

Chapter 5

Probabilistic Systems

Course “Specification and Verification of Parallel Systems”

Juliane Päßler, Tobias John

Spring 2024



Section

Checking Linear Time Properties

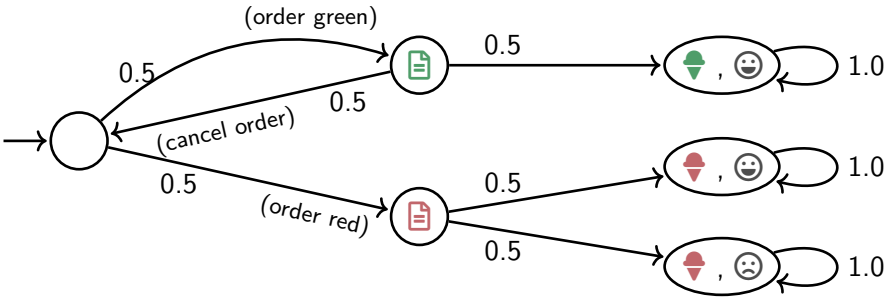
Spring 2024

Checking Linear Time Properties

- We aim to use LTL formulas to describe properties of Markov Chains
- The formulas are used to describe valid execution paths of the Markov Chain
- We distinguish between two problems:
  - Deciding if the formula is (almost always) satisfied (qualitative problem)
  - Computing the probability that the formula is satisfied (quantitative problem)
- Note: reachability properties are a special type of Linear Time Properties (represented by LTL formulas $\Diamond\varphi$ where φ does not contain temporal operators)
- Next slide:
  - Example of Markov Chain with LTL formulas describing properties

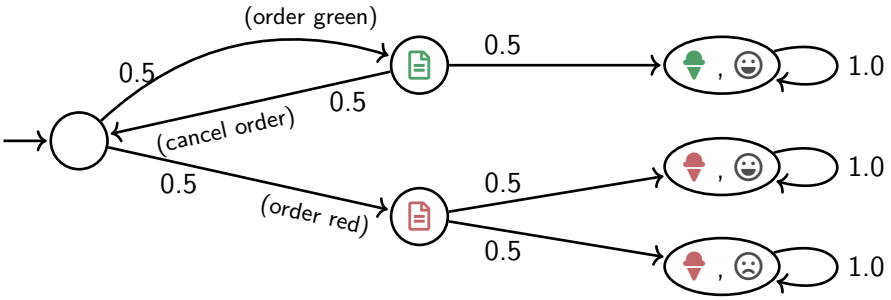


Example: Properties for Markov Chains





Example: Properties for Markov Chains



Property	Formula
	φ
customer will eventually have red ice cream	$\Diamond$🍦
customer will never be happy	$\Box\neg$😀
customer will eventually always be happy	$\Diamond\Box$😀
customer will eventually have ice cream	$\Diamond$(🍦 $\vee$ 🍦)

Qualitative LTL Model Checking

- Goal: Check if a finite Markov chain models an LTL formula (formula is (almost always) satisfied)
- Problem: Markov chains can include paths that do not model the formula but the probability of taking the paths is 0
  - Example: Consider the last formula on Slide 5–4; the infinite loop from the initial state to the first state on the top and back does not satisfy the formula, but the probability of taking this path is 0

$\rightarrow$ Only checking existence of a path that is a counter example is not enough



Qualitative LTL Model Checking

- Goal: Check if a finite Markov chain models an LTL formula (formula is (almost always) satisfied)
- Problem: Markov chains can include paths that do not model the formula but the probability of taking the paths is 0
  - Example: Consider the last formula on Slide 5–4; the infinite loop from the initial state to the first state on the top and back does not satisfy the formula, but the probability of taking this path is 0

  $\rightarrow$ Only checking existence of a path that is a counter example is not enough
- Solution: We check if probability of satisfying the formula is 1

Definition

Given a finite Markov chain $\mathcal{M}$ with initial state s_0 and an LTL formula φ, the relation $\mathcal{M} \models \varphi$ holds, if $\Pr^{\mathcal{M}}(s_0 \models \varphi) = 1$
The problem of deciding if $\mathcal{M} \models \varphi$ holds is the *qualitative problem*.



Quantitative LTL Model Checking

- Goal: Compute the probability of taking a path in a Markov Chain for which an LTL formula holds.

Definition

Given a finite Markov chain $\mathcal{M}$ with initial state s_0 and an LTL formula φ, the *quantitative problem* is calculating

$$\Pr^{\mathcal{M}}(\varphi) = \Pr^{\mathcal{M}}(s_0 \models \varphi).$$



Quantitative LTL Model Checking

- We are only going to discuss model checking of *quantitative* properties.
- The procedure is similar to LTL model checking of transition systems.
- This can also be used for *qualitative* properties.
- However, there exist more tailored algorithms for qualitative properties.



Repetition: LTL Model Checking

$$\mathrm{TS} \stackrel{?}{\models} \varphi$$

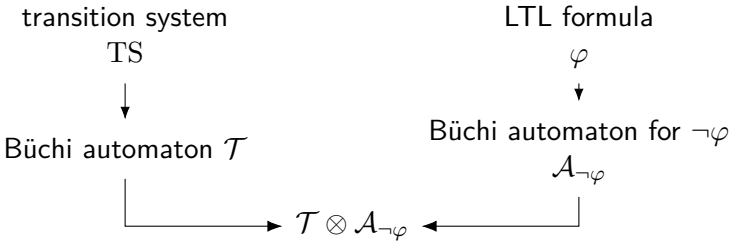
transition system
TS

LTL formula
φ

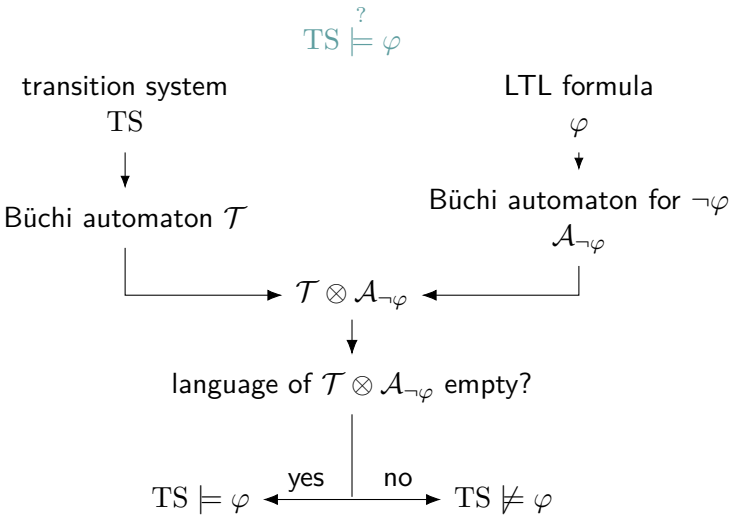


Repetition: LTL Model Checking

$$\mathrm{TS} \overset{?}{\models} \varphi$$



Repetition: LTL Model Checking





Quantitative LTL Model Checking for Markov Chains

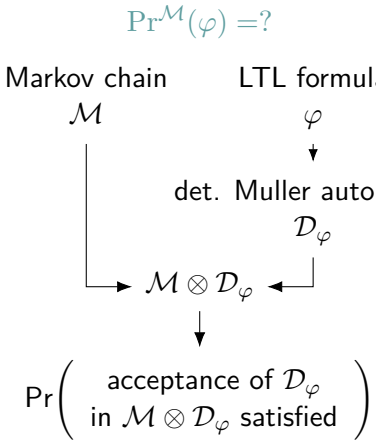
$$\Pr^{\mathcal{M}}(\varphi) = ?$$

Markov chain $\mathcal{M}$ LTL formula φ

- The procedure has a similar structure as LTL model checking of TS (see slide 5–8).



Quantitative LTL Model Checking for Markov Chains



- The procedure has a similar structure as LTL model checking of TS (see slide 5–8).
- The formula is transformed into a *deterministic* Muller automaton $\mathcal{D}_\varphi$.
- The product $\mathcal{M} \otimes \mathcal{D}_\varphi$ is a Markov Chain (because the automaton $\mathcal{D}_\varphi$ is deterministic).
- The acceptance check is similar to the emptiness check in Büchi automata.



Deterministic Automata on Infinite Words

- We need a *deterministic* automaton accepting infinite words for our construction
  - Only then we can build the product with the Markov chain later
- Deterministic Büchi automata are less expressive than nondeterministic Büchi automata
  - $\rightarrow$ Deterministic Büchi automata are not expresive enough for our purposes because of their acceptance condition
  - $\rightarrow$ We need an automaton with another acceptance condition: Muller automaton



Muller Automata: Intuition

- Like Büchi automata, Muller automata are a special kind of automata on infinite words
- The only difference between Büchi automata and Muller automata lies in their acceptance condition
- Acceptance condition in Muller automata:
  - Acceptance condition is a set of sets of states (in Büchi automata it is a set of states)
  - To decide acceptance: Consider the states that occur *infinitely* often in a path
  - A path is accepting, if this set of states is contained in the acceptance condition
- Deterministic Muller automata are as expressive as nondeterministic Büchi automata, i.e. characterize the ω-regular languages



Deterministic Muller Automata

Definition

A *deterministic Muller automaton* is a tuple $\mathcal{D} = (Q, q_0, \Sigma, \mathrm{Acc}, \delta)$ where

- Q is a finite set of states
- $q_0 \in Q$ is the initial state
- Σ is a finite alphabet
- $\mathrm{Acc} \subseteq \mathcal{P}(Q)$ is the acceptance condition
- $\delta : Q \times \Sigma \to Q$ is the transition function

A path π of states from Q is accepted if (and only if) $\inf(\pi) \in \mathrm{Acc}$, where $\inf(\pi)$ is the set of states that occur infinitely often in π.



Construct Automaton for LTL Formula φ

- Construction via nondeterministic Büchi automata
  1. Construct nondeterministic generalized Büchi automaton for φ
  2. Construct nondeterministic Büchi automaton from nondeterministic generalized Büchi automaton
  3. Construct deterministic Muller automaton from the nondeterminsitic Büchi automaton (determinization)
- Steps 1 and 2 can be done using the constructions discussed earlier in this lecture
- Step 3 can be done using *Safra's construction* (we don't discuss any details in this lecture)
- Remember: Nondet. Büchi automaton is exponentially larger than φ
- Complexity: Step 3 involves exponential growth of state space

  $\rightarrow$ The deterministic automaton is double-exponentially larger than the formula φ, which cannot be avoided



Product Construction of MC and Muller Automaton

- MC and Muller automaton execute synchronously: Label on transition of Muller automaton must match the label of the state in the MC
- States in product MC get labeled by states from Muller automaton $\rightarrow$ We can later use the acceptance condition of the automaton for product MC

Definition

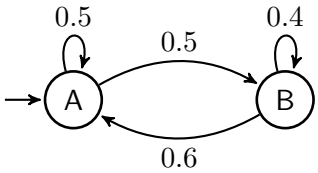
Let $\mathcal{M} = (S, P, s_0, AP, L)$ be a Markov chain and $\mathcal{D} = (Q, q_0, AP, \mathrm{Acc}, \delta)$ a deterministic Muller automaton. The product $\mathcal{M} \otimes \mathcal{D}$ is the following Markov chain:

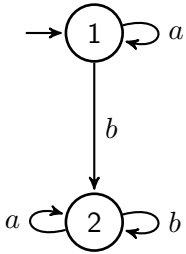
$$(S \times Q, P', s_0', Q, L') \text{ with}$$

- $P'((s,q),(s',q')) = \begin{cases} P(s,s') & q' = \delta(q, L(s)) \\ 0 & \text{else} \end{cases}$
- $s_0' = (s_0, q_0)$
- $L'(s,q) = q$

Example: Product Construction of $\mathcal{M}$ and $\mathcal{D}$

$L(A) = a$
$L(B) = b$

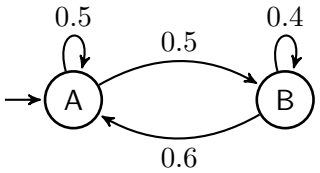


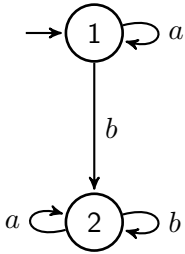


$\text{Acc} = \{\{1\}, \{1, 2\}\}$

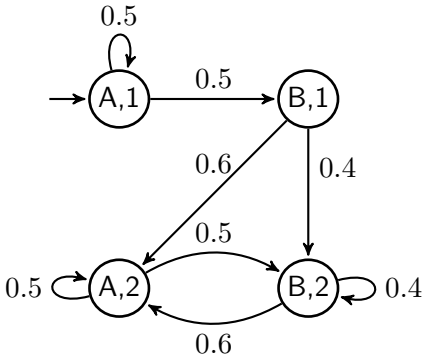
Example: Product Construction of $\mathcal{M}$ and $\mathcal{D}$

$L(A) = a$
$L(B) = b$



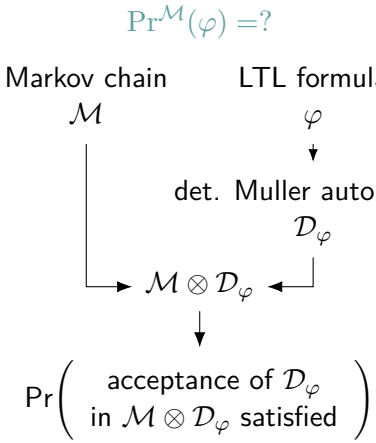


$\text{Acc} = \{\{1\}, \{1,2\}\}$



$L'((A,1)) = 1$
$L'((A,2)) = 2$
$L'((B,1)) = 1$
$L'((B,2)) = 2$

Quantitative LTL Model Checking for Markov Chains

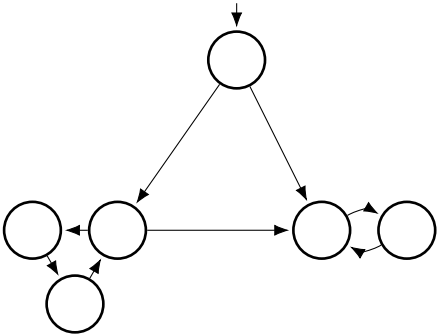


- The procedure has a similar structure as LTL model checking of TS (see slide 5–8).
- The formula is transformed into a *deterministic* Muller automaton $\mathcal{D}_\varphi$.
- The product $\mathcal{M} \otimes \mathcal{D}_\varphi$ is a Markov Chain (because the automaton $\mathcal{D}_\varphi$ is deterministic).
- The acceptance check is similar to the emptiness check in Büchi automata.

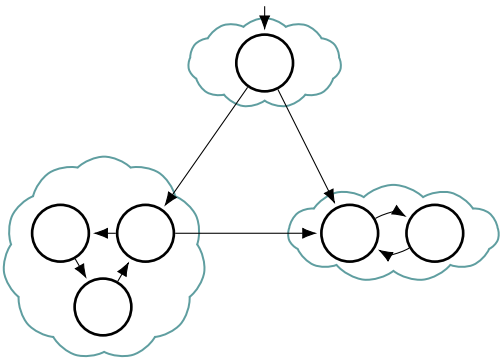


Acceptance Check (Idea)

- Goal: Analysis of acceptance in product using acceptance in $\mathcal{D}_\varphi$



Acceptance Check (Idea)

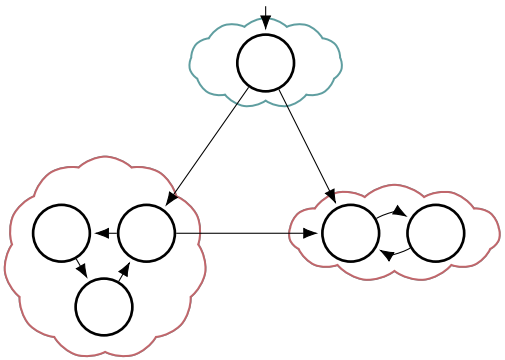


- Goal: Analysis of acceptance in product using acceptance in $\mathcal{D}_\varphi$

(1) Partition in strongly connected components (SCCs)



Acceptance Check (Idea)



- Goal: Analysis of acceptance in product using acceptance in $\mathcal{D}_\varphi$

(1) Partition in strongly connected components (SCCs)

(2) Find bottom SCCs (BSCCs) (SCCs that cannot be left once entered)

(3) Identify, which BSCCs are a set of states contained in the acceptance condition of $\mathcal{D}_\varphi$

(4) Calculate probability of reaching the accepting BSCCs (a reachability property, as discussed in the last lecture)



Repetition: SCCs

Definition (SCC)

A subset $S' \subseteq S$ in a directed graph, such as a Markov Chain, is **strongly connected** if there is a path between any pair of nodes in S', passing only through nodes in S'.
A **strongly-connected component** (SCC) is a *maximal* set of such nodes, i.e. it is not possible to add any node to that set and still maintain strong connectivity.



Accepting End Components

Definition (BSCC)

A **bottom SCC (BSSC)** is an SCC S from which no state outside S is reachable, i.e. for all $s \in S$:

$$\sum_{t \in S} \Pr(s, t) = 1.$$

Definition

An **accepting end component** w.r.t. a Muller acceptance condition Acc is a BSCC S such that $S \in Acc$.

- BSCCs can be determined using common graph algorithms (e.g. breath-first search)
- Identifying accepting end components from BSCCs is trivial



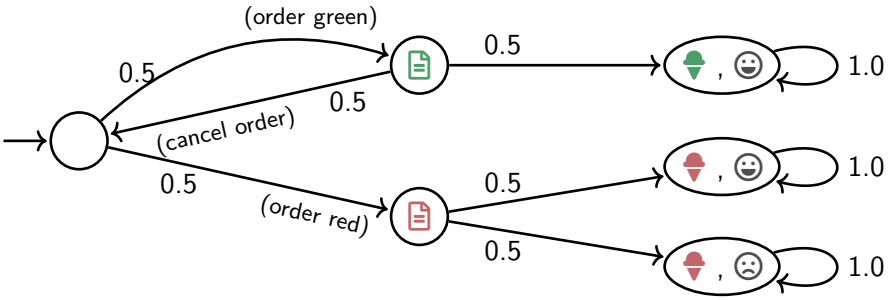
Complete Algorithm for Quantitive LTL MC

Goal: Calculate $\Pr^{\mathcal{M}}(\varphi)$

1. Construct deterministic Muller automaton $\mathcal{D}_\varphi$ with acceptance condition Acc for formula φ
2. Build product Markov chain $\mathcal{M}_\otimes = \mathcal{M} \otimes \mathcal{D}_\varphi$
3. Compute accepting end components of $\mathcal{M}_\otimes$ w.r.t. Acc
4. Compute the union of all states in accepting end components (let this set be B)
5. Compute $\Pr^{\mathcal{M}_\otimes}(\Diamond B)$, i.e. analyze reachability property
6. $\Pr^{\mathcal{M}}(\varphi) = \Pr^{\mathcal{M}_\otimes}(\Diamond B)$

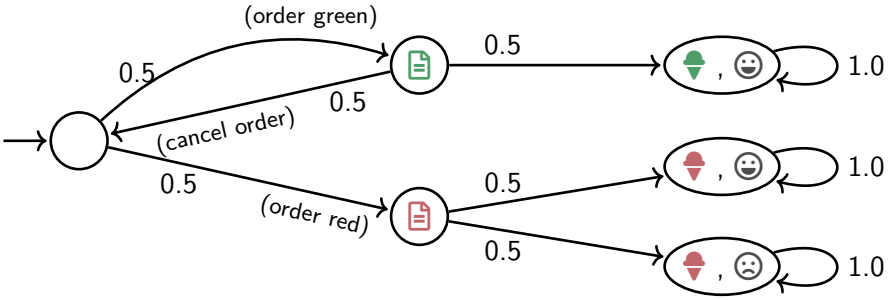


Example: Properties for Markov Chains



Property	Formula
	φ
customer will eventually have red ice cream	$\Diamond$🍦
customer will never be happy	$\Box\neg$😀
customer will eventually always be happy	$\Diamond\Box$😀
customer will eventually have ice cream	$\Diamond$(🍦 $\vee$ 🍦)

Example: Properties for Markov Chains



Property	Formula φ	Qualitative MC $\Pr^{\mathcal{M}}(\varphi)$	Quantitative MC $\mathcal{M} \models \varphi$
customer will eventually have red ice cream	$\Diamond$🍦	0.66	no
customer will never be happy	$\Box\neg$😀	0.33	no
customer will eventually always be happy	$\Diamond\Box$😀	0.66	no
customer will eventually have ice cream	$\Diamond$(🍦 $\vee$ 🍦)	1.00	yes

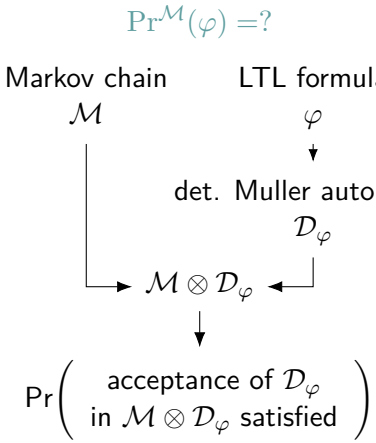
Complexity of Quantitive LTL MC

- Our algorithm:
  - Size of $\mathcal{D}_\varphi$: double exponential in length of φ
  - Size of $\mathcal{M}_\otimes$ linear in size of $\mathcal{M}$ and $\mathcal{D}_\varphi$
  - Identifying BSCCs: polynomial in size of $\mathcal{M}_\otimes$
  - Solving reachability analysis: polynomial in size of $\mathcal{M}_\otimes$
  - Overall: double-exponential in size of φ and polynomial in size of $\mathcal{M}$

- In general:
  - There exist advanced algorithms, such that the computation time is only exponential in the size of φ, which is optimal



Quantitative LTL Model Checking for Markov Chains



- The procedure has a similar structure as LTL model checking of TS (see slide 5–8).
- The formula is transformed into a *deterministic* Muller automaton $\mathcal{D}_\varphi$.
- The product $\mathcal{M} \otimes \mathcal{D}_\varphi$ is a Markov Chain (because the automaton $\mathcal{D}_\varphi$ is deterministic).
- The acceptance check is similar to the emptiness check in Büchi automata.



Summary: Markov Chains

- Markov Chains are a model of probabilistic processes
- Linear Time Properties
  - qualitative
  - quantitative
- Model checking for quantitative properties
  - Similar to LTL model checking for transition systems
  - Can be reduced to reachability analysis
- Next lecture:
  - Probabilistic processes with nondeterminism (MDPs)

