Finite State Stochastic and Continuous Time Models

Łukasz Kaiser and Daniel Willems

AlgoSyn Workshop

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .
- . . . not really **natural**

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .
- . . . not really **natural**

Key ingredients of our models:

- Distinguished finite set of **states**

What kind of Models do we want?

- Differential equations and Turing machines are great but ...
- ...not really **natural**

Key ingredients of our models:

- Distinguished finite set of **states**
- Someone taking **actions** possibly against an **adversary**

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .
- . . . not really **natural**

Key ingredients of our models:

- Distinguished finite set of **states**
- Someone taking **actions** possibly against an **adversary**
- **Stochastic** events can take place

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .
- . . . not really **natural**

Key ingredients of our models:

- Distinguished finite set of **states**
- Someone taking **actions** possibly against an **adversary**
- **Stochastic** events can take place
- It might be useful to model the **continuous flow of time**

What kind of Models do we want?

- Differential equations and Turing machines are great but . . .
- . . . not really **natural**

Key ingredients of our models:

- Distinguished finite set of **states**
- Someone taking **actions** possibly against an **adversary**
- **Stochastic** events can take place
- It might be useful to model the **continuous flow of time**
- **concurrency, players, communication, quantities, populations, . . .**

Basic Discrete Model

Definition

Basic discrete model $\mathcal{M} = (S, \mu, A, L)$ is given by:

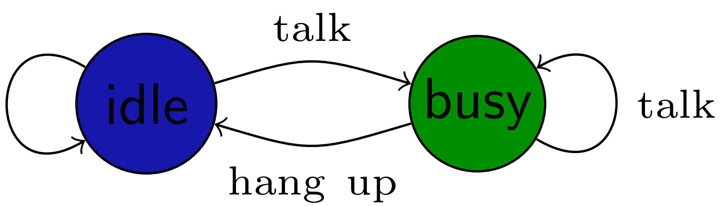
- finite set of states S
- the set of actions A
- transition function $\mu \ : \ S \times A \to S$
- labelling function (colours) $L \ : \ S \to 2^{AP}$

Basic Discrete Model

Definition

Basic discrete model $\mathcal{M} = (S, \mu, A, L)$ is given by:

- finite set of states S
- the set of actions A
- transition function $\mu \ : \ S \times A \to S$
- labelling function (colours) $L \ : \ S \to 2^{AP}$



Basic Discrete Game

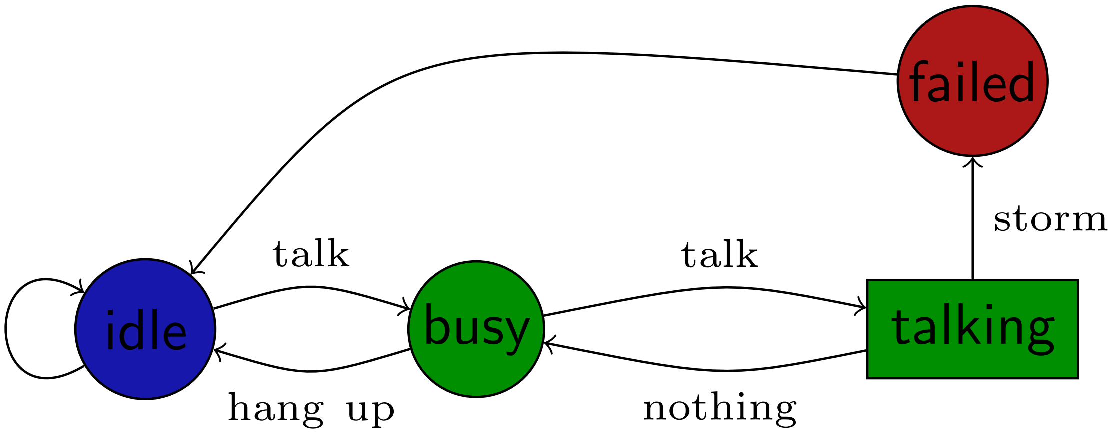
Definition

In a basic discrete game $\mathcal{G} = (S = S_0 \cup S_1, \mu, A, L)$ the set of states is partitioned into states of **Player 0** and states of **Player 1**.

Basic Discrete Game

Definition

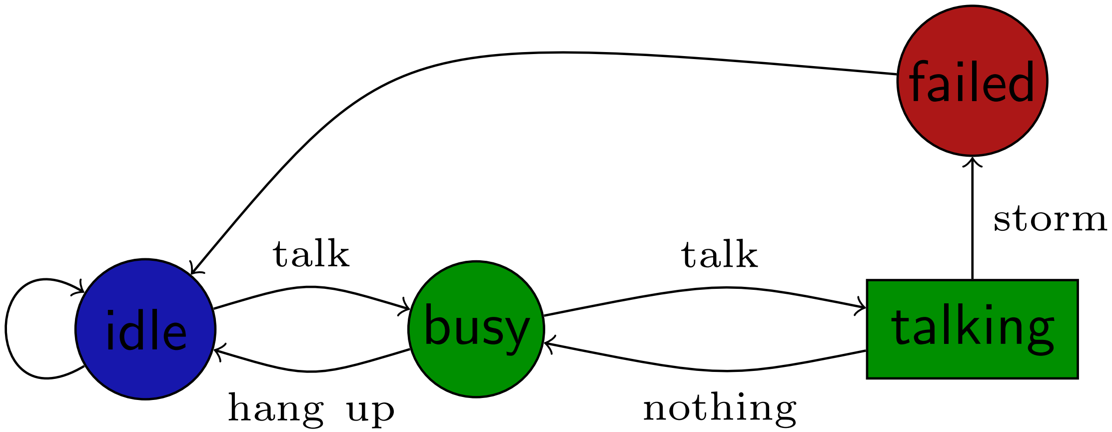
In a basic discrete game $\mathcal{G} = (S = S_0 \cup S_1, \mu, A, L)$ the set of states is partitioned into states of **Player 0** and states of **Player 1**.



Basic Discrete Game

Definition

In a basic discrete game $\mathcal{G} = (S = S_0 \cup S_1, \mu, A, L)$ the set of states is partitioned into states of **Player 0** and states of **Player 1**.



Question: can we repeatedly reach green without ever encountering red?

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

- just say what colours will be seen

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

- just say what colours will be seen repeatedly (infinitely often)

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

- just say what colours will be seen repeatedly (infinitely often)
- use a temporal logic, e.g. **LTL**

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

- just say what colours will be seen repeatedly (infinitely often)
- use a temporal logic, e.g. **LTL**
- use a fixpoint logic like μ-**calculus**

How to ask Questions?

General Pattern:
Do I have a **strategy** so that whatever my **adversary** does a **property** holds?

Different ways to specify properties:

- just say what colours will be seen repeatedly (infinitely often)
- use a temporal logic, e.g. **LTL**
- use a fixpoint logic like μ-**calculus**
- use the full **monadic second order logic**

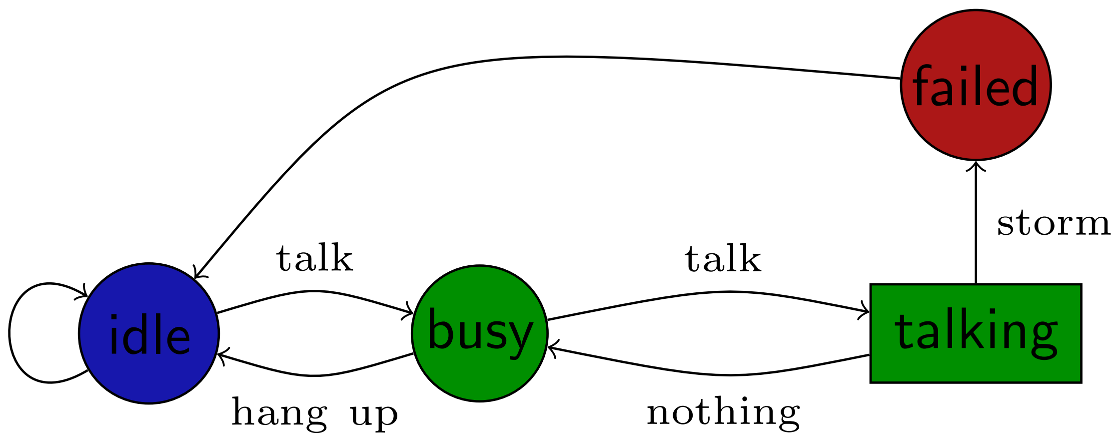
Temporal Logic

What does a temporal logic formula look like?

Temporal Logic

What does a temporal logic formula look like?

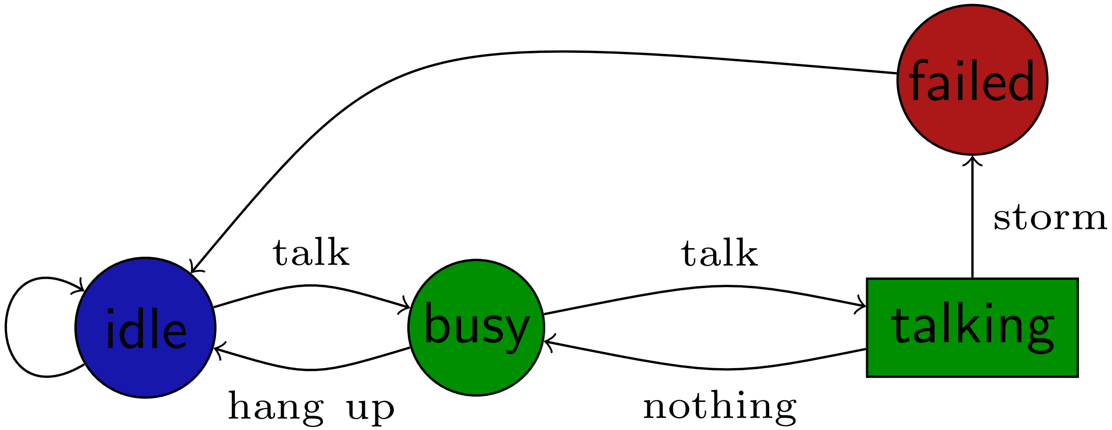
- specify current colour (R, G, B)



Temporal Logic

What does a temporal logic formula look like?

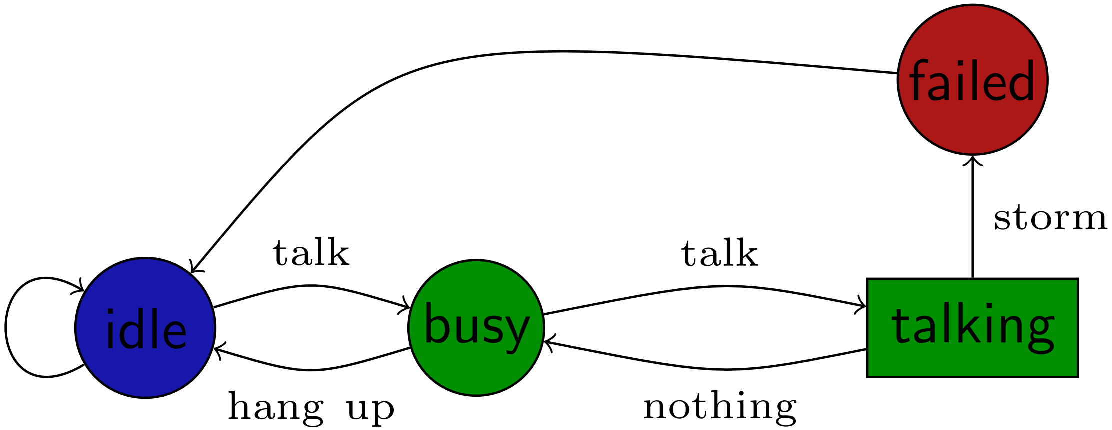
- specify current colour (R, G, B)
- talk about the **next** step $(\mathrm{X}R)$



Temporal Logic

What does a temporal logic formula look like?

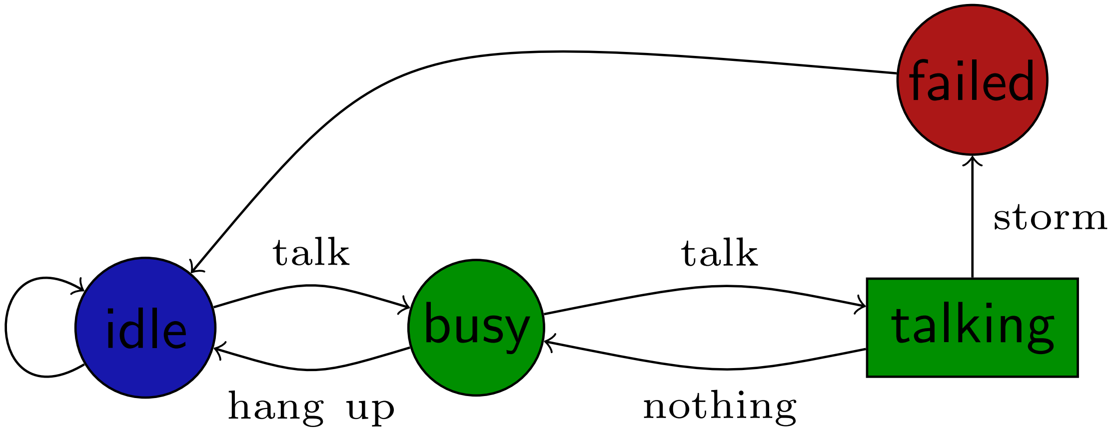
- specify current colour (R, G, B)
- talk about the **next** step $(\mathrm{X}R)$
- in the future **finally** an event happens $(\mathrm{F}G)$



Temporal Logic

What does a temporal logic formula look like?

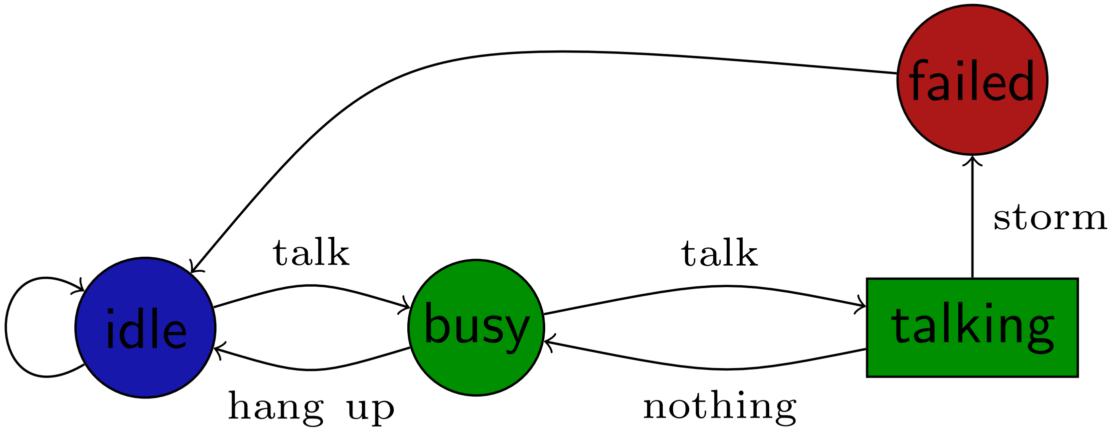
- specify current colour (R, G, B)
- talk about the **next** step $(\mathrm{X}R)$
- in the future **finally** an event happens $(\mathrm{F}G)$
- one property holds **until** another happens $(B\mathrm{U}G)$



Temporal Logic

What does a temporal logic formula look like?

- specify current colour (R, G, B)
- talk about the **next** step $(\mathrm{X}R)$
- in the future **finally** an event happens $(\mathrm{F}G)$
- one property holds **until** another happens $(B\mathrm{U}G)$
- **boolean combinations** of formulas $(\neg\mathrm{F}\neg\mathrm{F}G \wedge \neg\mathrm{F}R)$



Stochastic and Continuous Time Models

Important Property:

- future depends on the **state** but **not on the time**

Overview

Markovian models

- low level view: random variables
- high level view: extension of transition systems
- Markov property: behavior depends on current state
- time-homogenious: behavior does not depend on current time

In the following:

- 0-player vs. 1-player (deterministic vs. nondeterministic)
- discrete-time vs. continuous-time

Overview

Markovian models

- low level view: random variables
- high level view: extension of transition systems
- Markov property: behavior depends on current state
- time-homogenious: behavior does not depend on current time

In the following:

- 0-player vs. 1-player (deterministic vs. nondeterministic)
- discrete-time vs. continuous-time

Overview

Markovian models

- low level view: random variables
- high level view: extension of transition systems
- Markov property: behavior depends on current state
- time-homogenious: behavior does not depend on current time

In the following:

- 0-player vs. 1-player (deterministic vs. nondeterministic)
- discrete-time vs. continuous-time

DTMC - Discrete-time Markov chain

Type of game

- 0-player (deterministic)
- discrete-time

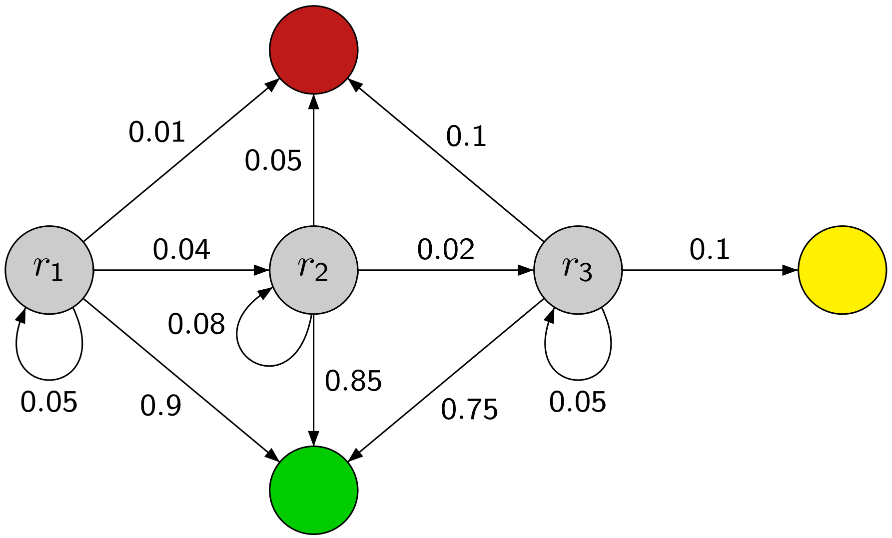
Definition

A DTMC $\mathcal{M} = (S, P, L)$ is given by:

- S, a non-empty, finite set of states
- $P : S \times S \rightarrow [0, 1]$, a matrix of one-step probabilities
- $L : S \rightarrow 2^{AP}$, a labeling function

DTMC - Discrete-time Markov chain

Example: biometric authentification



r_1	fingerprint scan
r_2	typing style recognition
r_3	retina scan
green	correct authentification
red	false authentification
yellow	automatic authentification not possible

- What is the probability for reaching *green* (or *red* or *yellow*)?
- How many steps are needed to reach one of the colored state with probability 90 percent?

DTMDP - Discrete-time Markov decision process

Type of game

- 1-player (nondeterministic)
- discrete-time

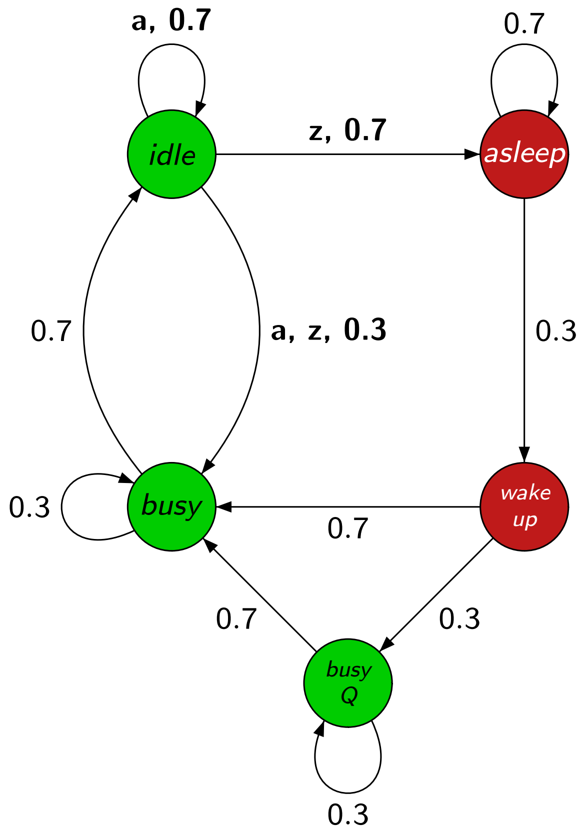
Definition

A DTMDP $\mathcal{M} = (S, P, A^1, L)$ is given by:

- S, a non-empty, finite set of states
- $P : S \times \mathbf{A}^1 \times S \to [0, 1]$, a matrix of one-step probabilities
- **$\mathbf{A}^1$, the set of actions**
- $L : S \to 2^{AP}$, a labeling function

DTMDP - Discrete-time Markov decision process

Example: power management



0.3	new request prob.
0.7	no new request prob.
a	keep activated
z	go sleeping
green	activated
red	sleeping

- Is there a scheduler which is sleeping at least 50 percent of the time, but serves more than 50 percent of all requests without delay?

CTMC - Continuous-time Markov chain

Type of game

- 0-player (deterministic)
- continuous-time

Definition

A CTMC $\mathcal{M} = (S, P, E, L)$ is given by:

- S is a non-empty, finite set of states
- $P : S \times S \to [0, 1]$ is a matrix of one-step probabilities
- $\mathbf{E} : \mathbf{S} \to \mathbb{R}_{\geq \mathbf{0}}$ **is a rate vector, describing the mean residence time**
- $L : S \to 2^{AP}$ is a labeling function

CTMC - Continuous-time Markov chain

One-step probabilities

- $P(s, s', t) = P(s, s') \cdot (1 - e^{-E(s) \cdot t})$
- $\lim_{t \to \infty} P(s, s', t) = P(s, s')$

Why exponential distribution?

- It is the only continuous distribution with Markov property!

CTMC - Continuous-time Markov chain

One-step probabilities

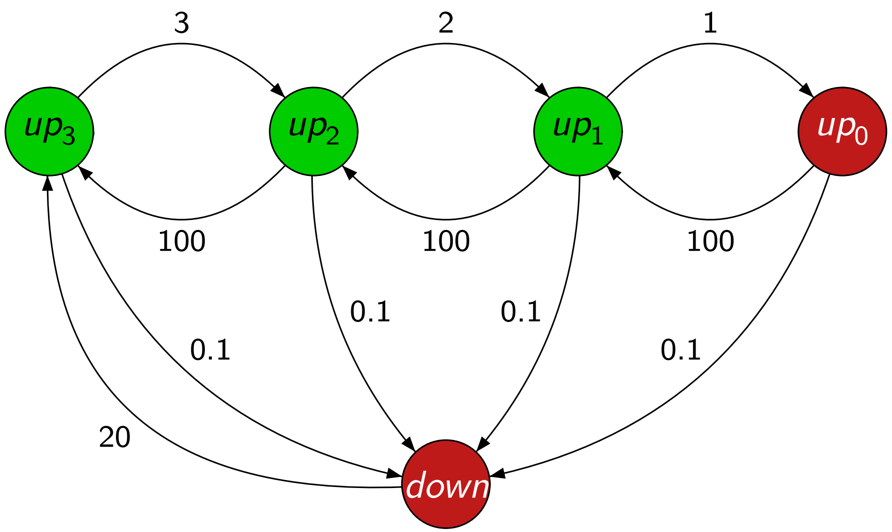
- $P(s, s', t) = P(s, s') \cdot (1 - e^{-E(s) \cdot t})$
- $\lim_{t \to \infty} P(s, s', t) = P(s, s')$

Why exponential distribution?

- It is the only continuous distribution with Markov property!

CTMC - Continuous-time Markov chain

Example: triple modular redundant system



up$_i$	i components up and central unit up
down	central unit down
green	system up
red	system down

- What is the probability to be in a *red* state on the long run?
- What is the probability to keep in *green* states within the next 10 time units?

CTMDP - Continuous-time Markov decision process

Type of game

- 1-player (nondeterministic)
- continuous-time

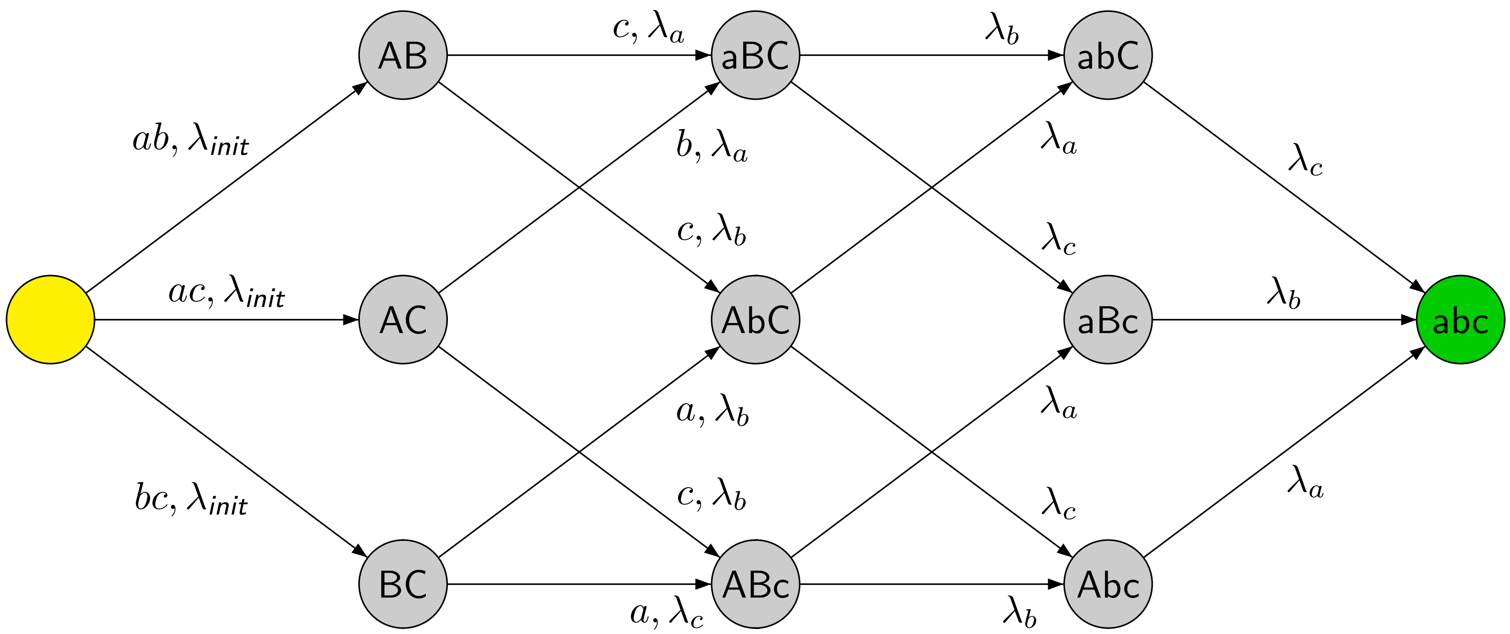
Definition

A CTMDP $\mathcal{M} = (S, P, E, A^1, L)$ is given by:

- S, a non-empty, finite set of states
- $P : S \times \mathbf{A^1} \times S \to [0, 1]$, a matrix of one-step probabilities
- $\mathbf{E : S \times A^1 \to \mathbb{R}_{\geq 0}}$ **is a rate vec., describing the mean residence time**
- $\mathbf{A^1}$**, the set of actions**
- $L : S \to 2^{AP}$, a labeling function

CTMDP - Continuous-time Markov decision process

Example: 2-processor job scheduling



- Find a *makespan* optimal scheduler.
- What savings are possible switching from the worst scheduler to the optimal scheduler w.r.t. the *makespan*?

High-level models

- Stochastic Petri-nets
- Queueing networks
- Production rule systems
- ...

Extensions

- Rewards
- Abstract models
- ...

Tools

- MRMC (DTMCs, CTMCs, Rewards)
- Prism (DTMCs, CTMCs, MDPs, *GUI*)
- . . .

High-level models

- Stochastic Petri-nets
- Queueing networks
- Production rule systems
- ...

Extensions

- Rewards
- Abstract models
- ...

Tools

- MRMC (DTMCs, CTMCs, Rewards)
- Prism (DTMCs, CTMCs, MDPs, *GUI*)
- ...

High-level models

- Stochastic Petri-nets
- Queueing networks
- Production rule systems
- ...

Extensions

- Rewards
- Abstract models
- ...

Tools

- MRMC (DTMCs, CTMCs, Rewards)
- Prism (DTMCs, CTMCs, MDPs, *GUI*)
- . . .

Thanks for your attention!