Model Checking

Complexity of LTL Model Checking

[Baier & Katoen, Chapter 5.2.1+5.2.2]

Joost-Pieter Katoen and Tim Quatmann

Software Modeling and Verification Group

RWTH Aachen, SoSe 2022

Topic

What is the theoretical complexity of LTL model checking?

Main Result:

[Sistla and Clarke, 1985]

The LTL model-checking problem is PSPACE-complete.

LTL Decision Problems

The LTL Model-Checking Problem ($\texttt{LTL}$)

Given a finite transition system TS and LTL-formula φ, is $TS \models \varphi$?

The LTL Satisfiability Problem ($\texttt{LTL}_{\texttt{SAT}}$)

Given LTL-formula φ, does there exist a transition system TS such that $TS \models \varphi$?

The LTL Validity Problem ($\texttt{LTL}_{\texttt{valid}}$)

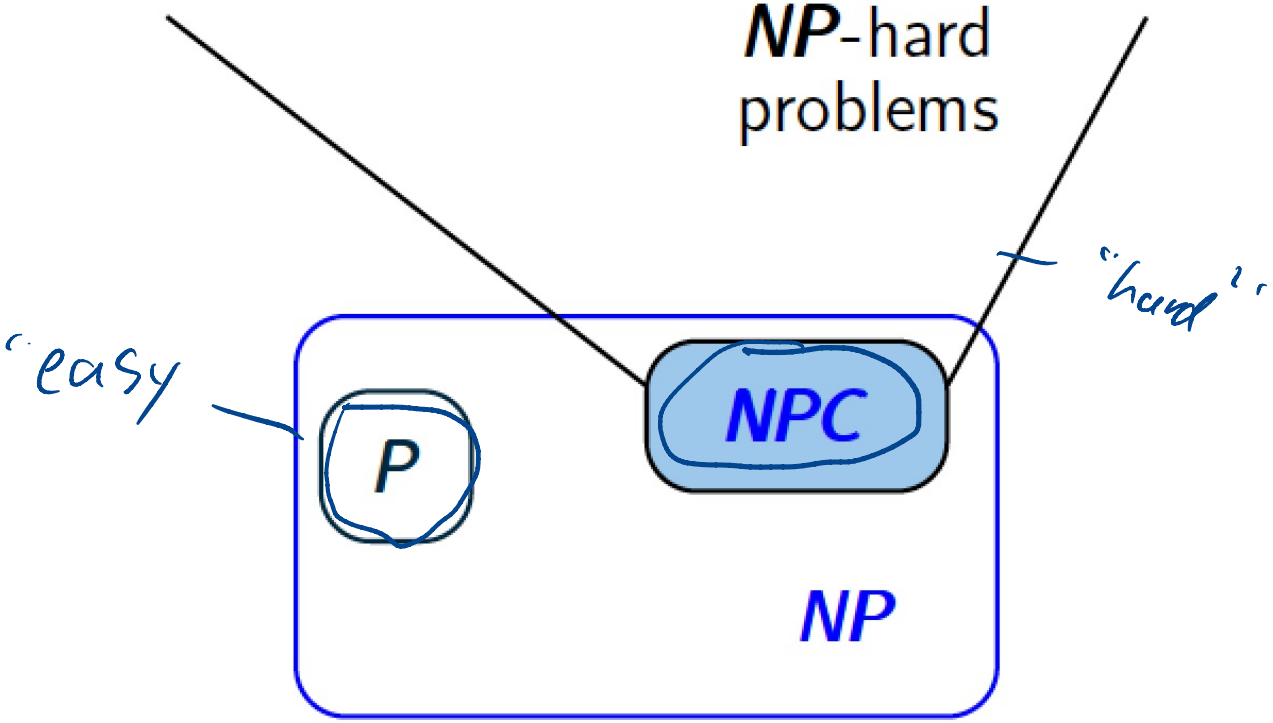
Given LTL-formula φ, does $TS \models \varphi$ for all transition systems TS?

The validity problem for φ is equivalent to the satisfiability problem for $\neg\varphi$

Overview

Overview

Complexity Classes P and NP



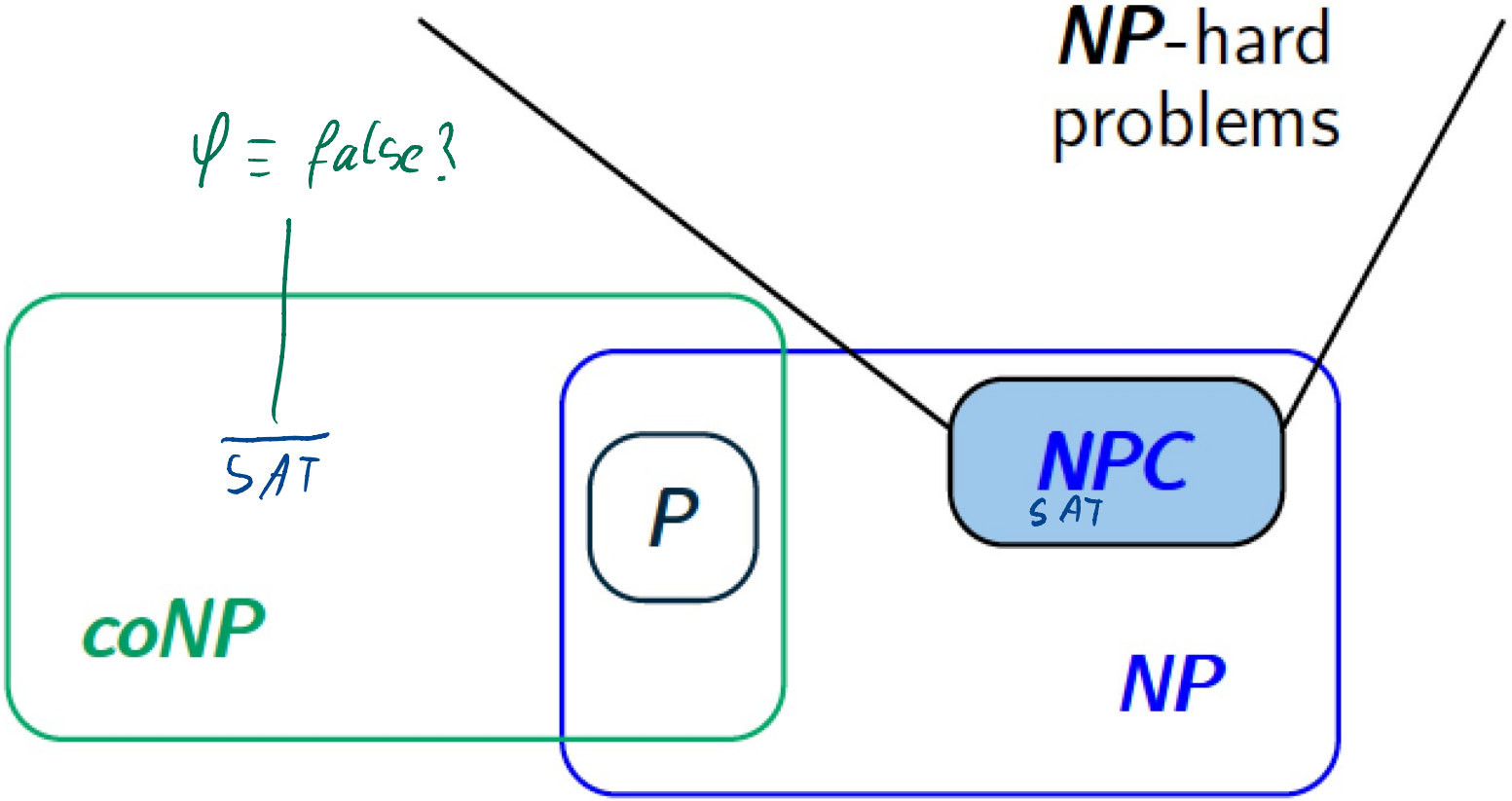
NPC = class of **NP**-complete problems

(1) $L \in NP$

(2) L is **NP**-hard, i.e., $K \leq_{poly} L$ for all $K \in NP$

polynomial reduction

Complexity Class co-NP

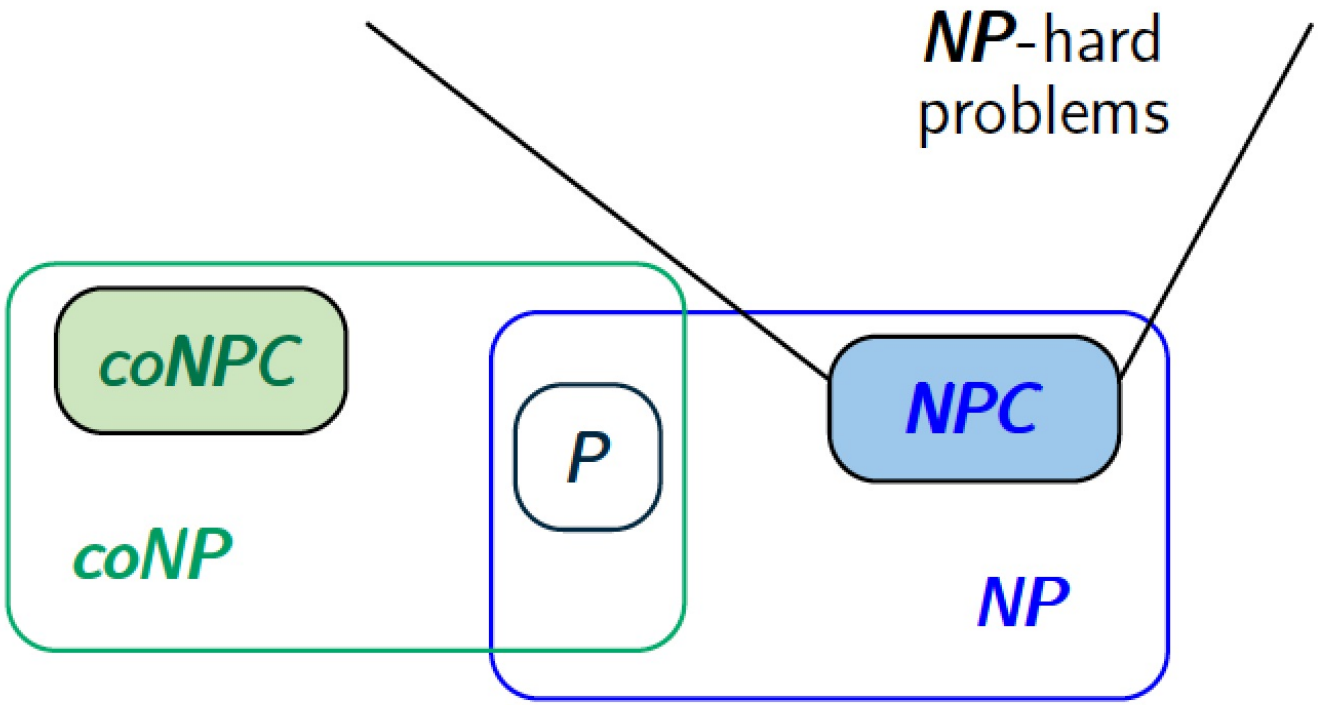


$$coNP = \{ \overline{L} : L \in NP \}$$

↑

complement of L

Co-NP Completeness

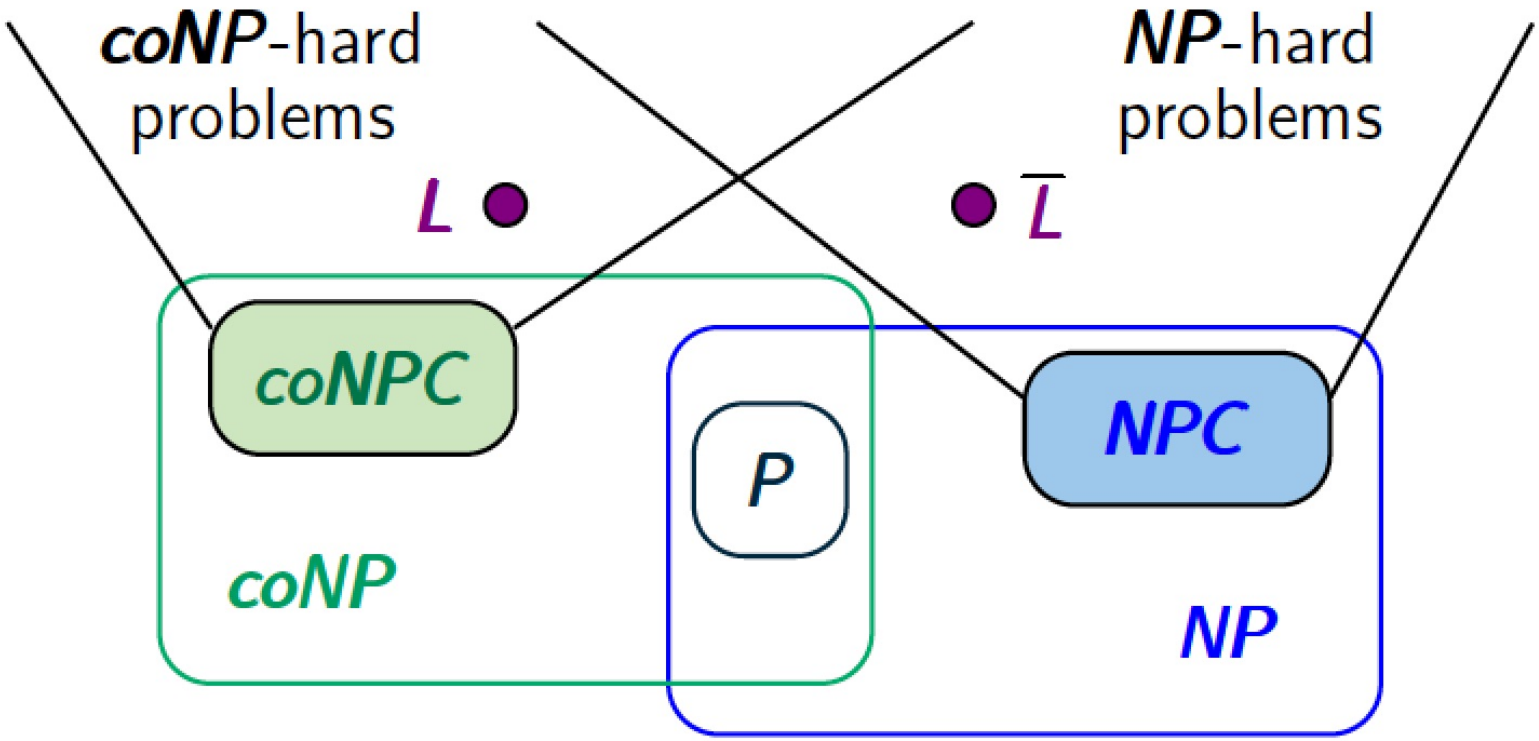


coNPC = class of **coNP**-complete problems

(1) $L \in$ **coNP**

(2) L is **coNP**-hard, i.e., $K \leq_{poly} L$ for all $K \in$ **coNP**

Complexity Classes P, NP, and co-NP



coNPC = class of **coNP**-complete problems

> L is **coNP**-hard iff $\overline{L}$ is **NP**-hard

The Hamilton Path Problem

A Hamilton path of a (directed and finite) graph $G = (V, E)$ is a path $\pi = v_1 \dots v_n$ that visits each vertex of G exactly once, i.e.,

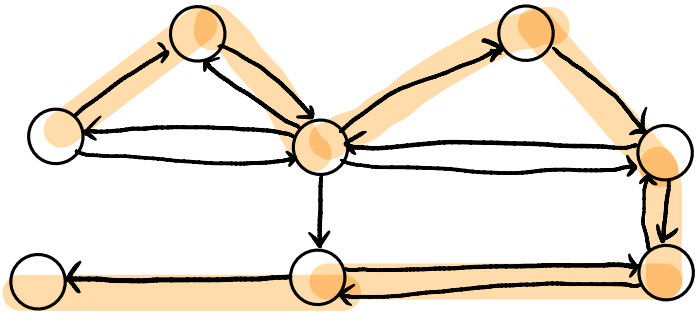
$$\forall\, v \in V \,.\, \exists\, i \leq n \,.\, \big(v_i = v \text{ and } \forall\, j \neq i \,.\, v_j \neq v\big).$$

The Hamiltopn Path Problem (HAM)

Given a graph G, does G have a Hamilton path?

HAM is NP-complete.

Example:



Co-NP Hardness

The LTL model-checking problem (LTL) is co-NP-hard.

Proof.

- We show that $\overline{\texttt{LTL}}$ is NP-hard.
- To this end, we provide a polynomial reduction: $\texttt{HAM} \leq_{\text{poly}} \overline{\texttt{LTL}}$.

□

The complement $\overline{\texttt{LTL}}$ of the problem LTL asks:

Given finite TS and LTL-formula φ, is $TS \not\models \varphi$?

$\forall \sigma \in Trace(TS) : \sigma \not\models \varphi$

Polynomial Reduction for $\texttt{HAM} \leq_{\text{poly}} \overline{\texttt{LTL}}$

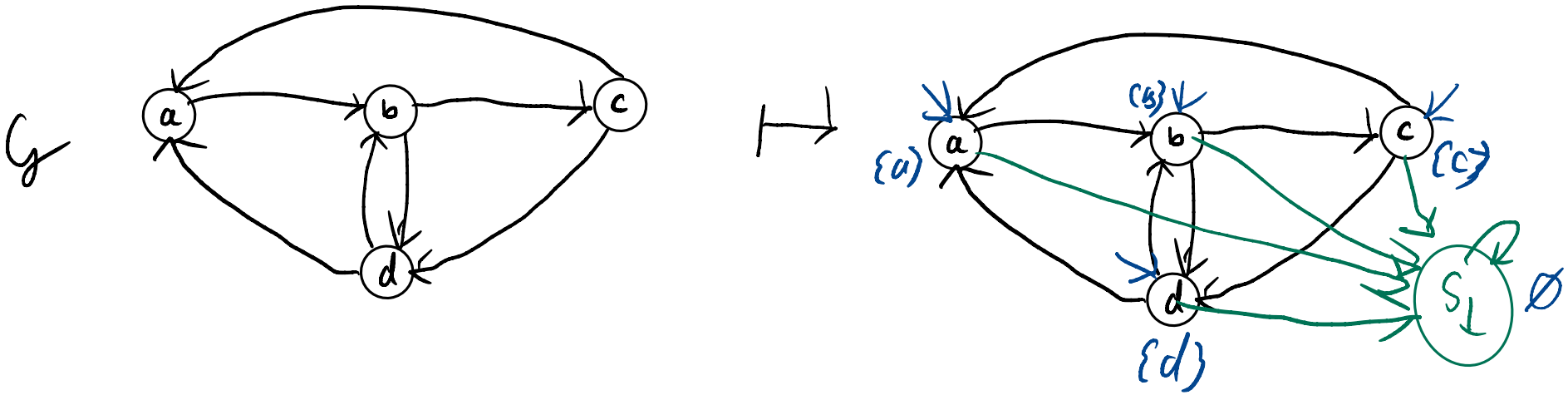
Transform input $G = (V, E)$ for HAM to input (TS, φ) for LTL such that

→ G has a Hamilton path iff $TS \not\models \varphi$

→ the transform. can be computed in Polynomial time

- $TS := (V \uplus \{s_\perp\},\ \{\alpha\},\ \rightarrow,\ V,\ V,\ L)$, where (ACT, Init., AP)
  - $L(s_\perp) := \emptyset, \quad \forall v \in V.\ L(v) := v$, and
  - $\rightarrow := \{(v, \alpha, v') \mid (v, v') \in E\} \cup \{(s, \alpha, \{s_\perp\}) \mid s \in V \uplus \{s_\perp\}\}$

Example:



Polynomial Reduction for $\texttt{HAM} \leq_{\text{poly}} \overline{\texttt{LTL}}$

Transform input $G = (V, E)$ for HAM to input (TS, φ) for LTL such that

G has a Hamilton path iff $TS \not\models \varphi$

$\exists \sigma \in Traces(TS)$
$\sigma \not\models \varphi$

- $TS := (V \uplus \{s_\bot\}, \{\alpha\}, \rightarrow, V, V, L)$, where
  - $L(s_\bot) := \varnothing, \quad \forall v \in V . L(v) := v$, and
  - $\rightarrow := \{(v, \alpha, v') \mid (v, v') \in E\} \cup \{(s, \alpha, \{s_\bot\}) \mid s \in V \uplus \{s_\bot\}\}$
- $\varphi = \neg\Big(\bigwedge_{v \in V} \Diamond v \wedge \Box\big(v \Rightarrow \bigcirc \Box \neg v\big)\Big)$

v is reached — whenever v is reached, then we don't see v afterwards

there is a Ham path

$|TS| + |\varphi|$ is linear in the size of the graph $|G|$

Overview

The Complexity Class PSPACE

PSPACE is the set of decision problems that can be solved by a deterministic, polynomially-bounded space algorithm.

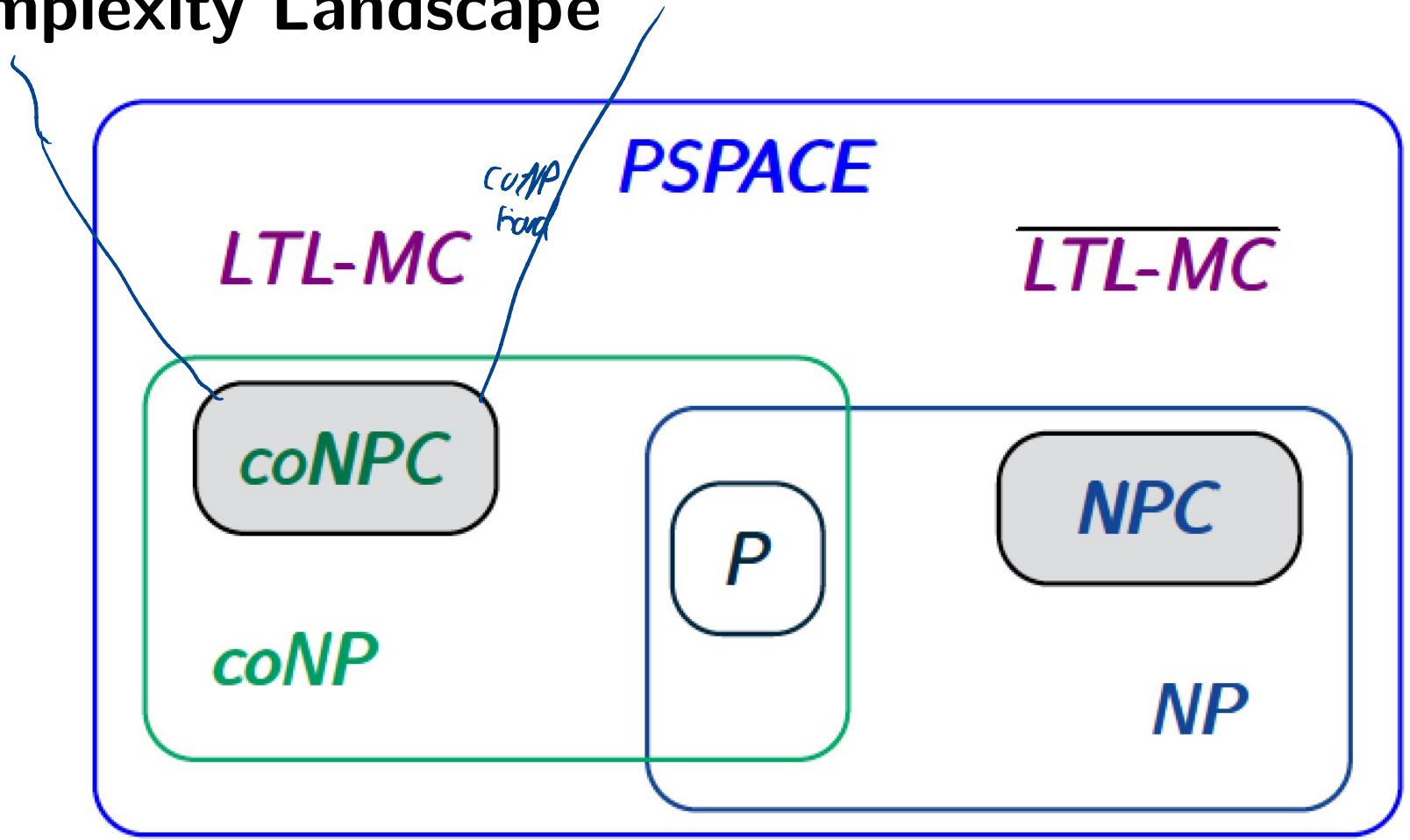
Known facts:

- NP $\subseteq$ PSPACE
- PSPACE = co-PSPACE (this holds for any deterministic complexity class)
- PSPACE = NPSPACE (Savitch theorem)

Due to Savitch's theorem, to show L is in PSPACE, it suffices to provide a nondeterministic, polynomially-bounded space algorithm for $\overline{L}$.

NPSPACE = PSPACE = co-PSPACE = co-NPSPACE

Complexity Landscape



PSPACE is the set of decision problems that can be solved by a deterministic, polynomially-bounded space algorithm

Complexity Landscape

decision problem L is *PSPACE*-complete iff

(1) $L \in \mathit{PSPACE}$

(2) L is *PSPACE*-hard $\longleftarrow$ $K \leq_{\mathit{poly}} L$ for all $K \in \mathit{PSPACE}$

as $\mathit{PSPACE} = \mathit{coPSPACE} = \mathit{NPSPACE}$:

L is *PSPACE*-hard $\iff$ $\overline{L}$ is *PSPACE*-hard

$L \in \mathit{PSPACE}$ $\iff$ $\overline{L} \in \mathit{NPSPACE}$

Existential LTL Model-Checking

The Existential LTL Model-Checking Problem ($\exists\mathtt{LTL}$)

Given a finite transition system TS and LTL-formula φ, is there some path π in TS such that $\pi \vDash \varphi$?

- Recall: $\overline{\mathtt{LTL}}$ asks if $TS \nvDash \varphi$.
- The problems $\exists\mathtt{LTL}$ and $\overline{\mathtt{LTL}}$ are equally hard:

$$
\begin{aligned}
(TS, \varphi) \in \exists\mathtt{LTL} \text{ iff } & \exists \pi \in \mathit{Paths}(TS) \,.\, \pi \vDash \varphi \\
\text{iff } & \exists \pi \in \mathit{Paths}(TS) \,.\, \pi \nvDash \neg\varphi \\
\text{iff not } & \big(\forall \pi \in \mathit{Paths}(TS) \,.\, \pi \vDash \neg\varphi\big) \\
\text{iff } & TS \nvDash \neg\varphi \\
\text{iff } & (TS, \neg\varphi) \in \overline{\mathtt{LTL}}
\end{aligned}
$$

LTL Model Checking is PSPACE-complete

[Sistla and Clarke, 1985]

The LTL model-checking problem is PSPACE-complete.

Proof.

We prove that the existential LTL model-checking problem ($\exists\texttt{LTL}$) is

(a) in NPSPACE (and thus in PSPACE), and

(b) PSPACE-hard.

It follows that $\exists\texttt{LTL}$ (and thus also $\overline{\texttt{LTL}}$) is PSPACE-complete. The claim follows from co-PSPACE = PSPACE. □

$\exists$LTL is in NPSPACE—Proof Idea

The Existential LTL Model-Checking Problem ($\exists$LTL)

Given finite TS and LTL-formula φ, $\exists \pi \in \mathit{Paths}(TS) \,.\, \pi \vDash \varphi$?

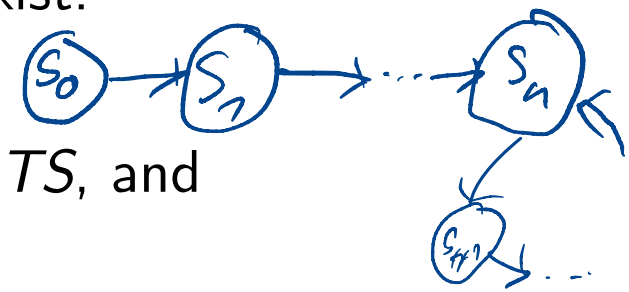
Goal:
Find a criterion for the existence of a path π in TS with $\pi \vDash \varphi$ that can be checked non-deterministically in poly-space

Idea:
Use the GNBA $\mathfrak{G}_\varphi$ for φ

Problem: The size of $\mathfrak{G}_\varphi$ can be exponential in $|\varphi|$

Criterion For ∃LTL Properties

There exists a path π in TS with $\pi \vDash \varphi$ iff there exist:



- an initial path fragment $\underbrace{s_0 s_1 \ldots s_n \ldots s_{n+m}}_{\pi}$ in TS, and
- a run $B_0\, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in GNBA $\mathfrak{G}_\varphi$ for $\mathit{trace}(\pi)$

such that:

1. $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$
2. if $\psi_1 \,\mathsf{U}\, \psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$ then $\psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$
3. $n \leq |S| \cdot 2^{|cl(\varphi)|}$ and $m \leq |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

$\approx |TS \otimes G_\varphi|$

NPSPACE Algorithm for the $\exists$LTL Problem (1)

1. Guess nondeterministically a path:

$$\pi \;=\; u_0 u_1 \ldots u_{n-1} \big(u_n \ldots u_{n+m}\big)^{\omega} \text{ in } TS \otimes \mathfrak{G}_{\varphi}$$

   where GNBA $\mathfrak{G}_{\varphi}$ is obtained as explained in the previous lecture

2. Check whether the guessed path is accepted by $\mathfrak{G}_{\varphi}$

PSPACE Algorithm for the $\exists$LTL Problem (2)

- Guess natural numbers $n, m \leq |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$ such that $m > 0$
  ↳ stored in binary ⇝ space $\in O(\log(|S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|))$ (polynomial)
- Guess initial path fragment $\pi = s_0 s_1 \ldots s_n \ldots s_{n+m}$ in TS
  $|\pi|$ has exp. length
- Guess subsets $B_0, \ldots, B_n, \ldots, B_{n+m+1}$ of $cl(\varphi)$

- Check whether the following three conditions hold:
  1. $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$
  2. $B_0 \ldots B_n \ldots B_{n+m+1}$ is an initial run for $trace(\pi\, s_n)$ in $\mathfrak{G}_\varphi$
  3. $\{\psi_2 \mid \psi_1 \,\mathsf{U}\, \psi_2 \in \bigcup_{n<i\leq n+m} B_i\} \subseteq \bigcup_{n<i\leq n+m} B_i$

  If so, return "yes", otherwise return "no".

Guessing and Checking can be implemented in an interleaved way so that the complete path does not need to be stored.

LTL Model Checking is PSPACE-complete

[Sistla and Clarke, 1985]

The LTL model-checking problem is PSPACE-complete.

Proof.

We prove that the existential LTL model-checking problem ($\exists\texttt{LTL}$) is

(a) in NPSPACE (and thus in PSPACE), and

(b) PSPACE-hard.

It follows that $\exists\texttt{LTL}$ (and thus also $\overline{\texttt{LTL}}$) is PSPACE-complete. The claim follows from co-PSPACE = PSPACE. □

We now show PSPACE-hardness of $\exists\texttt{LTL}$ by a polynomial reduction from a PSPACE-hard problem

Deterministic Turing machines

A deterministic Turing machine (DTM) is a tuple $M = (Q, \Sigma, \delta, q_I, q_F)$ with

- finite state space Q, initial state $q_I \in Q$, accept state $q_F \subseteq Q$
- tape alphabet Σ, and
- transition function $\delta\colon Q \times \Sigma \to Q \times \Sigma \times \{L, N, R\}$ such that $\delta(q_F, A) = (q_F, A, N)$ for all $A \in \Sigma$.

Left neutral Right

- $\delta(q, A) = (p, B, L)$ means: at state q with current tape cell content A, write B to the tape, go to state p, and move the cursor to the left
- A configuration is a triple $c = (q, i, w) \in Q \times \mathbb{N} \times \Sigma^*$ with $i \leq |w|$
- The (unique) run on input word $w \in \Sigma^*$ is the infinite sequence of configurations starting with $(q_I, 0, w)$ as predetermined by the transition function δ
- M accepts input word $w \in \Sigma^*$ iff the run on w reaches q_F

A PSPACE-hard Problem

$a_1 x^{n_1} + a_2 x^{n_2} + \dots$

The polynomially bounded DTM-Acceptence Problem (`polyDTM`)

Given a DTM M, input word $w \in \Sigma^*$, and a polynomial $P\colon \mathbb{N} \to \mathbb{N}$, does M accept w using at most $P(|w|)$ tape cells?

`polyDTM` is PSPACE-hard.

To show PSPACE-hardness of $\exists$`LTL` we now show

$$\texttt{polyDTM} \leq_{\text{poly}} \exists\texttt{LTL}$$

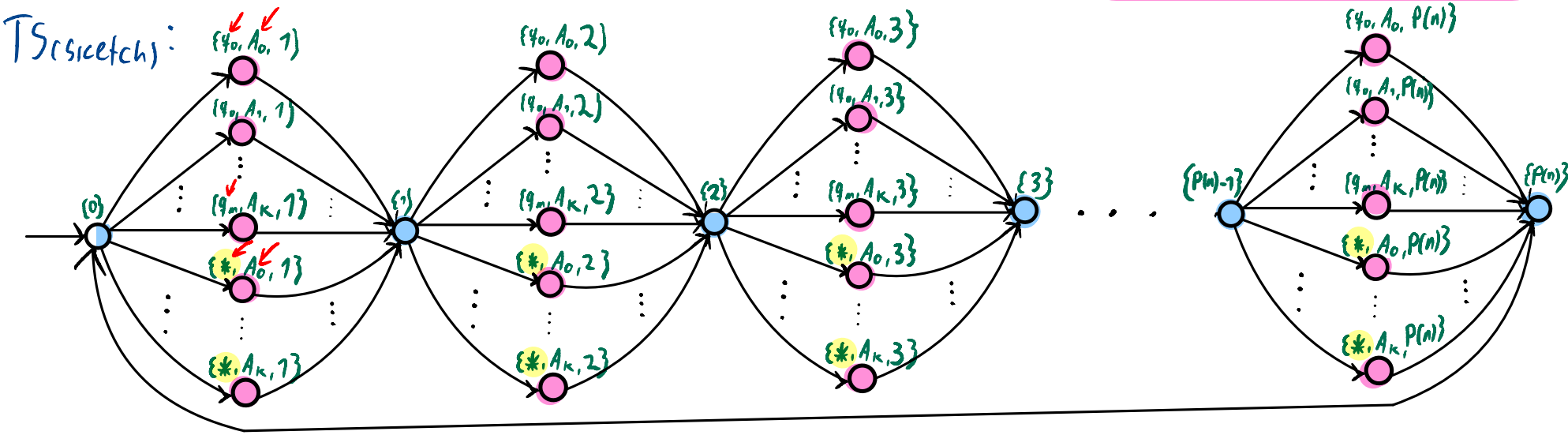
by providing a transformation from an input (M, w, P) for `polyDTM` to an input (TS, φ) for $\exists$`LTL`.

Transformation: $(M, w, P) \mapsto (TS, \varphi)$

$n := |w|$

states of TS are given by $S = \{0, 1, \dots, P(n)\} \cup Q \cup \{*\} \times \Sigma \times \{0, 1, \dots, P(n)\}$

TS (sketch):



where $Q = \{q_0, \dots q_m\}$, $\Sigma = \{A_1, \dots, A_K\}$, $|w| = n$

A path fragment of the following form encodes a configuration:

$$0\ (*, A_{j_1}, 1)\ 1\ (*, A_{j_2}, 2)\ 2 \cdots i\text{-}1\ \underbrace{(q, A_{j_i}, i)}_{\text{only position without } *}\ i \cdots P(n)\text{-}1\ (*, A_{j_{P(n)}}, P(n))\ P(n)$$

$\rightsquigarrow$ Tape content: $A_{j_1} A_{j_2} \cdots A_{j_i} \cdots A_{P(n)}$

$\rightsquigarrow$ Current state: q $\rightsquigarrow$ Cursor position: i

Transformation: $(M, w, P) \mapsto (TS, \varphi)$

φ asserts that a trace of TS encodes the valid run of M on w:

$$\varphi = \varphi_{\text{start}} \wedge \varphi_{\text{conf}} \wedge \varphi_{\delta} \wedge \varphi_{\text{acc}}$$

- φ_{start} encodes the start configuration—for $w = A_1 \dots A_n$ we get $\varphi_{\text{start}} = \bigcirc q_I \wedge \bigwedge_{i=1}^{n} \bigcirc^{2i-1} A_i \wedge \bigwedge_{i=n+1} \bigcirc^{2i-1} \sqcup$
- φ_{conf} ensures a valid encoding of a sequence of configurations
- φ_{δ} encodes the transition function of M
- $\varphi_{\text{acc}} = \Diamond q_F$ encodes that the run is accepting

We get that $(M, w, P) \in \texttt{polyDTM}$ iff $\sigma \vDash \varphi$ for some trace σ of TS.

Overview

LTL Satisfiability is PSPACE Complete

The LTL Satisfiability Problem ($\mathtt{LTL}_{\mathtt{SAT}}$)

Given LTL-formula φ, does there exist *finite* TS such that $TS \vDash \varphi$?

The LTL satisfiability problem is PSPACE-complete.

Proof (sketch).

- The satisfiability of LTL-formula φ amounts to checking whether $\mathit{Words}(\varphi) \neq \varnothing$ or—equivalently—$\mathfrak{L}_\omega(\mathfrak{G}_\varphi) \neq \varnothing$.
- $\mathtt{LTL}_{\mathtt{SAT}}$ is in NPSPACE = PSPACE: guess and check an accepting run of the GNBA $\mathfrak{G}_\varphi$ as in our proof for $\exists\mathtt{LTL}$.
- $\mathtt{LTL}_{\mathtt{SAT}}$ is PSPACE-hard: show that $\exists\mathtt{LTL} \leq_{\mathrm{poly}} \mathtt{LTL}_{\mathtt{SAT}}$ $(TS, \varphi) \mapsto \psi$

Overview

Summary

- The LTL model-checking problem is PSPACE-complete
- The LTL satisfiability problem is PSPACE-complete
- The Hamiltonian path problem is polynomially reducible to the complement of the LTL model-checking problem.

Next Lecture

Monday May 16, 10:30