Model Checking

Lecture #9: Complexity of LTL Model Checking

[Baier & Katoen, Chapter 5.2.1+5.2.2]

Joost-Pieter Katoen

Software Modeling and Verification Group

Model Checking Course, RWTH Aachen, WiSe 2019/2020

Overview

Topic

What is the theoretical complexity of LTL model checking?

LTL Decision Problems

The LTL Model-Checking Problem

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

The LTL Satisfiability Problem

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

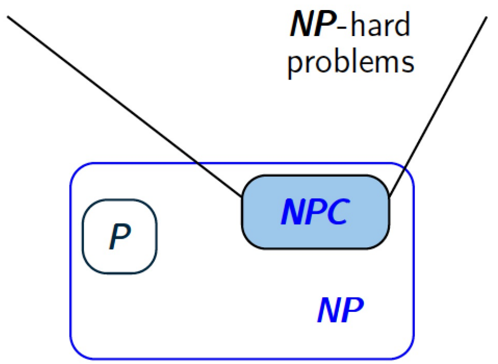
The LTL Validity Problem

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

Complexity Classes P and NP

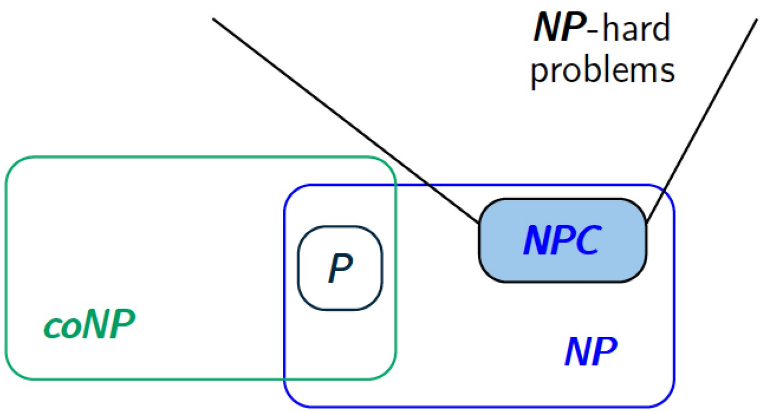


$$\boldsymbol{NPC} \;=\; \text{class of } \boldsymbol{NP}\text{-complete problems}$$

(1) $L \in NP$

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

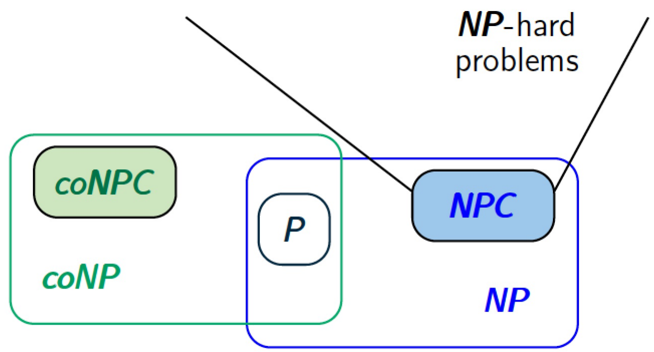
Complexity Class co-NP



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

complement of L

Co-NP Completeness

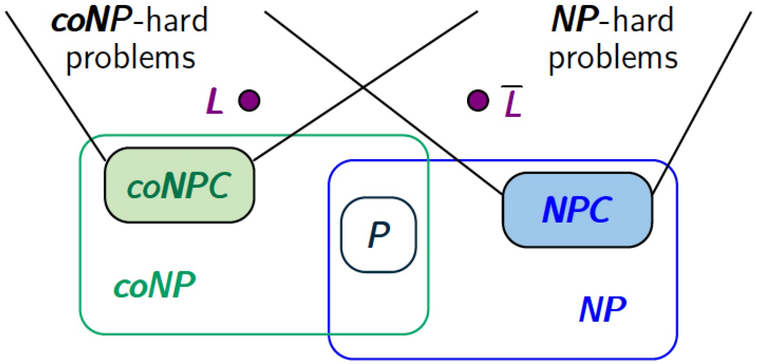


$$coNPC \;=\; \text{class of } coNP\text{-complete problems}$$

(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

Co-NP Hardness

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

Proof.

We show that the Hamiltonian path problem –which is NP-complete– for directed graphs is polynomially reducible to the complement of the LTL model-checking problem. The complement LTL model-checking problem is: given finite transition system TS and LTL-formula φ, is $TS \not\models \varphi$?

As the complement LTL model-checking problem is NP-complete, the result follows. □

The Complexity Class PSPACE

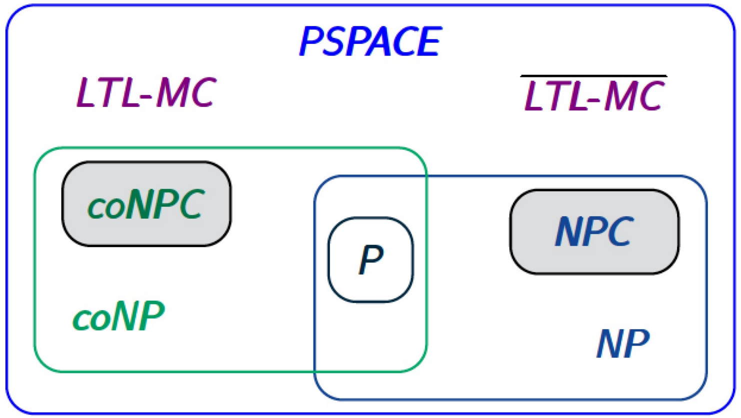
PSPACE is the set of decision problems that can be solved by a deterministic, polynomially-bounded space algorithm.

Known facts:

- NP ⊆ 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 $\overline{L}$.

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 $\longleftarrow$ $K \leq_{poly} L$ for all $K \in \mathit{PSPACE}$

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

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

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

Overview

LTL Model Checking is PSPACE Complete

The LTL model-checking problem is PSPACE-complete.

Proof.

We prove that the existential LTL model-checking problem is (a) in NPSPACE (and thus in PSPACE), and (b) PSPACE-hard.

The complement LTL model-checking problem is: given finite transition system *TS* and LTL-formula φ, is $TS \not\models \varphi$? That is, is there some path π in *TS* such that $\pi \not\models \varphi$, i.e., $\pi \models \neg\varphi$?

This is the existential LTL-model-checking problem for $\psi = \neg\varphi$. □

Existential LTL Model-Checking Problem

Given:

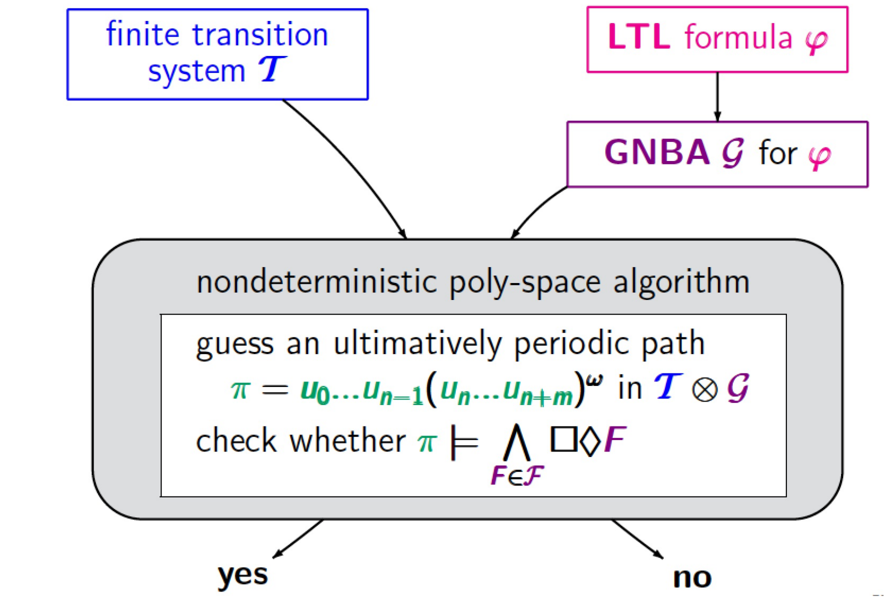
- finite transition system *TS*
- LTL formula φ

Q: is there some path π in *TS* such that $\pi \models \varphi$?

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

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

Proof Idea



Criterion For ∃LTL Properties

There exists a path π in TS with $\pi \models \varphi$ 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 $trace(\pi)$

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 $m \;\leq\; |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

NPSPACE Algorithm for ∃LTL MC Problem

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 $\mathfrak{G}_\varphi$

NPSPACE Algorithm for Existential LTL MC Problem

- Guess two natural number n and $m \leq k$ such that $m > 0$ and

$$k \;=\; |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$$

- Guess initial path fragment $\pi = s_0 s_1 \ldots s_n \ldots s_{n+m}$ in TS
- Guess $n+m+2$ 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".

LTL Model Checking is PSPACE Complete

The LTL model-checking problem is PSPACE-complete.

Proof.

We have seen that the∃LTL model-checking problem is in NPSPACE = PSPACE. We will prove that this problem is PSPACE-hard. (This is done on the black board.) Since PSPACE = co-PSPACE, these two results together give that the LTL model-checking problem is PSPACE-complete.

We concentrate on PSPACE-hardness of the existential LTL model-checking problem. □

Existential LTL Model Checking is PSPACE-Hard

Overview

LTL Satisfiability is PSPACE Complete

The LTL satisfiability problem is PSPACE-complete.

Proof.

The satisfiability of LTL-formula φ amounts to checking whether $Words(\varphi) \neq \varnothing$. Automata-based satisfiability checking: construct the NBA $\mathfrak{A}_\varphi$. Check whether $\mathfrak{L}_\omega(\mathfrak{A}) \neq \varnothing$. This can be done using a nested DFS. The complexity is exponential in $|\varphi|$ . . . . . . and PSPACE-complete. □

Overview

Summary

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

Next Lecture

Thursday November 21, 10:30