s_0, z) = \Omega(F_1)(s_0, z) = \int_0^z 3e^{-3t} \cdot \max \left( \frac{1}{3} \cdot F_1(s_2, z - t) \right) \, dt.
$$
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

**Fixed point characterization**

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

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

**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:

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

**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:

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

### Solving the reachability problem analytically

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

\[
F_2(s_0, z) = \Omega(F_1)(s_0, z) = \frac{1}{3} \left( 1 - e^{-3z} \right).
\]
Applying the fixed point characterization directly

Fixed point characterization

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

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

Solving the reachability problem analytically

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

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

\[
F_2(s_1, z) = \Omega(F_1)(s_1, z) = \int_0^z e^{-t} \cdot F_1(s_2, z - t) \, dt.
\]
Applying the fixed point characterization directly

Fixed point characterization

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

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

Solving the reachability problem analytically

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

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

\[
F_2(s_1, z) = \Omega(F_1)(s_1, z) = \int_0^z e^{-t} \cdot F_1(s_2, z - t) \, dt.
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**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:

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

**Solving the reachability problem analytically**

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

\[
\begin{align*}
F_2(s_0, z) &= \Omega(F_1)(s_0, z) = \frac{1}{3} \left( 1 - e^{-3z} \right), \\
F_2(s_1, z) &= \Omega(F_1)(s_1, z) = 1 - e^{-z}, \\
F_2(s_2, z) &= \Omega(F_1)(s_2, z) = 1, \\
F_2(s_3, z) &= \Omega(F_1)(s_3, z) = 0.
\end{align*}
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

![Diagram of a model with states and transitions](image.png)
Applying the fixed point characterization directly

### Fixed point characterization

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

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

### Solving the reachability problem analytically

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

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

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

### Fixed point characterization
If \( s \in G \): \( \Omega(F)(s, z) = 1 \). Otherwise:

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

### Solving the reachability problem analytically
Fixed-point computation: \( \forall s \in S. \ F_0(s, z) = 0 \).

\[
F_3(s_0, z) = \Omega(F_2)(s_0, z) = \int_0^z 3e^{-3t} \cdot \max \left( \frac{1}{3}, F_2(s_1, z - t) \right) \, dt.
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

\( F_3(s_0, z) = \Omega(F_2)(s_0, z) = \int_0^z 3e^{-3t} \cdot \max \left( \frac{1}{3}, 1 - e^{-z} \right) \, dt \).
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

\[
F_3(s_1, z) = \Omega(F_2)(s_1, z) = \int_0^z e^{-t} \cdot F_2(s_2, z - t) \, dt.
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

\[
F_3(s_1, z) = \Omega(F_2)(s_1, z) = \int_0^z e^{-t} \cdot F_2(s_2, z - t) \, dt.
\]
Applying the fixed point characterization directly

**Fixed point characterization**

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

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

**Solving the reachability problem analytically**

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

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

\[
F_3(s_1, z) = \Omega(F_2)(s_1, z) = 1 - e^{-z}.
\]
Applying the fixed point characterization directly

### Fixed point characterization

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

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

### Solving the reachability problem analytically

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

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

![Diagram](image_url)
Applying the fixed point characterization directly

Fixed point characterization

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

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

Solving the reachability problem analytically

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

\[
\begin{align*}
F_3(s_0, z) &= \Omega(F_2)(s_0, z) = \int_0^z 3e^{-3t} \cdot \max \left( \frac{1}{3}, 1 - e^{-z} \right) \, dt. \\
F_3(s_1, z) &= \Omega(F_2)(s_1, z) = 1 - e^{-z}.
\end{align*}
\]

\[
\begin{align*}
F_3(s_2, z) &= \Omega(F_2)(s_2, z) = 1. \\
F_3(s_3, z) &= \Omega(F_2)(s_3, z) = 0.
\end{align*}
\]

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:

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

**Solving the reachability problem analytically**

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

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

\[
F_3(s_1, z) = \Omega(F_2)(s_1, z) = 1 - e^{-z}.
\]

\[
F_3(s_2, z) = \Omega(F_2)(s_2, z) = 1.
\]

\[
F_3(s_3, z) = \Omega(F_2)(s_3, z) = 0.
\]

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

For \( z = 1 \): \( p_{\max}^c(s_0, 1) = 1 + \frac{19}{24} e^{-3} - \frac{3}{2} e^{-1} \approx 0.48759. \)
What is achieved so far:

Analytical solution

Allows to compute $p_{max}^{C}(s, z)$ for small problem instances.
What is achieved so far:

### Analytical solution

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

### Disadvantages:

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

⇒ Fixed-point characterization not suitable for an algorithmic solution.
What is achieved so far:

**Analytical solution**

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

**Disadvantages:**

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

⇒ 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 $p_{max}^C$ to step-bounded reachability $p_{max}^{C\tau}$ in MDPs.
Each discrete step corresponds to a time-interval of length $\tau$. 
A discretization that computes $p_{\text{max}}(s, z)$

Reduce $p_{\text{max}}^C$ to step-bounded reachability $p_{\text{max}}^{C_{\tau}}$ in MDPs.
Each discrete step corresponds to a time-interval of length $\tau$.

Continuous-time vs. discrete-time Markov decision processes

Continuous-time MDP $C$

Discrete-time MDP $C_{\tau}$

Exponential distributions

Reachability within time $z$ $\equiv$

Discrete probability distributions

Reachability in $\frac{z}{\tau}$ steps!
Recall the fixed-point characterization:

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

$$p_{max}^C(s, z) = \int_0^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s', \in S} \mathbf{P}(s, \alpha, s') \cdot p_{max}^C(s', z-t) \, dt.$$
Recall the fixed-point characterization:

The function $p_{\text{max}}^C(s, z)$ is the least fixed point of $\Omega$: If $s \notin G$, then

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

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

Choose $\tau \ll z$ and split $p_{\text{max}}^C(s, z)$ accordingly:
Recall the fixed-point characterization:

The function \( p_{\text{max}}^c(s, z) \) is the least fixed point of \( \Omega \): If \( s \notin G \), then

\[
p_{\text{max}}^c(s, z) = \int_0^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot p_{\text{max}}^c(s', z - t) \, dt.
\]

The idea for a discretization to compute \( p_{\text{max}}^c(s, z) \):

Choose \( \tau \ll z \) and split \( p_{\text{max}}^c(s, z) \) accordingly:

\[
A(s, z) = \int_0^\tau E(s) e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot p_{\text{max}}^c(s', z - t) \, dt
\]
Recall the fixed-point characterization:

The function $p_{\text{max}}^C(s, z)$ is the least fixed point of $\Omega$: If $s \notin G$, then

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

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

Choose $\tau \ll z$ and split $p_{\text{max}}^C(s, z)$ accordingly:

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

$$B(s, z) = \int_\tau^z E(s)e^{-E(s)t} \cdot \max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot p_{\text{max}}^C(s', z - t) \, dt$$

$$= e^{-E(s)\tau} \cdot p_{\text{max}}^C(s, z - \tau).$$
Recall the fixed-point characterization:

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

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

The idea for a discretization to compute $p_{max}^C(s, z)$:

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

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

$$B(s, z) = \int_\tau^z E(s) e^{-E(s)t} \cdot \max_{\alpha \in Act} \sum_{s' \in S} P(s, \alpha, s') \cdot p_{max}^C(s', z - t) \, dt = e^{-E(s)\tau} \cdot p_{max}^C(s, z - \tau).$$

Relation to $p_{max}^C(s, z)$: $p_{max}^C(s, z) = A(s, z) + B(s, z)$.
Discretization II

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

- $A(s, z) = \text{Prob. to reach } G \text{ within time } z \text{ with } \geq 1 \text{ transitions in } [0, \tau]$.
Discretization II

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

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

  ![Diagram of A(s, z)]

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

  ![Diagram of B(s, z)]
A step-wise approximation of $p_{\text{max}}^C(s, z)$

A single step in the discretized MDP

CTMDP $C$ and step duration $\tau < z$ induce the discretized MDP $C_\tau$:

$$P_\tau(s, \alpha, s') = \begin{cases} 
(1 - e^{-E(s)\tau}) \cdot P(s, \alpha, s') & \text{if } s \neq s' \\
(1 - e^{-E(s)\tau}) \cdot P(s, \alpha, s) + e^{-\lambda(s)\tau} & \text{if } s = s'.
\end{cases}$$
A step-wise approximation of $p_{max}^C(s, z)$

A single step in the discretized MDP

CTMDP $C$ and step duration $\tau < z$ induce the discretized MDP $C_\tau$:

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

Theorem (Correctness of our reduction)

Let $C$ 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

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

1. $p_{max}^{C_\tau}(s, k)$ is the probability to reach $G$ in at most $k$ steps in $C_\tau$,  
2. $\lambda = max_{s \in S} E(s)$ is the maximum exit rate in $C$ and  
3. $k$ is the number of discretization steps.
Value iteration for discrete-time MDPs [Bellman '57]

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

$$
\vec{v}_0(s) = \begin{cases} 
1 & \text{if } s \in G \\
0 & \text{if } s \notin G
\end{cases} \quad \vec{v}_{n+1}(s) = \begin{cases} 
1 & \text{if } s \in G \\
\max_{\alpha \in \text{Act}} \sum_{s' \in S} P(s, \alpha, s') \cdot v_n(s') & \text{if } s \notin G
\end{cases}
$$

Then $p_{\max}^{c_T}(s, k) = \vec{v}_k(s)$. 
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: \( \varepsilon = 10^{-3} \)

Compute the number of discretization steps \( k \):

\[
\frac{(\lambda z)^2}{2\varepsilon} \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{2}{k} = 0.0002
\]
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 \( k \)

\[
\frac{(\lambda z)^2}{2k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.0002.
\]
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 \( k \)

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

Value iteration to compute \( p_{max}^c(s, k) \)

\( \tilde{v}_0 = (0, 0, 1, 0) \)
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** \( k \)

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

**Value iteration to compute** \( p^{c_{\tau}}_{\max}(s, k) \)

\[
\vec{v}_0 = (0, 0, 1, 0) \quad \vec{v}_1 = \left( \max\left( \right), \right), \left( \right), \left( \right)
\]
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** \( k \)

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

**Value iteration to compute** \( p_{\text{max}}^{c_{\tau}}(s, k) \)

\[
\tilde{v}_0 = (0, 0, 1, 0), \\
\tilde{v}_1 = \left( \max\left(\frac{1}{3} (1 - e^{-3\tau}), \right), \right), \quad \text{with}, \quad \text{variables}
\]
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 \( k \)

\[
\frac{(\lambda z)^2}{2^k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.0002.
\]

Value iteration to compute \( p_{max}^{c_\tau} (s, k) \)

\[
\begin{align*}
\vec{v}_0 &= (0, 0, 1, 0) \\
\vec{v}_1 &= \left( \max \left( \frac{1}{3} (1 - e^{-3\tau}), \right), \right), \quad , \quad , \quad)
\end{align*}
\]
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 $k$

\[
\frac{(\lambda z)^2}{2^k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.0002.
\]

### Value iteration to compute $p_{\tau}^{C_\tau}(s, k)$

\[
\bar{v}_0 = (0, 0, 1, 0)
\]
\[
\bar{v}_1 = \left( \max \left( \frac{1}{3} \left( 1 - e^{-3\tau} \right), 0 \right), \quad \text{, , , } \right)
\]
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 \( k \)

\[
\frac{(\lambda z)^2}{2^k} \leq \varepsilon \quad \Rightarrow \quad k \geq 4500 \quad \Rightarrow \quad \tau = \frac{z}{k} = 0.0002.
\]

Value iteration to compute \( p_{\text{max}}^{c_\tau}(s, k) \)

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

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

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

Value iteration to compute $p_{\text{max}}^c(s, k)$

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

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

Value iteration to compute \( p^c_{\tau} (s, k) \)

\[
\begin{aligned}
\vec{v}_0 &= (0, 0, 1, 0) \\
\vec{v}_1 &= \left( \frac{1}{3} \left( 1 - e^{-3\tau} \right), (1 - e^{-\tau}), \ldots \right)
\end{aligned}
\]
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 \( p^{\tau}_{\text{max}}(s, k) \)

\[
\tilde{v}_0 = (0, 0, 1, 0) \\
\tilde{v}_1 = \left( \frac{1}{3} (1 - e^{-3\tau}), (1 - e^{-\tau}), \ldots \right)
\]
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 $p_{c_\tau}^{max}(s, k)$

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

\[
\frac{(\lambda z)^2}{2k} \leq \varepsilon \implies k \geq 4500 \implies \tau = \frac{z}{k} = 0.000\overline{2}.
\]

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

\[
\vec{v}_0 = (0, 0, 1, 0) \\
\vec{v}_1 = \left( \frac{1}{3} \left(1 - e^{-3\tau}\right), (1 - e^{-\tau}), 1, \right)
\]
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 \( p_{C\tau}^{c}(s, k) \)

\[
\vec{v}_0 = (0, 0, 1, 0)
\]
\[
\vec{v}_1 = \left( \frac{1}{3} \left(1 - e^{-3\tau}\right), (1 - e^{-\tau}), 1, 0\right)
\]
### 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 \( k \)

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

#### Value iteration to compute \( p^c_{max}(s, k) \)

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

\[
\ldots
\]
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 \( p_{\max}^{c_\tau}(s, k) \)

\[
\tilde{v}_0 = (0, 0, 1, 0) \\
\tilde{v}_1 = \left( \frac{1}{3} \left( 1 - e^{-3\tau} \right), (1 - e^{-\tau}), 1, 0 \right) \\
\ldots
\]

Result: \( p_{\max}^{c_\tau}(s_0, 4500) = \tilde{v}_{4500}(s_0) \approx 0.487 \)
Complexity of the discretization approach

Complexity

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

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

**Overall complexity:** $\mathcal{O}\left(|\mathcal{C}| \cdot \frac{(\lambda z)^2}{\varepsilon}\right)$. 

Martin R. Neuhäußer (RWTH Aachen)
Nondeterministic & Stochastic Model Checking
January 25, 2010 23 / 35
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

\[ \text{Prob}(z) \]

\[ \lambda_1, \lambda_2, \lambda_3, \lambda_4 \]

\[ \alpha_1, \alpha_2, \alpha_3, \alpha_4 \]

\[ \alpha_0 \]

\[ 0.25, 0.25, 0.25, 0.25 \]

\[ 1.50, 1.50, 1.50, 1.50 \]

\[ 0.25, 0.25, 0.25, 1.50 \]

\[ 0.25, 0.33, 1.25, 1.50 \]

\[ 0.25, 1.50, 1.50, 1.50 \]

\[ 0.75, 1.50, 1.50, 1.50 \]
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)

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4</td>
<td>1 ↦ {2, 3}</td>
</tr>
<tr>
<td></td>
<td>2 ↦ {1, 3}</td>
</tr>
<tr>
<td>1, 2, 3, 4</td>
<td>2 ↦ {1, 3}</td>
</tr>
<tr>
<td>1, 2, 3, 4</td>
<td>3 ↦ {1, 2}</td>
</tr>
<tr>
<td>1, 2, 3, 4</td>
<td>4 ↦ {1, 2}</td>
</tr>
</tbody>
</table>
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
Interactive Markov chains

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

Continuous-time MDP

Combines actions and rates.

Only one type of transitions:

Extending the discretization from CTMDPs to IMCs

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

Model Checking Interactive Markov Chains [Zhang, Neuhäuser ’10]

**Continuous-time MDP**

- Only one type of transitions:
  - \( s_0 \xrightarrow{\beta, 2} s_1 \)

**Interactive Markov Chain** [Hermanns’02]

- Two types of transitions:
  - Markovian \( s_0 \xrightarrow{2} s_1 \)
  - Interactive \( s_1 \xrightarrow{\alpha} s_2 \)

Maximal progress assumption!

Combines actions and rates.

Separates actions and rates.

Extending the discretization from CTMDPs to IMCs

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

Model Checking Interactive Markov Chains [Zhang, Neuhäuser ’10]

Continuous-time MDP

Interactive Markov Chain [Hermanns’02]

Combines actions and rates.

Separates actions and rates.

Only one type of transitions:

Two types of transitions:

• Markovian
• Interactive

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]

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

\[
\begin{array}{c}
\text{Places: } p_0, p_1, p_2, p_3 \\
\text{Input: } t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8 \\
\text{Output: } \lambda, \eta, \mu \\
\end{array}
\]
Model checking generalized stochastic Petri nets

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

- Places $P$
- Input, output and inhibitor arcs
- Tokens
Generalized stochastic Petri nets (GSPNs) [Marsan, Conte, Balbo ’84]

- Places $P$
- Timed transitions (exp. rate)
- Input, output and inhibitor arcs
- Tokens
Generalized stochastic Petri nets (GSPNs) [Marsan, Conte, Balbo ’84]

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

![Diagram of a generalized stochastic Petri net](image)
Generalized stochastic Petri nets (GSPNs) [Marsan, Conte, Balbo ’84]

- **Places** $P$
- **Timed transitions** (exp. rate)
- **Immediate transitions**
- **Input, output and inhibitor arcs**
  - **Tokens**
Generalized stochastic Petri nets (GSPNs) [Marsan, Conte, Balbo ’84]

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

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

1. Classical answer: Avoid this case by using weights!
2. New approach: Nondeterminism!
   Interpret reachability graph of a GSPN as an IMC!
A GSPN model for the dependable workstation cluster

Result of the nondeterministic analysis: System is 18% less reliable than predicted by earlier analysis!
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

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
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?
List of Publications

Published papers

1. **Model Checking Interactive Markov Chains.**
   Zhang, Neuhausser.  
   TACAS 2010

2. **Delayed Nondeterminism in Continuous-Time Markov Decision Processes.**
   Neuhausser, Stoelinga, Katoen.  
   FoSSaCS 2009

3. **Compositional Abstraction for Stochastic Systems.**
   Katoen, Klink, Neuhausser.  
   FORMATS 2009

4. **Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes.**
   Neuhausser, Katoen.  
   CONCUR 2007

5. **Abstraction and Model Checking of Core Erlang Programs in Maude.**
   Neuhausser, Noll.  
   WRLA 2007

The pipeline

1. **Time-Bounded Reachability in Continuous-Time Markov Decision Processes.**
   Neuhausser, Zhang.  
   submitted to LICS 2010

2. **GSPN model checking despite confusion.**
   Hermanns, Katoen, Neuhausser, 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 $\leq 1$ 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 $\leq 1$ time unit**

**Why is this:**
Generic scheduler decides upon leaving $s_0$:
- If long time remains: choose $\beta$
- If few time remains: choose $\alpha$.

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 $\leq 1$ time unit

\[ p(d, 1) = \ln 3 - \ln 2 \]
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 $\leq 1$ 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 $D : Paths^* \times \mathbb{R}_{\geq 0} \to Distr(Act)$ is total time positional iff
\[
\forall \pi, \pi' \in Paths^*. \forall t, t' \in \mathbb{R}_{\geq 0}.
\]
\[
\left( \text{last}(\pi) = \text{last}(\pi') \land \Delta(\pi) + t = \Delta(\pi') + t' \right) \Rightarrow D(\pi, t) = D(\pi', t').
\]

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

**Intuition:**
Total time positional schedulers only depend on
- the current state $\text{last}(\pi)$
- the total amount of time $\Delta(\pi) + t$ that has passed.

**Optimality of TTPD schedulers**

There exists $D \in \text{TTPD}$ such that $Pr_{\omega \in \mathcal{L}[0, z]}^\pi (G) = p_{\text{opt}}^z (\omega, \pi)$. 

Martin R. Neuhäuser (RWTH Aachen)  Nondeterministic & Stochastic Model Checking  January 25, 2010  35 / 35
A simpler class of optimal schedulers

Total time positional schedulers

A scheduler $D : Paths^* \times \mathbb{R}_{\geq 0} \rightarrow Distr(Act)$ is total time positional iff

$\forall \pi, \pi' \in Paths^*. \forall t, t' \in \mathbb{R}_{\geq 0}.$

\[
\left( \text{last}(\pi) = \text{last}(\pi') \land \Delta(\pi) + t = \Delta(\pi') + t' \right) \Rightarrow D(\pi, t) = D(\pi', t').
\]

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

Intuition:

Total time positional schedulers only depend on

1. the current state $\text{last}(\pi)$
2. the total amount of time $\Delta(\pi) + t$ that has passed.
A simpler class of optimal schedulers

Total time positional schedulers

A scheduler \( D : Paths^* \times \mathbb{R}_{\geq 0} \rightarrow Distr(Act) \) is total time positional iff
\[
\forall \pi, \pi' \in Paths^*. \forall t, t' \in \mathbb{R}_{\geq 0}.
\]
\[
\left( last(\pi) = last(\pi') \land \Delta(\pi) + t = \Delta(\pi') + t' \right) \Rightarrow D(\pi, t) = D(\pi', t').
\]
\( \Delta(\pi) \) is the total time spent on \( \pi \).

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 \( Pr_{s,D,z}^{\omega} (\Diamond^{[0,z]} G) = p^G_{\max} (s, z) \).