Introduction

Modelling parallel systems

Linear Time Properties

Regular Properties

Linear Temporal Logic (LTL)

- syntax and semantics of LTL
- automata-based LTL model checking
- complexity of LTL model checking ⟵

Computation-Tree Logic

Equivalences and Abstraction

Complexity of LTL model checking

main steps of automata-based LTL model checking:

construction of an NBA $\mathcal{A}$ for $\neg\varphi$

persistence checking in the product $\mathcal{T} \otimes \mathcal{A}$

Complexity of LTL model checking

main steps of automata-based LTL model checking:

construction of an NBA $\mathcal{A}$ for $\neg\varphi$	$\longleftarrow \mathcal{O}(\exp(\lvert\varphi\rvert))$
persistence checking in the product $\mathcal{T} \otimes \mathcal{A}$	

Complexity of LTL model checking

main steps of automata-based LTL model checking:

construction of an NBA $\mathcal{A}$ for $\neg\varphi$	$\longleftarrow \mathcal{O}(\exp(\lvert\varphi\rvert))$
persistence checking in the product $\mathcal{T} \otimes \mathcal{A}$	$\longleftarrow \mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \mathit{size}(\mathcal{A}))$

Complexity of LTL model checking

main steps of automata-based LTL model checking:

construction of an NBA $\mathcal{A}$ for $\neg\varphi$	$\longleftarrow \mathcal{O}(\exp(\lvert\varphi\rvert))$
persistence checking in the product $\mathcal{T} \otimes \mathcal{A}$	$\longleftarrow \mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \mathit{size}(\mathcal{A}))$

complexity: $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(\lvert\varphi\rvert))$

Complexity of LTL model checking

main steps of automata-based LTL model checking:

construction of an NBA $\mathcal{A}$ for $\neg\varphi$	$\longleftarrow \mathcal{O}(\exp(\|\varphi\|))$
persistence checking in the product $\mathcal{T} \otimes \mathcal{A}$	$\longleftarrow \mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \mathit{size}(\mathcal{A}))$

complexity: $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(|\varphi|))$

The **LTL** model checking problem is **PSPACE**-complete

Complexity of LTL model checking

LTL model checking problem

given: finite transition system $\mathcal{T}$
LTL-formula φ

question: does $\mathcal{T} \models \varphi$ hold **?**

Complexity of LTL model checking

LTL model checking problem

given: finite transition system $\mathcal{T}$
LTL-formula φ

question: does $\mathcal{T} \models \varphi$ hold **?**

we show

- just for fun: **coNP**-hardness
- **PSPACE**-completeness

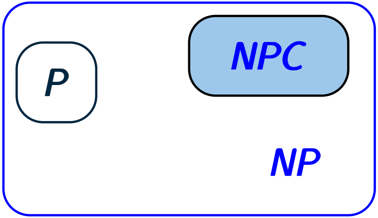
Recall: complexity classes

Complexity classes P, NP

P = class of decision problem solvable in deterministic polynomial time

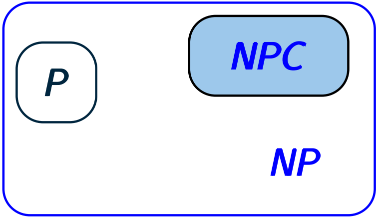
NP = class of decision problem solvable in nondeterministic polynomial time

Complexity classes *P*, *NP*



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

Complexity classes P, NP

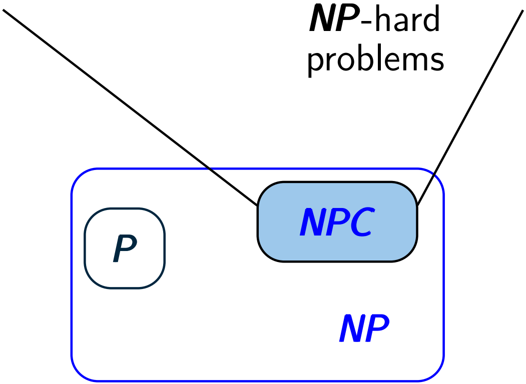


NPC = class of NP-complete problems

(1) $L \in NP$

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

Complexity classes *P*, *NP*

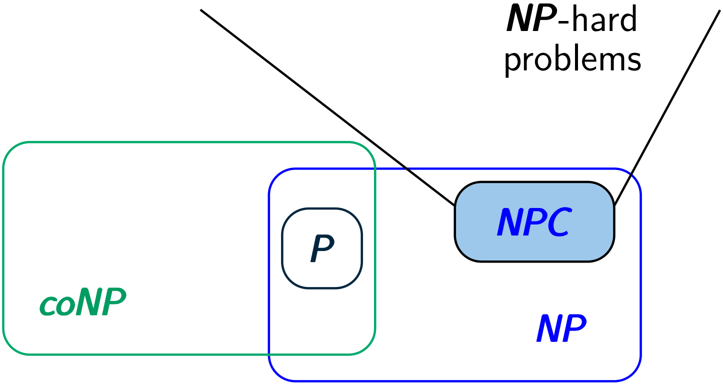


NPC = class of NP-complete problems

(1) $L \in NP$

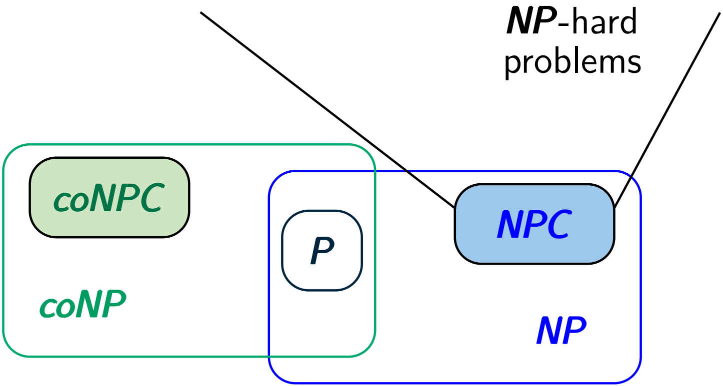
(2) L is NP-hard, i.e., $K \leq_{poly} L$ for all $K \in NP$

Complexity classes *P*, *NP*, *coNP*



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

$\uparrow$ complement of L



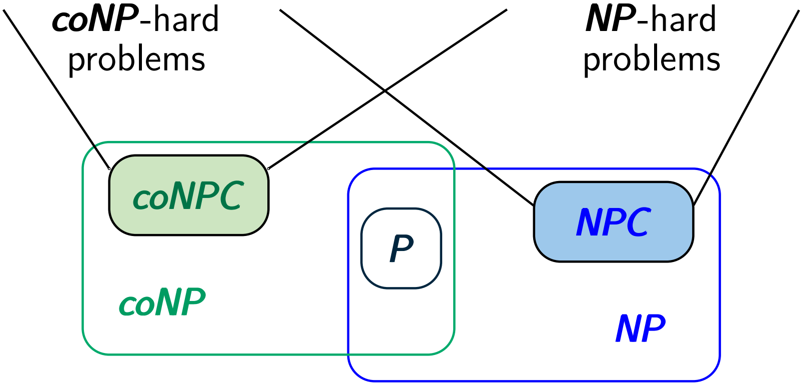
coNPC = class of ***coNP***-complete problems

↑

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

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

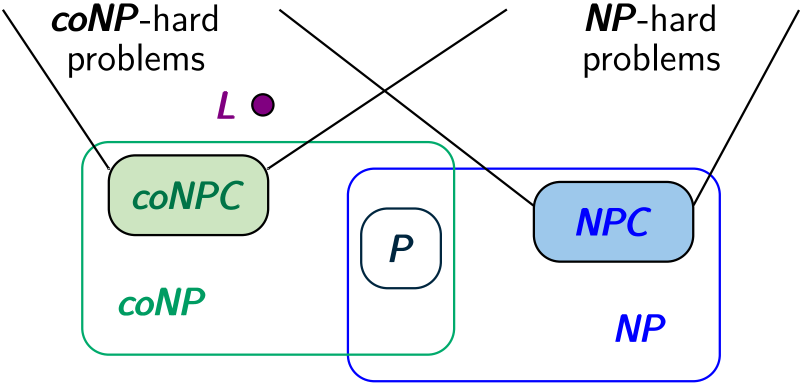
Complexity classes *P*, *NP*, *coNP*



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

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

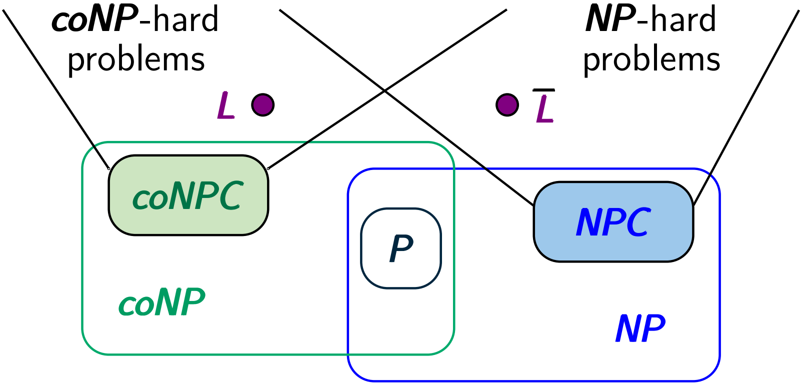
Complexity classes *P*, *NP*, *coNP*



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

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

Complexity classes *P*, *NP*, *coNP*



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

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

***coNP*-hardness**

The **LTL** model checking problem is ***coNP***-hard

The **LTL** model checking problem is ***coNP***-hard

proof by a polynomial reduction

$$HP \leq_{poly} \overline{LTL\text{-}MC}$$

HP: Hamilton path problem

$\overline{LTL\text{-}MC}$: complement of the **LTL** model checking problem

The **LTL** model checking problem is ***coNP***-hard

proof by a polynomial reduction

$$HP \leq_{poly} \overline{LTL\text{-}MC}$$

Hamilton path problem ↗ HP

complement of the **LTL** model checking problem ↖ $\overline{LTL\text{-}MC}$

complement of the **LTL** model checking problem:

given: finite transition system $\mathcal{T}$, LTL-formula φ

question: does $\mathcal{T} \not\models \varphi$ hold ?

coNP-hardness

The **LTL** model checking problem is ***coNP***-hard

proof by a polynomial reduction

$$HP \leq_{poly} \overline{LTL\text{-}MC}$$

Hamilton path problem ↗ *HP*

complement of the **LTL** model checking problem ↖ $\overline{LTL\text{-}MC}$

NP-complete ↑ Hamilton path problem

complement of the **LTL** model checking problem:

given: finite transition system $\mathcal{T}$, LTL-formula φ

question: does $\mathcal{T} \not\models \varphi$ hold ?

coNP-hardness

> The **LTL** model checking problem is ***coNP***-hard

proof by a polynomial reduction

$$\mathit{HP} \;\leq_{\mathit{poly}}\; \overline{\mathit{LTL\text{-}MC}}$$

Hamilton path problem ← **NP**-complete

complement of the **LTL** model checking problem ← **NP**-hard

complement of the **LTL** model checking problem:

given: finite transition system $\mathcal{T}$, LTL-formula φ

question: does $\mathcal{T} \not\models \varphi$ hold ?

Hamilton path problem *HP*

HP Hamilton path problem:

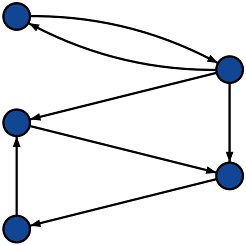
given: finite directed graph G

question: does G has a Hamilton path **?**, i.e., a path that visits each node exactly once

HP Hamilton path problem:

given: finite directed graph ***G***

question: does ***G*** has a Hamilton path **?**, i.e., a path that visits each node exactly once

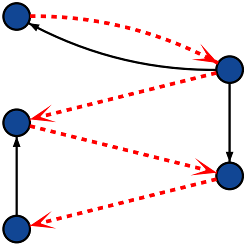


Hamilton path problem *HP*

HP Hamilton path problem:

given: finite directed graph G

question: does G has a Hamilton path **?**, i.e., a path that visits each node exactly once



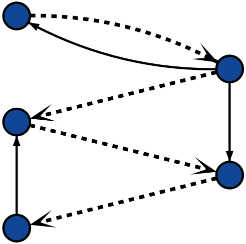
G has a Hamilton path

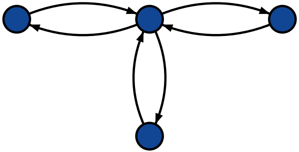
Hamilton path problem *HP*

HP Hamilton path problem:

given: finite directed graph G

question: does G has a Hamilton path **?**, i.e., a path that visits each node exactly once





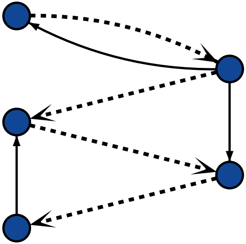
has no Hamilton path

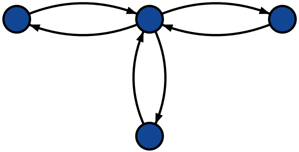
Hamilton path problem *HP*

HP Hamilton path problem:

given: finite directed graph ***G***

question: does ***G*** has a Hamilton path **?**, i.e., a path that visits each node exactly once





has no Hamilton path

HP is known to be ***NP***-complete

Polynomial reduction

$$HP \quad \leq_{poly} \quad \overline{\textit{LTL-MC}}$$

Hamilton path problem ↗ HP

↖ $\overline{\textit{LTL-MC}}$: complement of the **LTL** model checking problem

Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

Hamilton path problem $\nearrow$ HP

$\overline{LTL\text{-}MC}$ $\nwarrow$ complement of the **LTL** model checking problem

finite directed graph G	$\xrightarrow{\text{poly time}}$	finite TS $\mathcal{T}$, LTL formula φ

Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

Hamilton path problem

complement of the **LTL** model checking problem

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

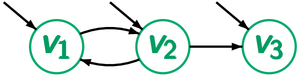
Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

---

node-set V of G $\quad \widehat{=} \quad$ states of $\mathcal{T}$



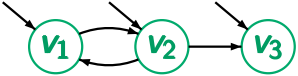
Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

---

node-set V of G $\quad \widehat{=} \quad$ states of $\mathcal{T}$ $\quad AP = V$



Polynomial reduction

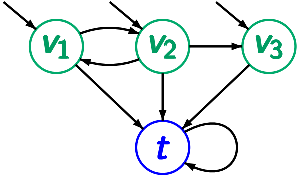
HP $\leq_{poly}$ $\overline{\textit{LTL-MC}}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

---

node-set V of G $\widehat{=}$ states of $\mathcal{T}$ $AP = V$

additional trap state t



Polynomial reduction

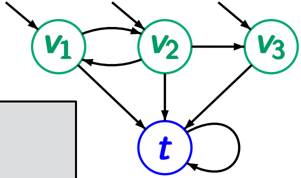
$HP \leq_{poly} \overline{LTL\text{-}MC}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

---

node-set V of G $\widehat{=}$ states of $\mathcal{T}$ $AP = V$

additional trap state t



$\varphi = ?$

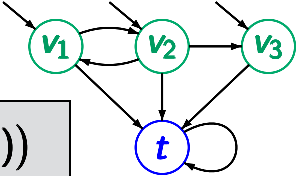
$$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

---

node-set V of G $\quad\widehat{=}\quad$ states of $\mathcal{T}$ $\quad AP = V$

additional trap state t



$$\varphi = \bigwedge_{v \in V} \left(\Diamond v \wedge \Box(v \rightarrow \bigcirc\Box\neg v) \right)$$

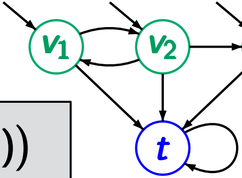
Polynomial reduction

$HP \quad \leq_{poly} \quad \overline{LTL\text{-}MC}$

finite directed graph G	poly time $\longrightarrow$	finite TS $\mathcal{T}$ LTL formula φ
G has a Hamilton path	iff	$\mathcal{T} \not\models \varphi$

node-set V of G $\quad\widehat{=}\quad$ states of $\mathcal{T}$ $\quad AP = V$

additional trap state t



$$\varphi = \neg \bigwedge_{v \in V} \left(\Diamond v \wedge \Box(v \rightarrow \bigcirc \Box \neg v) \right)$$

Complexity of LTL model checking

We just saw:

The **LTL** model checking problem is ***coNP***-hard

Complexity of LTL model checking

We just saw:

The **LTL** model checking problem is ***coNP***-hard

We now prove:

The **LTL** model checking problem is ***PSPACE***-complete

The complexity class *PSPACE*

The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- ***NP*** $\subseteq$ ***PSPACE***

The complexity class *PSPACE*

> ***PSPACE*** = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- ***NP*** ⊆ ***PSPACE***

  ↑ **DFS**-based analysis of the computation tree of an ***NP***-algorithm

The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- *NP* $\subseteq$ *PSPACE*

  ↑

  DFS-based analysis of the computation tree of an *NP*-algorithm

  space requirements:
  recursion depth $\widehat{=}$ height of computation tree

The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- ***NP*** $\subseteq$ ***PSPACE***
- ***PSPACE*** = ***coPSPACE***

  (holds for any deterministic complexity class)

The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- ***NP*** $\subseteq$ ***PSPACE***
- ***PSPACE*** = ***coPSPACE***
  (holds for any deterministic complexity class)
- ***PSPACE*** = ***NPSPACE*** (Savitch's Theorem)

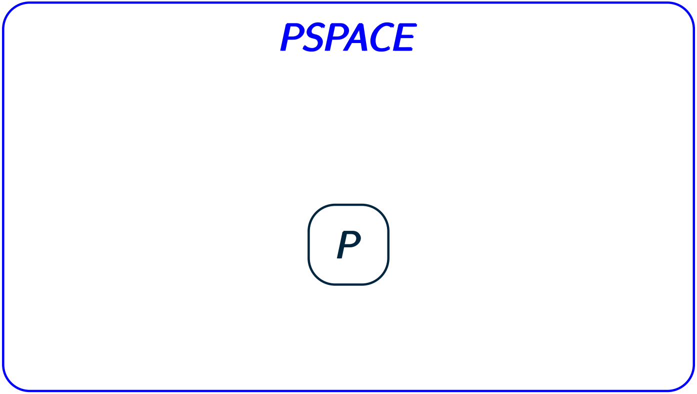
The complexity class *PSPACE*

PSPACE = class of decision problems solvable by a deterministic polynomially space-bounded algorithm

- *NP* $\subseteq$ *PSPACE*
- *PSPACE* = *coPSPACE*
  (holds for any deterministic complexity class)
- *PSPACE* = *NPSPACE* (Savitch's Theorem)

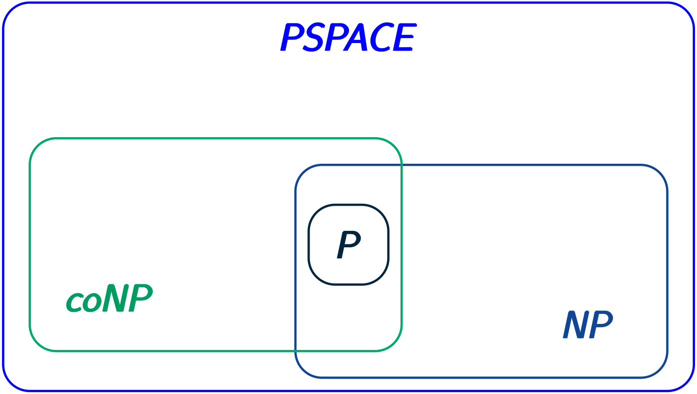
To prove $L \in$ *PSPACE* it suffices to provide a nondeterministic polynomially space-bounded algorithm for the complement $\overline{L}$ of L

Complexity classes *P*, *NP*, *coNP*, *PSPACE*



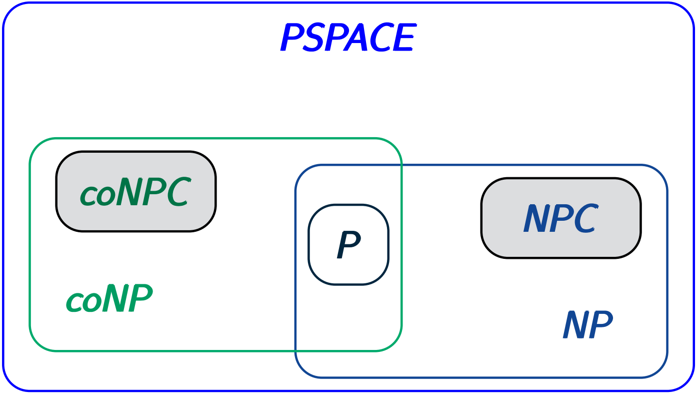
PSPACE = class of decision problems that are decidable in deterministic polynomial space

Complexity classes *P*, *NP*, *coNP*, *PSPACE*



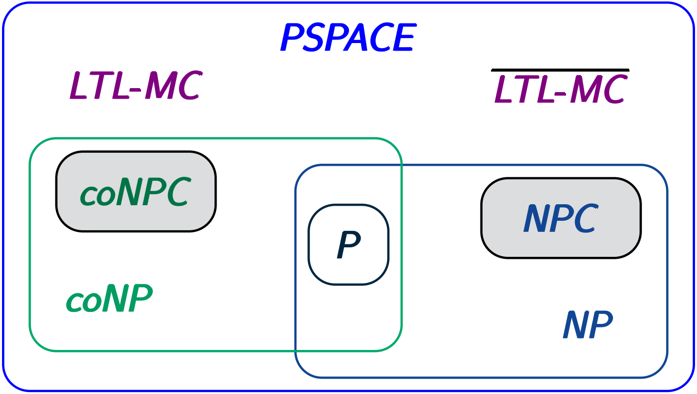
PSPACE = class of decision problems that are decidable in deterministic polynomial space

Complexity classes *P*, *NP*, *coNP*, *PSPACE*



PSPACE = class of decision problems that are decidable in deterministic polynomial space

Complexity classes *P*, *NP*, *coNP*, *PSPACE*



PSPACE = class of decision problems that are decidable in deterministic polynomial space

PSPACE-completeness

PSPACE-completeness

decision problem L is *PSPACE*-complete iff

(1) $L \in PSPACE$

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

PSPACE-completeness

decision problem L is *PSPACE*-complete iff

(1) $L \in PSPACE$

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

as $PSPACE = coPSPACE$:

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

PSPACE-completeness

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

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

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$ **?**"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$ **?**"

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$ **?**"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$ **?**"

↑

$\exists$***LTL-MC*** existential LTL model checking problem
for $\mathcal{T}$ and LTL formula $\psi = \neg\varphi$

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$?"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$?"

↑

∃*LTL-MC* existential LTL model checking problem for $\mathcal{T}$ and LTL formula $\psi = \neg\varphi$

"does $\pi \models \psi$ hold for some path π of $\mathcal{T}$?"

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$?"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$?"

↑

∃LTL-MC existential LTL model checking problem for $\mathcal{T}$ and LTL formula $\psi = \neg\varphi$

"does $\pi \models \psi$ hold for some path π of $\mathcal{T}$?"

show: ***∃LTL-MC*** $\in$ ***NPSPACE***

∃LTL-MC is ***PSPACE***-hard

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$?"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$?"

↑

∃LTL-MC existential LTL model checking problem for $\mathcal{T}$ and LTL formula $\psi = \neg\varphi$

"does $\pi \models \psi$ hold for some path π of $\mathcal{T}$?"

show: ***∃LTL-MC*** $\in$ ***NPSPACE*** $\Longrightarrow$ ***LTL-MC*** $\in$ ***PSPACE***

∃LTL-MC is ***PSPACE***-hard

PSPACE-completeness of *LTL-MC*

LTL-MC LTL model checking problem

"does $\pi \models \varphi$ hold for all paths π of $\mathcal{T}$ **?**"

$\overline{\textit{LTL-MC}}$ = complement of ***LTL-MC***

"does $\pi \not\models \varphi$ hold for some path π of $\mathcal{T}$ **?**"

↑

∃*LTL-MC* existential LTL model checking problem for $\mathcal{T}$ and LTL formula $\psi = \neg\varphi$

"does $\pi \models \psi$ hold for some path π of $\mathcal{T}$ **?**"

show: ∃*LTL-MC* ∈ ***NPSPACE***

∃*LTL-MC* is ***PSPACE***-hard ⟹ ***LTL-MC*** is ***PSPACE***-hard

Existential LTL model checking problem

given:	$\mathcal{T}$ be a finite transition system
	φ an LTL formula
question:	does there exist a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

Existential LTL model checking problem

given:	$\mathcal{T}$ be a finite transition system
	φ an LTL formula
question:	does there exist a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

goal: find a criterion for the existence of a path π in $\mathcal{T}$ with $\pi \models \varphi$ that can be checked nondeterministically in poly-space

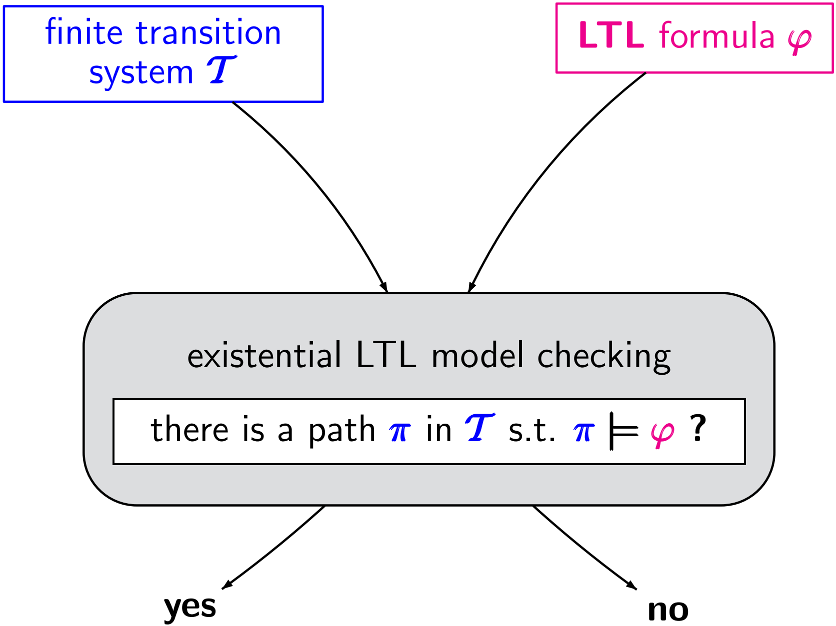
Existential LTL model checking problem

given:	$\mathcal{T}$ be a finite transition system
	φ an LTL formula
question:	does there exist a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

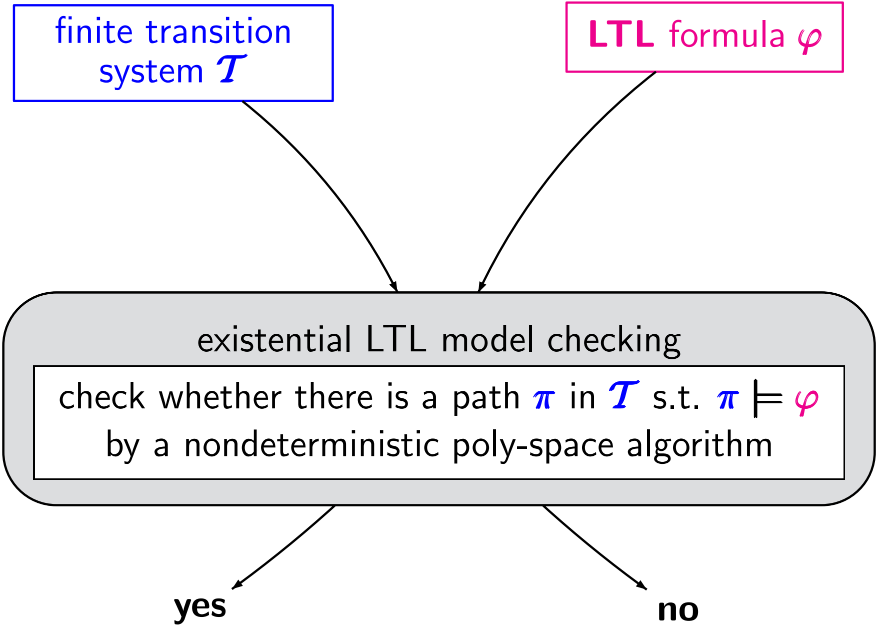
goal: find a criterion for the existence of a path π in $\mathcal{T}$ with $\pi \models \varphi$ that can be checked nondeterministically in poly-space

idea: use the **GNBA** $\mathcal{G}$ for φ (constructed by our LTL-2-GNBA algorithm)

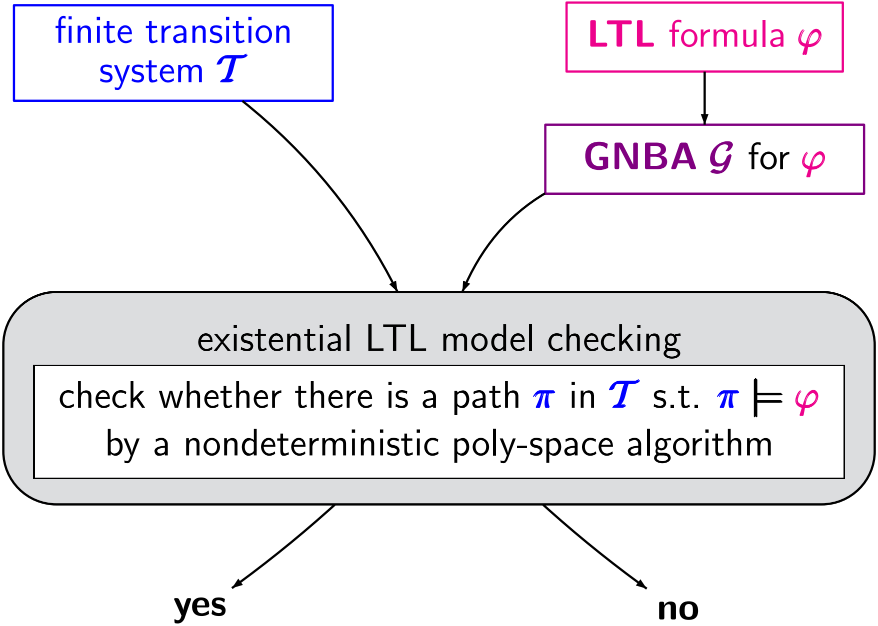
Existential LTL model checking



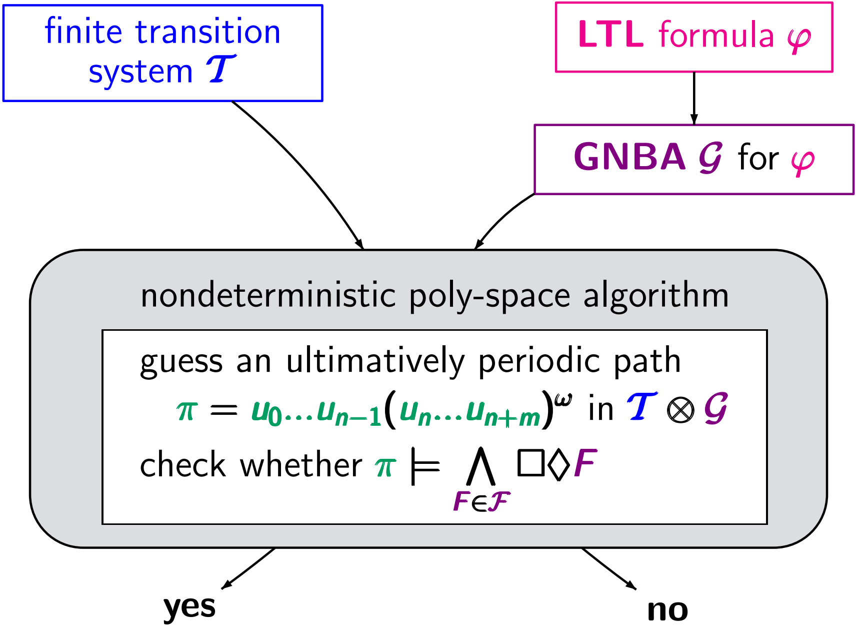
Existential LTL model checking



Existential LTL model checking



Existential LTL model checking



Recall: elementary formula-sets

closure $cl(\varphi)$:

- set of all subformulas of φ and their negations
- ψ and $\neg\neg\psi$ are identified

elementary formula-sets: subsets B of $cl(\varphi)$

- maximal consistent w.r.t. propositional logic
- locally consistent w.r.t. **U**

For $\varphi = a \, \mathsf{U}(\neg a \wedge b)$, the elementary sets are:

$\{\ a,\ \ b, \neg(\neg a \wedge b), \varphi\}$ $\{\ a,\ \ b, \neg(\neg a \wedge b), \neg\varphi\}$

$\{\ a, \neg b, \neg(\neg a \wedge b), \varphi\}$ $\{\ a, \neg b, \neg(\neg a \wedge b), \neg\varphi\}$

$\{\neg a,\ \ b,\ \ \neg a \wedge b\ , \varphi\}$ $\{\neg a, \neg b, \neg(\neg a \wedge b), \neg\varphi\}$

Recall: GNBA for LTL-formula φ

$$\mathcal{G} = (Q, 2^{AP}, \delta, Q_0, \mathcal{F})$$

state space: $Q = \{B \subseteq cl(\varphi) : B \text{ is elementary}\}$

initial states: $Q_0 = \{B \in Q : \varphi \in B\}$

transition relation: for $B \in Q$ and $A \in 2^{AP}$:

if $A \neq B \cap AP$ then $\delta(B, A) = \varnothing$

if $A = B \cap AP$ then $\delta(B, A) =$ set of all $B' \in Q$ s.t.

$$\bigcirc\psi \in B \quad \text{iff} \quad \psi \in B'$$

$$\psi_1 \mathsf{U} \psi_2 \in B \quad \text{iff} \quad (\psi_2 \in B) \vee (\psi_1 \in B \wedge \psi_1 \mathsf{U} \psi_2 \in B')$$

acceptance set $\mathcal{F} = \{F_{\psi_1 \mathsf{U} \psi_2} : \psi_1 \mathsf{U} \psi_2 \in cl(\varphi)\}$

where $F_{\psi_1 \mathsf{U} \psi_2} = \{B \in Q : \psi_1 \mathsf{U} \psi_2 \notin B \vee \psi_2 \in B\}$

Criterion for existential LTL properties

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

- an initial finite path fragment $s_0 \ldots s_n \ldots s_{n+m}$ in $\mathcal{T}$
- a run $B_0 \, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in $\mathcal{G}$ for the word
  $trace(s_0 \, s_1 \ldots s_n \ldots s_{n+m})$

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

- an initial finite path fragment $s_0 \ldots s_n \ldots s_{n+m}$ in $\mathcal{T}$
- a run $B_0\, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in $\mathcal{G}$ for the word $trace(s_0\, s_1 \ldots s_n \ldots s_{n+m})$

such that

$$(1) \quad \langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$$

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

- an initial finite path fragment $s_0 \ldots s_n \ldots s_{n+m}$ in $\mathcal{T}$
- a run $B_0\, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in $\mathcal{G}$ for the word $trace(s_0\, s_1 \ldots s_n \ldots s_{n+m})$

such that

(1) $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$

(2) whenever $\psi_1 \mathsf{U} \psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$ then $\psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

- an initial finite path fragment $s_0 \ldots s_n \ldots s_{n+m}$ in $\mathcal{T}$
- a run $B_0\, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in $\mathcal{G}$ for the word $trace(s_0\, s_1 \ldots s_n \ldots s_{n+m})$

such that

(1) $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$

(2) whenever $\psi_1 \,\mathsf{U}\, \psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$ then $\psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$

(3) $n < |S| \cdot 2^{|cl(\varphi)|}$

Criterion for existential LTL properties

There exists a path π in $\mathcal{T}$ with $\pi \models \varphi$ iff

there exist

- an initial finite path fragment $s_0 \ldots s_n \ldots s_{n+m}$ in $\mathcal{T}$
- a run $B_0\, B_1 \ldots B_{n+1} \ldots B_{n+m+1}$ in $\mathcal{G}$ for the word $trace(s_0\, s_1 \ldots s_n \ldots s_{n+m})$

such that

(1) $\langle s_n, B_{n+1}\rangle = \langle s_{n+m}, B_{n+m+1}\rangle$

(2) whenever $\psi_1 \mathbf{U}\, \psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$ then $\psi_2 \in B_{n+1} \cup \ldots \cup B_{n+m}$

(3) $n < |S| \cdot 2^{|cl(\varphi)|}$ and $m \leq |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

NPSPACE-algorithm for $\exists$*LTL-MC*

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path $\pi \in \mathit{Paths}(\mathcal{T})$ with $\pi \models \varphi$?

NPSPACE-algorithm for $\exists$*LTL-MC*

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path $\pi \in \mathit{Paths}(\mathcal{T})$ with $\pi \models \varphi$?

is solvable by a nondeterministic polynomially space-bounded algorithm:

NPSPACE-algorithm for $\exists$*LTL-MC*

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path $\pi \in \mathit{Paths}(\mathcal{T})$ with $\pi \models \varphi$?

is solvable by a nondeterministic polynomially space-bounded algorithm:

- guess nondeterministically an ultimatively periodic path $\pi = u_0\, u_1 \ldots u_{n-1}(u_n \ldots u_{n+m})^\omega$ of $\mathcal{T} \otimes \mathcal{G}$

  ↑ GNBA for φ obtained by our LTL-2-GNBA algorithm

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path $\pi \in Paths(\mathcal{T})$ with $\pi \models \varphi$?

is solvable by a nondeterministic polynomially space-bounded algorithm:

- guess nondeterministically an ultimately periodic path $\pi = u_0\, u_1 \ldots u_{n-1}(u_n \ldots u_{n+m})^\omega$ of $\mathcal{T} \otimes \mathcal{G}$

  ↑ GNBA for φ obtained by our LTL-2-GNBA algorithm

- check whether the guessed path meets the acceptance condition of $\mathcal{G}$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

guess $n+m+2$ subsets $B_0, \ldots, B_n, \ldots, B_{n+m+1}$ of $cl(\varphi)$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

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:

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

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:

- $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

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:

- $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$
- $B_0 \ldots B_n \ldots B_{n+m+1}$ is an initial run for $trace(s_0 \ldots s_n \ldots s_{n+m+1})$ in GNBA $\mathcal{G}$

NPSPACE-algorithm for $\exists$*LTL-MC*

guess two natural numbers $n, m \leq k$ s.t. $m \geq 1$

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

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

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:

- $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$
- $B_0 \ldots B_n \ldots B_{n+m+1}$ is an initial run for $trace(s_0 \ldots s_n \ldots s_{n+m+1})$ in GNBA $\mathcal{G}$
- $\left\{ \psi_2 : \psi_1 \mathsf{U} \psi_2 \in \bigcup_{n<i\leq n+m} B_i \right\} \subseteq \bigcup_{n<i\leq n+m} B_i$

NPSPACE-algorithm for ∃*LTL-MC*

guess two natural numbers $n, m \le k$ s.t. $m \ge 1$
where $k = |S| \cdot 2^{|cl(\varphi)|} \cdot |\varphi|$

guess $s_0 \ldots s_n \ldots s_{n+m} \in Paths_{fin}(\mathcal{T})$

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:

- $\langle s_n, B_{n+1} \rangle = \langle s_{n+m}, B_{n+m+1} \rangle$
- $B_0 \ldots B_n \ldots B_{n+m+1}$ is an initial run for $trace(s_0 \ldots s_n \ldots s_{n+m+1})$ in GNBA $\mathcal{G}$
- $\left\{ \psi_2 : \psi_1 \mathsf{U} \psi_2 \in \bigcup_{n<i\le n+m} B_i \right\} \subseteq \bigcup_{n<i\le n+m} B_i$

If so then return "**yes**". Otherwise return "**no**".

PSPACE-completeness of $\exists$*LTL-MC*

We saw that:

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ

question: is there a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

belongs to ***NPSPACE***

PSPACE-completeness of ∃*LTL*-*MC*

We saw that:

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

belongs to ***NPSPACE*** = ***PSPACE***

PSPACE-completeness of $\exists$*LTL-MC*

We saw that:

The existential LTL model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ
question: is there a path π in $\mathcal{T}$ with $\pi \models \varphi$?

belongs to ***NPSPACE*** = ***PSPACE***

It remains to prove the *PSPACE*-hardness

PSPACE-hardness of ∃*LTL-MC*

we show that for all problems $K \in PSPACE$:

$$K \leq_{poly} \exists LTL\text{-}MC$$

we show that for all problems $K \in PSPACE$:

$$K \leq_{poly} \exists LTL\text{-}MC$$

Let

- $\mathcal{M}$ be a deterministic Turing machine (DTM) that decides K,
- P a polynomial

such that $\mathcal{M}$ started with an input word w visits at most $P(|w|)$ tape cells

PSPACE-hardness of $\exists$*LTL-MC*

we show that for all problems $K \in \mathit{PSPACE}$:

$$K \leq_{poly} \exists \mathit{LTL\text{-}MC}$$

Given DTM $\mathcal{M}$ that decides K with polynomial space bound $P(n)$, provide a polynomial reduction:

input word w for $\mathcal{M}$	poly time $\longrightarrow$	TS $\mathcal{T}$ LTL-formula φ

PSPACE-hardness of ∃*LTL-MC*

we show that for all problems $K \in \mathit{PSPACE}$:

$$K \leq_{poly} \exists \mathit{LTL\text{-}MC}$$

Given DTM $\mathcal{M}$ that decides K with polynomial space bound $P(n)$, provide a polynomial reduction:

input word w for $\mathcal{M}$	$\xrightarrow{\text{poly time}}$	TS $\mathcal{T}$ LTL-formula φ
$\mathcal{M}$ accepts w, i.e., $w \in K$	iff	there is path π of $\mathcal{T}$ with $\pi \models \varphi$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, \ldots, P(n)$ for inputs of length n (where P is a polynomial)

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, \ldots, P(n)$ for inputs of length n (where P is a polynomial)

\$	A_1	$\ldots$	A_{n-1}	A_n	$\sqcup$	$\sqcup$	$\ldots$	$\sqcup$	$\ldots$
1	2		n	$n+1$					

initial tape configuration for input $w = A_1 A_2 \ldots A_n$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, ..., P(n)$ for inputs of length n (where P is a polynomial)

$	A_1	...	A_{n-1}	A_n	⊔	⊔	...	⊔	...
1	2		n	$n+1$					

initial tape configuration for input $w = A_1 A_2 \ldots A_n$

⊔ $\widehat{=}$ blank symbol of $\mathcal{M}$

$ $\widehat{=}$ symbol for the left border of the tape

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, ..., P(n)$ for inputs of length n (where P is a polynomial)

\$	A_1	...	A_{n-1}	A_n	$\sqcup$	$\sqcup$	...	$\sqcup$	...
1	2		n	$n+1$				$P(n)$	

initial tape configuration for input $w = A_1 A_2 \ldots A_n$

$\sqcup \;\widehat{=}\;$ blank symbol of $\mathcal{M}$

\$ $\;\widehat{=}\;$ symbol for the left border of the tape

w.l.o.g. $P(n) > n$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, ..., P(n)$ for inputs of length n (where P is a polynomial)

$\$$	A_1	...	A_{n-1}	A_n	$\sqcup$	$\sqcup$	...	$\sqcup$	...
1	2		n	$n+1$				$P(n)$	↑ not visited

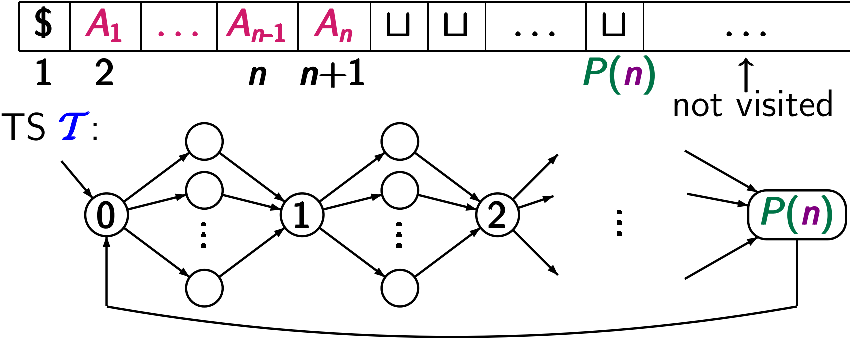
initial tape configuration for input $w = A_1 A_2 \ldots A_n$

$\sqcup \quad \widehat{=} \quad$ blank symbol of $\mathcal{M}$

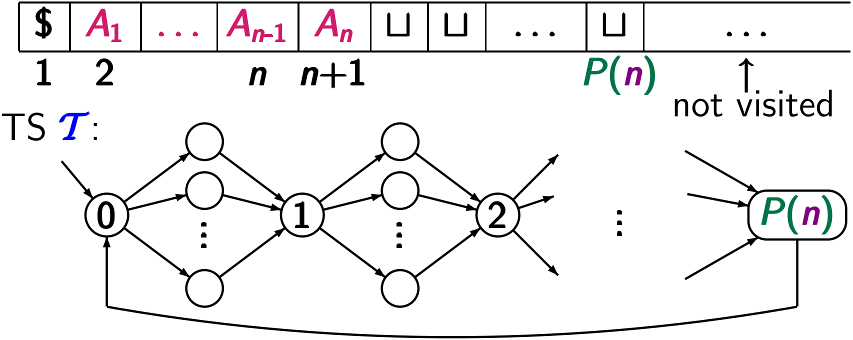
$\$ \quad \widehat{=} \quad$ symbol for the left border of the tape

w.l.o.g. $P(n) > n$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, \ldots, P(n)$ for inputs of length n (where P is a polynomial)



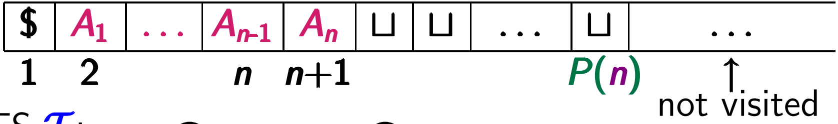
DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, ..., P(n)$ for inputs of length n (where P is a polynomial)



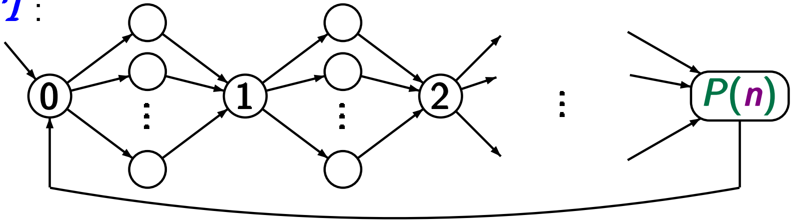
states of $\mathcal{T}$: $0, 1, \ldots, P(n)$,

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, \ldots, P(n)$ for inputs of length n (where P is a polynomial)



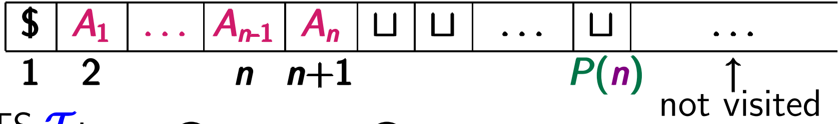
TS $\mathcal{T}$:



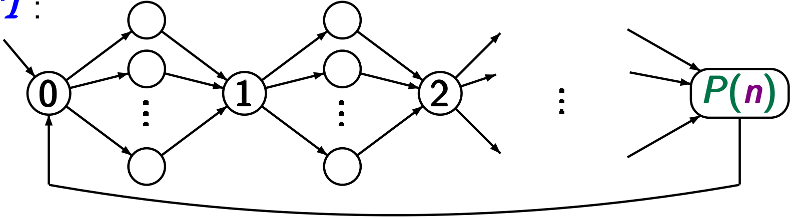
states of $\mathcal{T}$: $0, 1, \ldots, P(n)$, $\langle q, A, i \rangle$, $\langle *, A, i \rangle$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, ..., P(n)$ for inputs of length n (where P is a polynomial)



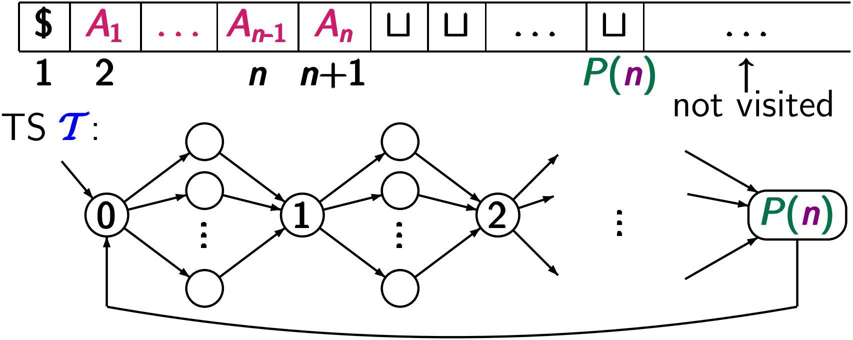
TS $\mathcal{T}$:



states of $\mathcal{T}$: $0, 1, \ldots, P(n)$, $\langle q, A, i \rangle$, $\langle *, A, i \rangle$
where q is a state of $\mathcal{M}$, A a tape symbol, $1 \leq i \leq P(n)$

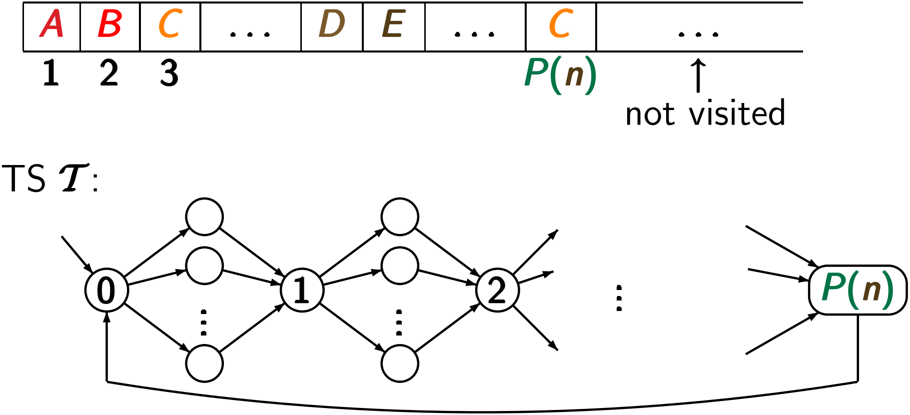
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

DTM $\mathcal{M}$ visits at the most the tape cells $1, 2, \ldots, P(n)$ for inputs of length n (where P is a polynomial)

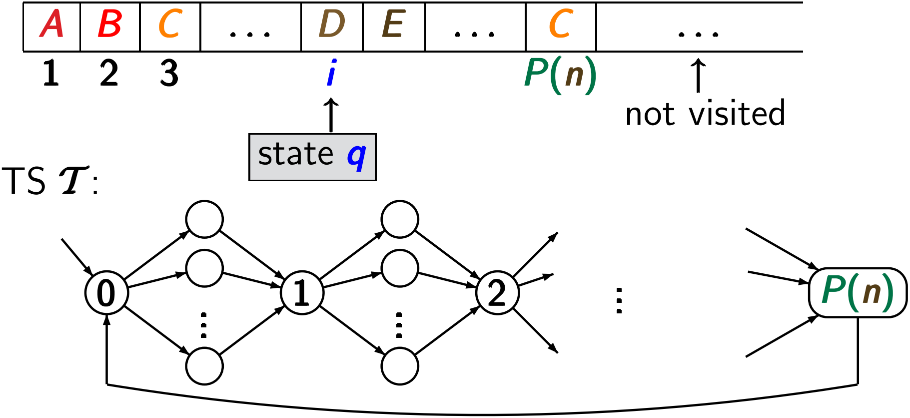


idea: TS $\mathcal{T}$ encodes each configuration of $\mathcal{M}$ by a path fragment from state 0 to state $P(n)$

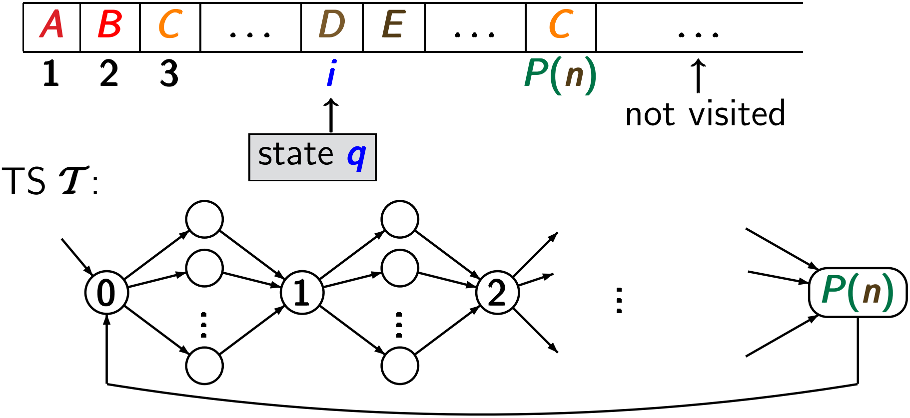
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

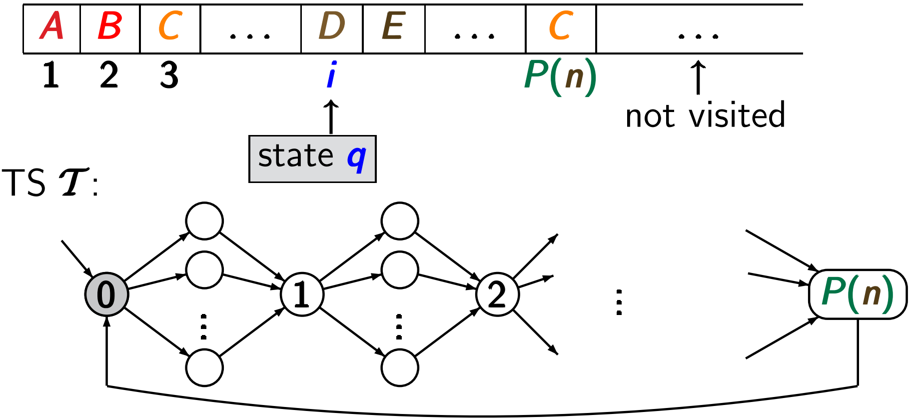


Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



suppose $\delta(q, D) = (p, B, +1)$

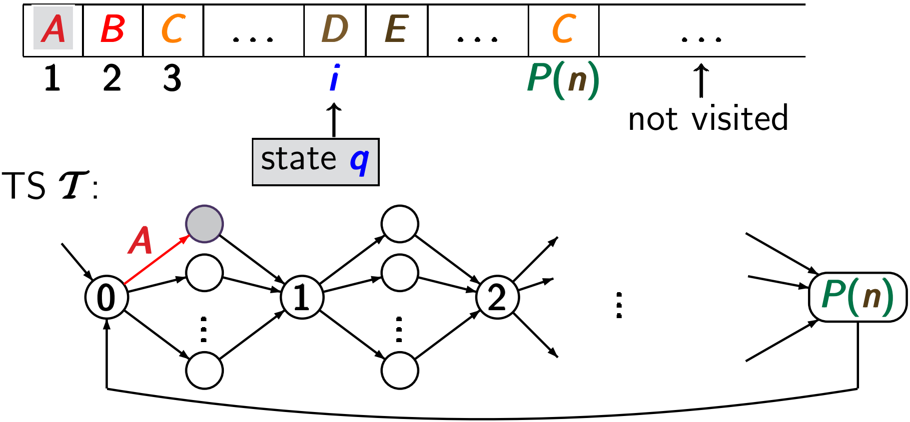
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



0

path fragment for the configuration $ABC\ldots\, q\, D\ldots C$

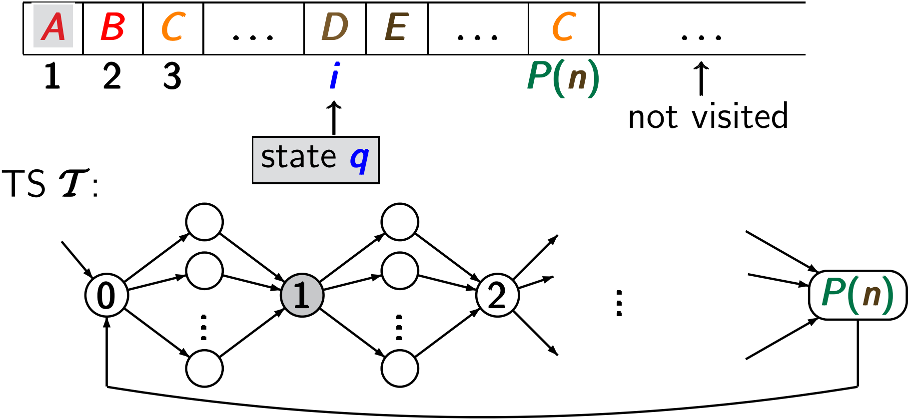
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1 \rangle$

path fragment for the configuration $ABC\ldots\ q\ D\ldots C$

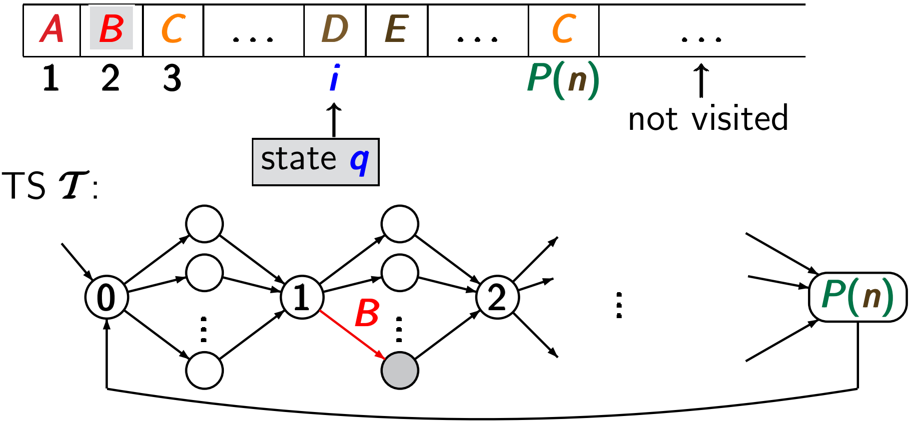
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \langle *, A, 1 \rangle 1$

path fragment for the configuration $ABC \ldots q\ D \ldots C$

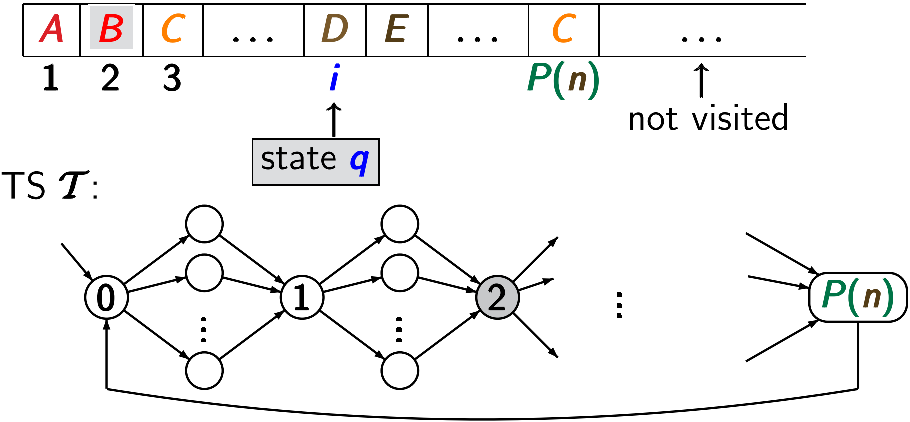
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1 \rangle \ 1 \ \langle *, B, 2 \rangle$

path fragment for the configuration $ABC\ldots\ q\ D\ldots C$

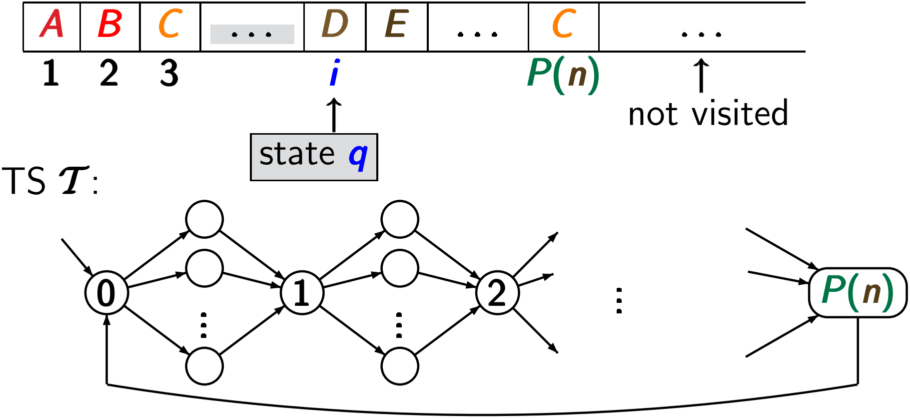
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1 \rangle \ 1 \ \langle *, B, 2 \rangle \ 2$

path fragment for the configuration $ABC\ldots\ q\ D\ldots C$

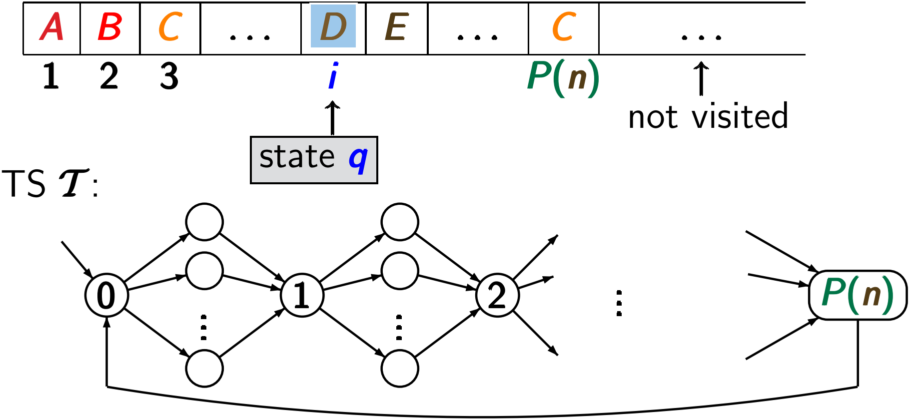
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \; \langle *, A, 1 \rangle \; 1 \; \langle *, B, 2 \rangle \; 2 \; \ldots \; (i{-}1)$

path fragment for the configuration $ABC \ldots\, q \; D \ldots C$

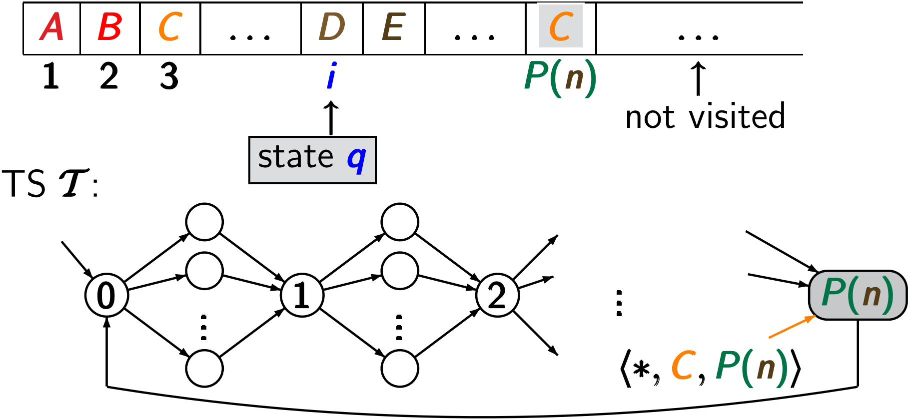
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$$0 \ \langle *, A, 1\rangle \ 1 \ \langle *, B, 2\rangle \ 2 \ \ldots \ (i{-}1)\langle q, D, i\rangle$$

path fragment for the configuration $ABC\ldots\ q\ D\ldots C$

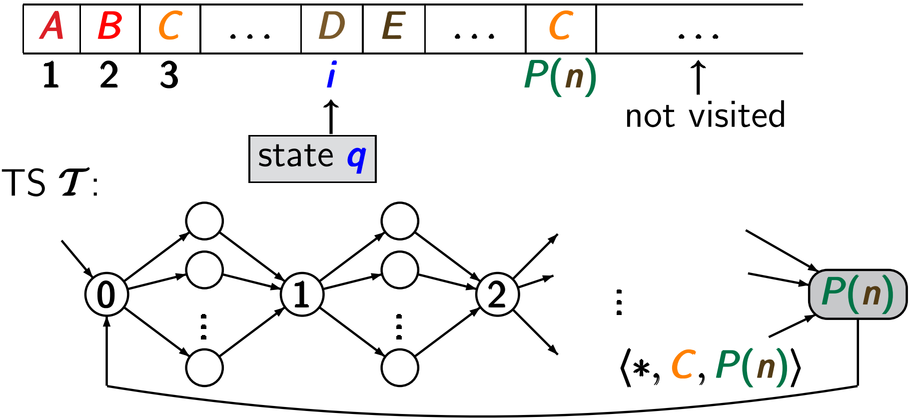
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ (i{-}1)\langle q, D, i\rangle\ i\ \ \ldots\ \ \ P(n)$$

path fragment for the configuration $ABC\ldots\ q\ D\ldots C$

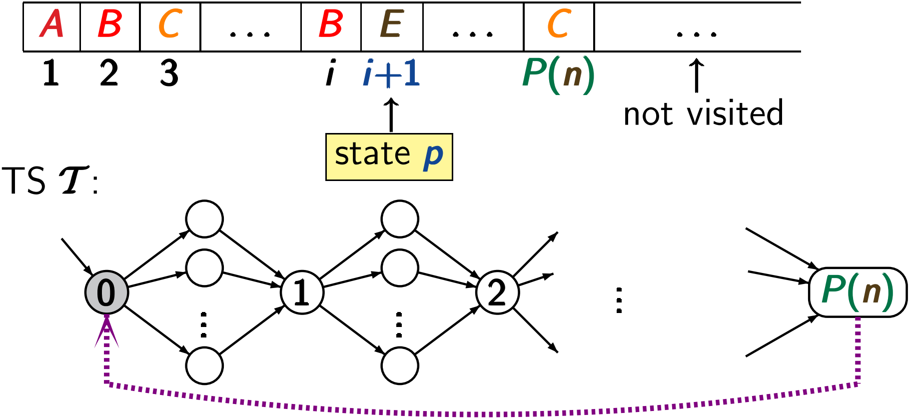
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1\rangle \ 1 \ \langle *, B, 2\rangle \ 2 \ \ldots \ (i{-}1)\langle q, D, i\rangle \ i \ \ \ldots \qquad P(n)$

suppose $\delta(q, D) = (p, B, +1)$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

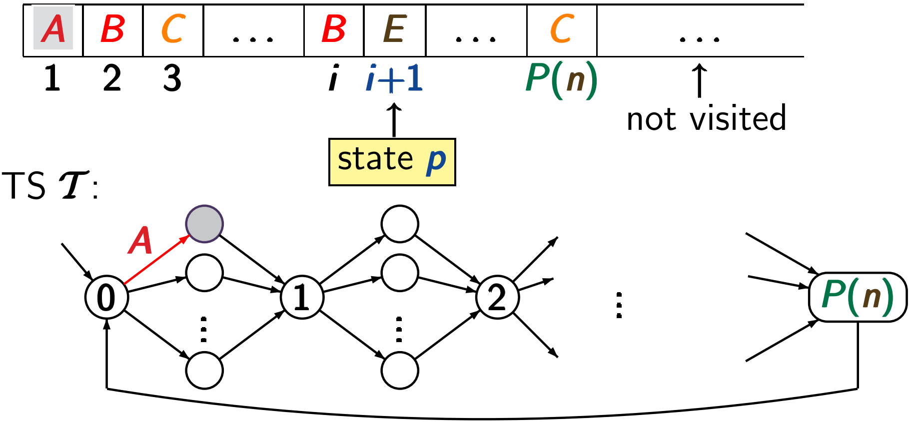


$0 \; \langle *, A, 1\rangle \; 1 \; \langle *, B, 2\rangle \; 2 \; \ldots \; \langle q, D, i\rangle \; i \; \langle *, E, i{+}1\rangle \; \ldots \; P(n)$

0

suppose $\delta(q, D) = (p, B, +1)$

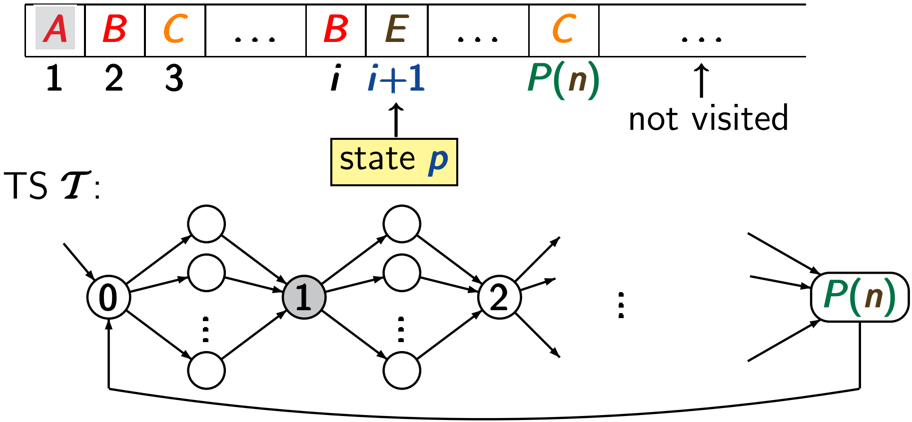
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1\rangle \ 1 \ \langle *, B, 2\rangle \ 2 \ \ldots \ \langle q, D, i\rangle \ i \ \langle *, E, i{+}1\rangle \ \ldots \ P(n)$

$0 \ \langle *, A, 1\rangle$

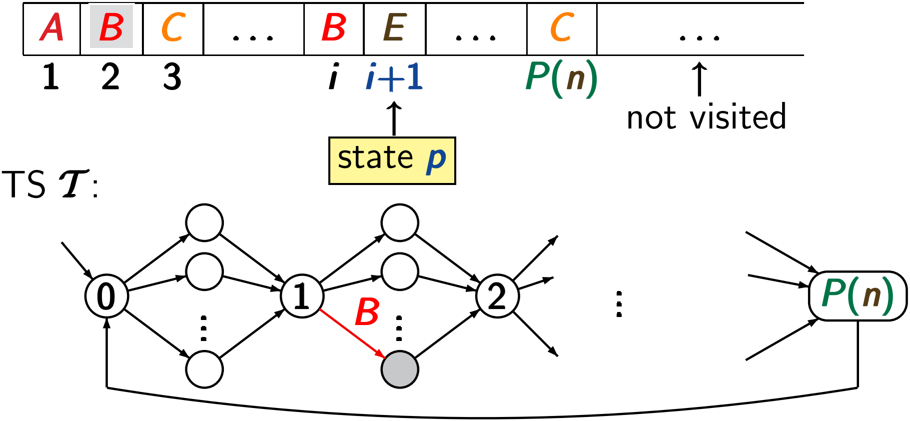
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle q, D, i\rangle\ i\ \langle *, E, i{+}1\rangle\ \ldots\ P(n)$

$0\ \langle *, A, 1\rangle\ 1$

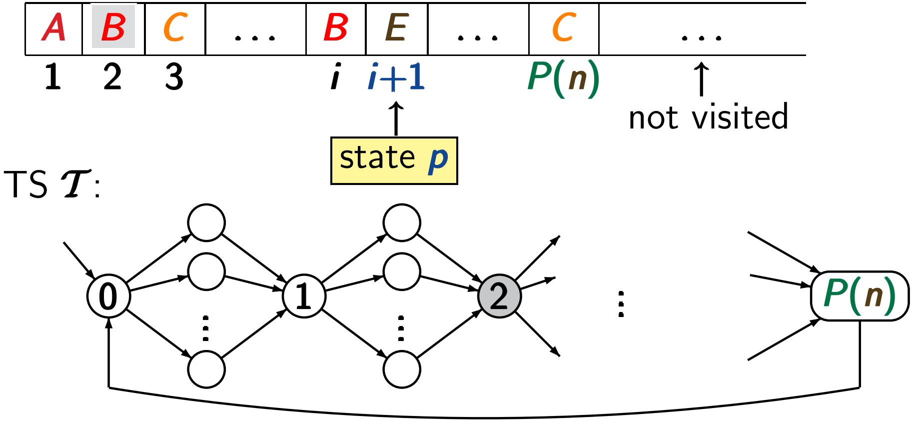
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0 \ \langle *, A, 1\rangle \ 1 \ \langle *, B, 2\rangle \ 2 \ \ldots \ \langle q, D, i\rangle \ i \ \langle *, E, i{+}1\rangle \ \ldots \ P(n)$

$0 \ \langle *, A, 1\rangle \ 1 \ \langle *, B, 2\rangle$

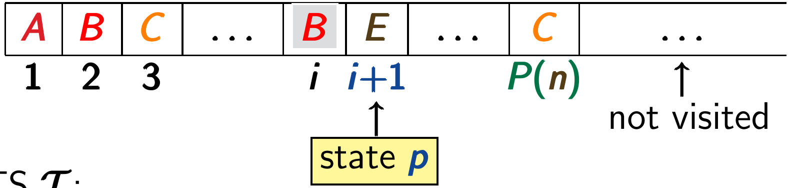
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



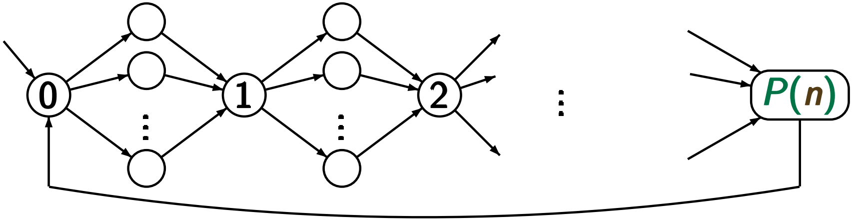
$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle q, D, i\rangle\ i\ \langle *, E, i{+}1\rangle\ \ldots\ P(n)$

$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



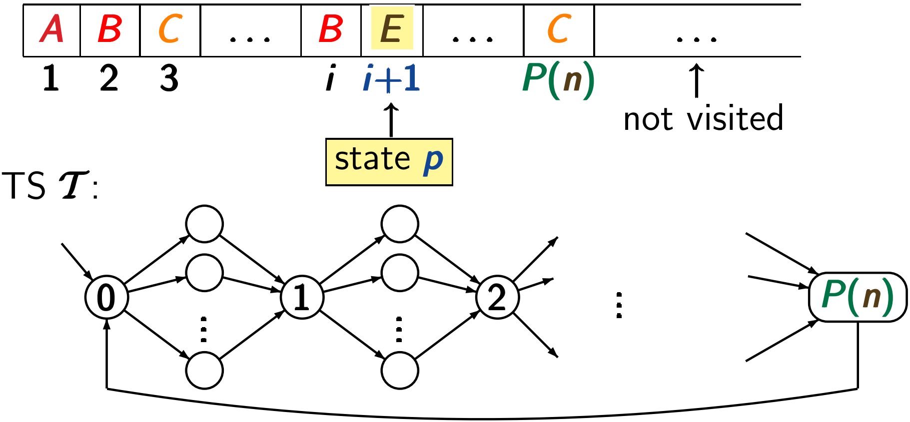
TS $\mathcal{T}$:



$0 \ \langle *, A, 1 \rangle \ 1 \ \langle *, B, 2 \rangle \ 2 \ \ldots \ \langle q, D, i \rangle \ i \ \langle *, E, i{+}1 \rangle \ \ldots \ P(n)$

$0 \ \langle *, A, 1 \rangle \ 1 \ \langle *, B, 2 \rangle \ 2 \ \ldots \ \langle *, B, i \rangle \ i$

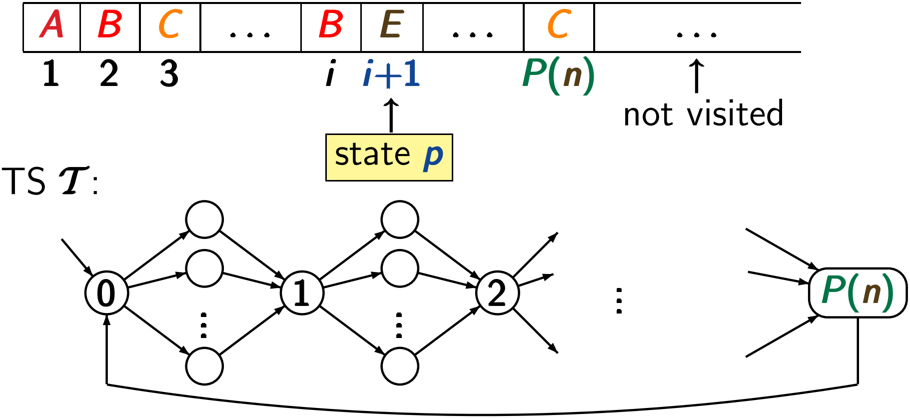
Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle q, D, i\rangle\ i\ \langle *, E, i{+}1\rangle\ \ldots\ P(n)$

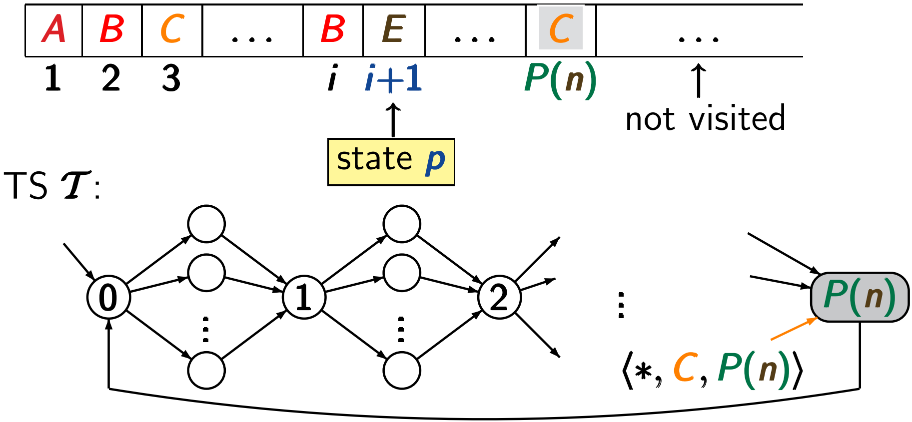
$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle *, B, i\rangle\ i\ \langle p, E, i{+}1\rangle$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$



$$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle q, D, i\rangle\ i\ \langle *, E, i{+}1\rangle\ \ldots\ P(n)$$

$$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle *, B, i\rangle\ i\ \langle p, E, i{+}1\rangle\ \ldots$$



$$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle q, D, i\rangle\ i\ \langle *, E, i{+}1\rangle\ \ldots\ P(n)$$

$$0\ \langle *, A, 1\rangle\ 1\ \langle *, B, 2\rangle\ 2\ \ldots\ \langle *, B, i\rangle\ i\ \langle p, E, i{+}1\rangle\ \ldots\ P(n)$$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M}$ be a DTM with polynomial space bound $P(n)$

- state space Q
- initial state q_0
- set of accept states F
- tape alphabet Γ
- input alphabet $\Sigma \subseteq \Gamma$
- blank symbol $\sqcup$

transition function $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 0, +1\}$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M}$ be a DTM with polynomial space bound $P(n)$

- state space Q
- initial state q_0
- set of accept states F
- tape alphabet Γ
- input alphabet $\Sigma \subseteq \Gamma$
- blank symbol $\sqcup$

transition function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{-1, 0, +1\}$

input word w for $\mathcal{M}$	$\xrightarrow{\text{poly time}}$	TS $\mathcal{T}$ LTL-formula φ
$\mathcal{M}$ accepts w, i.e., $w \in K$	iff	there is path π of $\mathcal{T}$ with $\pi \models \varphi$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w| = n$.

Transition system $\mathcal{T} \stackrel{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w|=n$.

Transition system $\mathcal{T} \overset{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \le i \le P(n)\}$$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w| = n$.

Transition system $\mathcal{T} \stackrel{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \leq i \leq P(n)\}$$

$$S_0 = \{0\}$$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w|=n$.

Transition system $\mathcal{T} \stackrel{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \leq i \leq P(n)\}$$

$S_0 = \{0\}$

$AP = S$ with obvious labeling function

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w|=n$.

Transition system $\mathcal{T} \stackrel{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \leq i \leq P(n)\}$$

$$S_0 = \{0\}$$

$AP = S$ with obvious labeling function

transitions:

$$\left.\begin{array}{l} i{-}1 \longrightarrow \langle q, A, i\rangle \\ \langle q, A, i\rangle \longrightarrow i \end{array}\right\} \begin{array}{l} \text{for } 1 \leq i \leq P(n) \\ \text{and } q \in Q \cup \{*\} \end{array}$$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w|=n$.

Transition system $\mathcal{T} \overset{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \leq i \leq P(n)\}$$

$$S_0 = \{0\}$$

$AP = S$ with obvious labeling function

transitions: $i{-}1 \longrightarrow \langle q, A, i\rangle \qquad P(n) \longrightarrow 0$

$\langle q, A, i\rangle \longrightarrow i$

Polynomial reduction $w \mapsto (\mathcal{T}, \varphi)$

Let $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, \sqcup, F)$ be a DTM with polynomial space bound $P(n)$, and $w \in \Sigma^*$, $|w|=n$.

Transition system $\mathcal{T} \stackrel{\text{def}}{=} (S, Act, \rightarrow, S_0, AP, L)$ where

$$S = \{0, 1, \ldots, P(n)\} \cup \{\langle q, A, i\rangle, \langle *, A, i\rangle : q \in Q, A \in \Gamma, 1 \leq i \leq P(n)\}$$

$$S_0 = \{0\}$$

$AP = S$ with obvious labeling function

transitions: $i{-}1 \longrightarrow \langle q, A, i\rangle$ $\quad P(n) \longrightarrow 0$
$\langle q, A, i\rangle \longrightarrow i$

LTL formula $\varphi \stackrel{\text{def}}{=} \varphi^w_{start} \wedge \varphi_\delta \wedge \varphi_{conf} \wedge \varphi_{accept}$

Complexity of LTL model checking problem

Complexity of LTL model checking problem

We saw that:

The existential **LTL** model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ

question: is there a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

is ***PSPACE***-complete.

Complexity of LTL model checking problem

We saw that:

The existential **LTL** model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ

question: is there a path π in $\mathcal{T}$ with $\pi \models \varphi$ **?**

is ***PSPACE***-complete.

As ***PSPACE*** = ***coPSPACE*** we get:

The **LTL** model checking problem

given: finite TS $\mathcal{T}$, LTL formula φ

question: does $\pi \models \varphi$ hold for all paths π in $\mathcal{T}$ **?**

is ***PSPACE***-complete.

Summary: LTL model checking problem

Summary: LTL model checking problem

The LTL model checking problem is

- solvable by an automata-based approach
  complexity: $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(|\varphi|))$
- *PSPACE*-complete

Summary: LTL model checking problem

The LTL model checking problem is

- solvable by an automata-based approach
  complexity: $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(|\varphi|))$
- ***PSPACE*-complete**

  proof of the lower bound:
  generic reduction from poly-space bounded DTM

  proof of the upper bound:
  uses the LTL-2-GNBA algorithm

Summary: LTL model checking problem

The LTL model checking problem is

- solvable by an automata-based approach
  complexity: $\mathcal{O}(\mathit{size}(\mathcal{T}) \cdot \exp(|\varphi|))$
- ***PSPACE***-complete
  proof of the lower bound:
  generic reduction from poly-space bounded DTM
  proof of the upper bound:
  uses the LTL-2-GNBA algorithm

additionally we proved ***coNP***-hardness
using an LTL-encoding of the Hamilton-path problem

NBA are more powerful than LTL

NBA are more powerful than LTL

There is **no** LTL formula φ over $AP = \{a\}$ s.t.

$$Words(\varphi) = \text{set of words } A_0A_1A_2\ldots \in \left(2^{AP}\right)^{\omega} \text{ s.t. } a \in A_{2i} \text{ for all } i \in \mathbb{N}$$

(without proof)

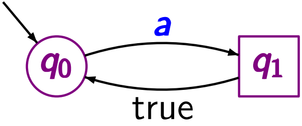
NBA are more powerful than LTL

There is **no** LTL formula φ over $AP = \{a\}$ s.t.

$$Words(\varphi) = \text{set of words } A_0A_1A_2\ldots \in (2^{AP})^\omega \text{ s.t.}$$
$$a \in A_{2i} \text{ for all } i \in \mathbb{N}$$

(without proof)

NBA $\mathcal{A}$:



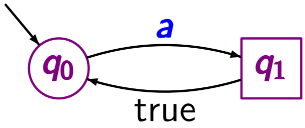
NBA are more powerful than LTL

There is **no** LTL formula φ over $AP = \{a\}$ s.t.

> $Words(\varphi)$ = set of words $A_0 A_1 A_2 \ldots \in (2^{AP})^\omega$ s.t.
> $a \in A_{2i}$ for all $i \in \mathbb{N}$

(without proof)

NBA $\mathcal{A}$:



LTL formula $\varphi \;=\; a \wedge \Box(a \rightarrow \bigcirc\bigcirc a)$?

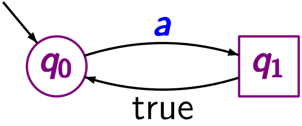
NBA are more powerful than LTL

There is **no** LTL formula φ over $AP = \{a\}$ s.t.

$$Words(\varphi) = \text{set of words } A_0A_1A_2\ldots \in (2^{AP})^\omega \text{ s.t. } a \in A_{2i} \text{ for all } i \in \mathbb{N}$$

(without proof)

NBA $\mathcal{A}$:



LTL formula $\varphi \;=\; a \wedge \Box(a \rightarrow \bigcirc\bigcirc a)$?

$\sigma \;=\; \{a\}\{a\}\{a\}\varnothing\{a\}^\omega \;\not\models\; \varphi$, but $\sigma \in \mathcal{L}_\omega(\mathcal{A})$

LTL satisfiability problem

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable **?**

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $\mathit{Words}(\varphi) \neq \varnothing$?

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

examples:	$\Diamond\Box a \wedge \Box\Diamond\neg a$	unsatisfiable
	$a \,\mathsf{U}\, b \wedge \Box\neg b$	unsatisfiable
	$\Diamond\Box a \wedge a \,\mathsf{U}(\Box b)$	satisfiable

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

check whether $\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

check whether $\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$

nested DFS: check whether $\mathcal{A} \not\models \Diamond\Box\neg F$

LTL satisfiability problem

given: **LTL** formula φ over $\boldsymbol{AP}$

question: is φ satisfiable, i.e., is $\mathit{Words}(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

check whether $\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$

↑

nested DFS: check whether $\mathcal{A} \not\models \Diamond\Box\neg F$

if yes, return "**yes**", otherwise "**no**"

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

check whether $\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$

↑

nested DFS: check whether $\mathcal{A} \not\models \Diamond\Box\neg F$

if yes, return "**yes**", otherwise "**no**"

complexity: $\mathcal{O}(\exp(|\varphi|))$

LTL satisfiability problem

given: **LTL** formula φ over AP

question: is φ satisfiable, i.e., is $Words(\varphi) \neq \varnothing$?

automata-based satisfiability checking algorithm:

construct an NBA $\mathcal{A} = (Q, 2^{AP}, \delta, Q_0, F)$ for φ

check whether $\mathcal{L}_\omega(\mathcal{A}) \neq \varnothing$

↑

nested DFS: check whether $\mathcal{A} \not\models \Diamond\Box\neg F$

if yes, return "**yes**", otherwise "**no**"

complexity: $\mathcal{O}(\exp(|\varphi|))$ ... and *PSPACE*-complete

LTL validity problem

LTL validity problem

given:	**LTL** formula φ over AP
question:	is φ valid, i.e. is $Words(\varphi) = \left(2^{AP}\right)^{\omega}$?

LTL validity problem

given:	**LTL** formula φ over AP
question:	is φ valid, i.e. is $Words(\varphi) = \left(2^{AP}\right)^{\omega}$?

is solvable by a **LTL** satisfiability checker as

φ is valid iff $\neg\varphi$ is not satisfiable

LTL validity problem

given:	**LTL** formula φ over AP
question:	is φ valid, i.e. is $Words(\varphi) = (2^{AP})^\omega$?

is solvable by a **LTL** satisfiability checker as

φ is valid iff $\neg\varphi$ is not satisfiable

complexity: $\mathcal{O}(\exp(|\varphi|))$

LTL validity problem

given:	**LTL** formula φ over AP
question:	is φ valid, i.e. is $Words(\varphi) = \left(2^{AP}\right)^{\omega}$?

is solvable by a **LTL** satisfiability checker as

φ is valid iff $\neg\varphi$ is not satisfiable

complexity: $\mathcal{O}(\exp(|\varphi|))$ ... and *PSPACE*-complete