Modeling and verifying reactive systems

Temporal logics

Nicolas Markey

Lab. Specification et Verification
ENS Cachan & CNRS, France

Outline of the course

1. Linear-time temporal logics
   - Expressiveness of LTL and LTL+Past
   - How hard is LTL verification?
   - Algorithms for verifying LTL formulas

LTL and LTL+Past

Definition

$$\text{LTL} \ni \varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

$$\text{LTL+Past} \ni \varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \mathbf{X}^{-1}\,\varphi \mid \varphi\,\mathbf{S}\,\varphi$$

LTL and LTL+Past

Definition

$$\text{LTL} \ni \varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

$$\text{LTL+Past} \ni \varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \psi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi \mid \mathbf{X}^{-1}\,\varphi \mid \varphi\,\mathbf{S}\,\varphi$$

$\varphi\,\mathbf{U}\,\psi$: $\langle \mathcal{S}, t\rangle \models \mathbf{X}\,\varphi \;\Leftrightarrow\; \exists u > t.\ (\langle \mathcal{S}, u\rangle \models \varphi$ and $\forall v > t.\ (v > u \vee v = u))$

("next" φ)

$\varphi\,\mathbf{U}\,\psi$: $\langle \mathcal{S}, t\rangle \models \varphi\,\mathbf{U}\,\psi \;\Leftrightarrow\; \exists u > t.\ (\langle \mathcal{S}, u\rangle \models \psi$ and $\forall v > t.\ (v < u \Rightarrow \langle \mathcal{S}, v\rangle \models \varphi))$

(φ "until" ψ)

Expressiveness of LTL and LTL+Past

Lemma

LTL and LTL+Past can be translated in first-order logic (involving at most 3 variables).

Expressiveness of LTL and LTL+Past

Lemma

LTL and LTL+Past can be translated in first-order logic (involving at most 3 variables).

Lemma (Kamp (1968) and Gabbay et al. (1980))

First-order logic can be translated in LTL+Past and LTL.

Expressiveness of LTL and LTL+Past

Lemma

LTL and LTL+Past can be translated in first-order logic (involving at most 3 variables).

Lemma (Kamp (1968) and Gabbay et al. (1980))

First-order logic can be translated in LTL+Past and LTL.

Theorem

LTL and LTL+Past are equally expressive.

Expressiveness of LTL and LTL+Past

Lemma

LTL and LTL+Past can be translated in first-order logic (involving at most 3 variables).

Lemma (Kamp (1968) and Gabbay et al. (1980))

First-order logic can be translated in LTL+Past and LTL.

Theorem

LTL and LTL+Past are equally expressive.

Example

$$\mathbf{F}(a \wedge (b \mathbin{\mathbf{U}} c) \mathbin{\mathbf{S}} c) \equiv \ldots$$

Outline of the course

1. Linear-time temporal logics
   - Expressiveness of LTL and LTL+Past
   - How hard is LTL verification?
   - Algorithms for verifying LTL formulas

Hardness of LTL verification

Theorem

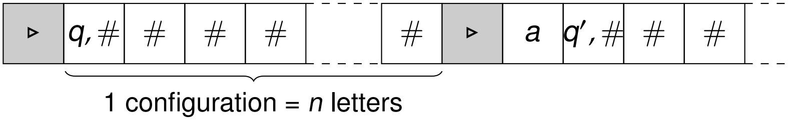
Satisfiability of an LTL formula is PSPACE*-hard.*

Hardness of LTL verification

Theorem

Satisfiability of an LTL formula is PSPACE*-hard.*

Proof. Encode a linear-bounded Turing machine as an LTL formula that is satisfiable if, and only if, the Turing machine halts on the empty input:



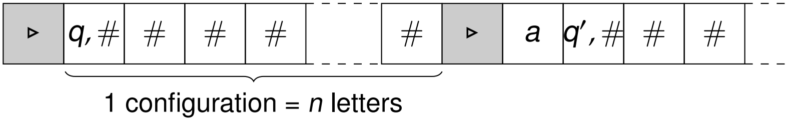
□

Hardness of LTL verification

Theorem

Satisfiability of an LTL formula is PSPACE*-hard.*

Proof. Encode a linear-bounded Turing machine as an LTL formula that is satisfiable if, and only if, the Turing machine halts on the empty input:



□

Corollary

LTL model-checking is PSPACE*-hard.*

Outline of the course

1. Linear-time temporal logics
   - Expressiveness of LTL and LTL+Past
   - How hard is LTL verification?
   - Algorithms for verifying LTL formulas

Büchi automata

Definition

A Büchi automaton is a 5-tuple $\mathcal{B} = \langle Q, Q_0, \Sigma, \rightarrow, F \rangle$ where

- Q is the set of states (or locations) of the automaton,
- $Q_0 \subseteq Q$ is the set of initial states,
- Σ is the alphabet,
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the transition relation,
- $F \subseteq Q$ is the set of repeated states

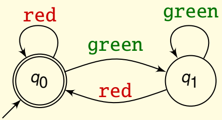
Büchi automata

Definition

A Büchi automaton is a 5-tuple $\mathcal{B} = \langle Q, Q_0, \Sigma, \rightarrow, F \rangle$ where

- Q is the set of states (or locations) of the automaton,
- $Q_0 \subseteq Q$ is the set of initial states,
- Σ is the alphabet,
- $\rightarrow \subseteq Q \times \Sigma \times Q$ is the transition relation,
- $F \subseteq Q$ is the set of repeated states

Example



$$
\begin{array}{rcl}
Q & = & \{q_0, q_1\},\ \ Q_0 = \{q_0\}, \\
\Sigma & = & \{\mathtt{green}, \mathtt{red}\}, \\
\rightarrow & = & \{(q_0, \mathtt{green}, q_1), (q_1, \mathtt{green}, q_1), \\
 & & (q_1, \mathtt{red}, q_0), (q_0, \mathtt{red}, q_0)\}, \\
F & = & \{q_0\}.
\end{array}
$$

Büchi automata

Definition

An (infinite) word $w_0\, w_1$ ... is *accepted* by a Büchi automaton $\mathcal{B}$ if there exists an infinite sequence $\pi = (\ell_0, \ell_1, \ldots)$ of states s.t.:

- $\ell_0 \in Q_0$,
- for each i, $(\ell_i, w_i, \ell_{i+1}) \in \rightarrow$;
- at least one state in F occurs infinitely often in π.

Büchi automata

Definition

An (infinite) word $w_0\, w_1$ ... is *accepted* by a Büchi automaton $\mathcal{B}$ if there exists an infinite sequence $\pi = (\ell_0, \ell_1, \ldots)$ of states s.t.:

- $\ell_0 \in Q_0$,
- for each i, $(\ell_i, w_i, \ell_{i+1}) \in \rightarrow$;
- at least one state in F occurs infinitely often in π.

Büchi automata

Definition

An (infinite) word $w_0\, w_1$ ... is *accepted* by a Büchi automaton $\mathcal{B}$ if there exists an infinite sequence $\pi = (\ell_0, \ell_1, \ldots)$ of states s.t.:

- $\ell_0 \in Q_0$,
- for each i, $(\ell_i, w_i, \ell_{i+1}) \in \rightarrow$;
- at least one state in F occurs infinitely often in π.

We write $\mathcal{L}(\mathcal{B})$ for the set of words accepted by $\mathcal{B}$.

Büchi automata

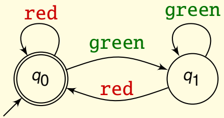
Definition

An (infinite) word $w_0\ w_1$ ... is *accepted* by a Büchi automaton $\mathcal{B}$ if there exists an infinite sequence $\pi = (\ell_0, \ell_1, \ ...)$ of states s.t.:

- $\ell_0 \in Q_0$,
- for each i, $(\ell_i, w_i, \ell_{i+1}) \in \rightarrow$;
- at least one state in F occurs infinitely often in π.

We write $\mathcal{L}(\mathcal{B})$ for the set of words accepted by $\mathcal{B}$.

Example



$$\texttt{green} \cdot \texttt{red}^{\omega} \in \mathcal{L}(\mathcal{B}),$$

$$\texttt{green} \cdot \texttt{red} \cdot \texttt{green}^{\omega} \notin \mathcal{L}(\mathcal{B}).$$

From LTL+Past to Büchi automata

Theorem (Lichtenstein, Pnueli, Zuck, 1985)

Let φ a formula in LTL+Past. There exists a Büchi automaton $\mathcal{B}_\varphi$ s.t.

$$\forall w \in (2^{AP})^\omega. \qquad w \in \mathcal{L}(\mathcal{B}_\varphi) \;\Leftrightarrow\; w, 0 \models \varphi.$$

Sketch of proof.

- each state of the automaton corresponds to a set of subformulas of φ (and negations thereof),
- if a word w is accepted from a location q_0, then any subformula represented by that state holds initially along w.

From LTL+Past to Büchi automata

Theorem (Lichtenstein, Pnueli, Zuck, 1985)

Let φ a formula in LTL+Past. There exists a Büchi automaton $\mathcal{B}_\varphi$ s.t.

$$\forall w \in (2^{AP})^\omega. \qquad w \in \mathcal{L}(\mathcal{B}_\varphi) \;\Leftrightarrow\; w, 0 \models \varphi.$$

Sketch of proof.

- each state of the automaton corresponds to a set of subformulas of φ (and negations thereof),
- if a word w is accepted from a location q_0, then any subformula represented by that state holds initially along w.

From LTL+Past to Büchi automata

Theorem (Lichtenstein, Pnueli, Zuck, 1985)

Let φ a formula in LTL+Past. There exists a Büchi automaton $\mathcal{B}_\varphi$ s.t.

$$\forall w \in (2^{AP})^\omega. \qquad w \in \mathcal{L}(\mathcal{B}_\varphi) \;\Leftrightarrow\; w, 0 \models \varphi.$$

Sketch of proof.

- each state of the automaton corresponds to a set of subformulas of φ (and negations thereof),
- if a word w is accepted from a location q_0, then any subformula represented by that state holds initially along w.

From LTL+Past to Büchi automata

Definition

The closure of φ, denoted by $\mathrm{Cl}(\varphi)$, is the smallest set of formulas containing φ and closed under the following rules:

- $\top$ and $\bot$ are in $\mathrm{Cl}(\varphi)$,
- $\neg\psi \in \mathrm{Cl}(\varphi)$ iff $\psi \in \mathrm{Cl}(\varphi)$ (identifying $\neg\neg\psi$ with ψ),
- if $\psi_1 \wedge \psi_2$ or $\psi_1 \vee \psi_2$ is in $\mathrm{Cl}(\varphi)$, then $\psi_1 \in \mathrm{Cl}(\varphi)$ and $\psi_2 \in \mathrm{Cl}(\varphi)$,
- if $\mathbf{X}\,\psi_1$ is in $\mathrm{Cl}(\varphi)$, then so ψ_1,
- if $\psi_1 \mathbin{\mathbf{U}} \psi_2$ is in $\mathrm{Cl}(\varphi)$, then so are ψ_1, ψ_2, and $\mathbf{X}(\psi_1 \mathbin{\mathbf{U}} \psi_2)$,
- if $\mathbf{X}^{-1}\,\psi_1$ is in $\mathrm{Cl}(\varphi)$, then so ψ_1,
- if $\psi_1 \mathbin{\mathbf{S}} \psi_2$ is in $\mathrm{Cl}(\varphi)$, then so are ψ_1, ψ_2, and $\mathbf{X}^{-1}(\psi_1 \mathbin{\mathbf{S}} \psi_2)$.

From LTL+Past to Büchi automata

Proposition

The size of $\mathrm{Cl}(\varphi)$ *is at most* $4|\varphi|$.

From LTL+Past to Büchi automata

Proposition

The size of $\mathrm{Cl}(\varphi)$ *is at most* $4|\varphi|$.

Proof.

By induction of the structure of φ:

- clear if φ is an atomic formula,

From LTL+Past to Büchi automata

Proposition

The size of $\mathrm{Cl}(\varphi)$ *is at most* $4|\varphi|$.

Proof.

By induction of the structure of φ:

- clear if φ is an atomic formula,
- if $\varphi = \psi_1 \wedge \psi_2$ or $\varphi = \psi_1 \vee \psi_2$, then

$$\mathrm{Cl}(\varphi) = \mathrm{Cl}(\psi_1) \cup \mathrm{Cl}(\psi_2) \cup \{\varphi, \neg \varphi\}.$$

From LTL+Past to Büchi automata

Proposition

The size of $\mathrm{Cl}(\varphi)$ *is at most* $4|\varphi|$.

Proof.
By induction of the structure of φ:

- clear if φ is an atomic formula,
- if $\varphi = \psi_1 \wedge \psi_2$ or $\varphi = \psi_1 \vee \psi_2$, then

$$\mathrm{Cl}(\varphi) = \mathrm{Cl}(\psi_1) \cup \mathrm{Cl}(\psi_2) \cup \{\varphi, \neg\varphi\}.$$

- if $\varphi = \psi_1 \mathbf{U} \psi_2$, then

$$\mathrm{Cl}(\varphi) = \mathrm{Cl}(\psi_1) \cup \mathrm{Cl}(\psi_2) \cup \{\varphi, \neg\varphi, \mathbf{X}\varphi, \neg\mathbf{X}\varphi\}.$$

From LTL+Past to Büchi automata

Proposition

The size of $\mathrm{Cl}(\varphi)$ *is at most* $4|\varphi|$.

Proof.
By induction of the structure of φ:

- clear if φ is an atomic formula,
- if $\varphi = \psi_1 \wedge \psi_2$ or $\varphi = \psi_1 \vee \psi_2$, then

$$\mathrm{Cl}(\varphi) = \mathrm{Cl}(\psi_1) \cup \mathrm{Cl}(\psi_2) \cup \{\varphi, \neg \varphi\}.$$

- if $\varphi = \psi_1 \mathbf{U} \psi_2$, then

$$\mathrm{Cl}(\varphi) = \mathrm{Cl}(\psi_1) \cup \mathrm{Cl}(\psi_2) \cup \{\varphi, \neg \varphi, \mathbf{X} \varphi, \neg \mathbf{X} \varphi\}.$$

- the other cases are similar.

From LTL+Past to Büchi automata

Example

Consider formula $\varphi = \mathbf{G}(\mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}))$.
Then:

$$
\begin{aligned}
\mathrm{Cl}(\varphi) = \{\varphi, \neg\varphi,\ & \\
& \mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}), \\
& \neg(\mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green})), \\
& \mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}, \\
& \neg(\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}), \\
& \mathbf{F}\,\mathtt{red}, \neg\,\mathbf{F}\,\mathtt{red}, \mathbf{X}\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{X}\,\mathbf{F}\,\mathtt{red}, \\
& \mathbf{G}^{-1}\,\mathtt{green}, \neg\,\mathbf{G}^{-1}\,\mathtt{green}, \\
& \mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\mathtt{green}, \neg\,\mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\mathtt{green}, \\
& \mathtt{green}, \neg\,\mathtt{green}, \mathtt{red}, \neg\,\mathtt{red}, \top, \bot\}.
\end{aligned}
$$

From LTL+Past to Büchi automata

Definition

A subset S of $\mathrm{Cl}(\varphi)$ is *maximal consistent* if:

- $\top \in S$,
- for any $\psi \in \mathrm{Cl}(\varphi)$, $\psi \in S$ iff $\neg\psi \notin S$,
- for any $\psi = \psi_1 \wedge \psi_2 \in \mathrm{Cl}(\varphi)$: $\psi \in S$ iff $\psi_1 \in S$ and $\psi_2 \in S$,
- for any $\psi = \psi_1 \vee \psi_2 \in \mathrm{Cl}(\varphi)$: $\psi \in S$ iff $\psi_1 \in S$ or $\psi_2 \in S$,
- for any $\psi = \psi_1 \mathbf{U} \psi_2 \in \mathrm{Cl}(\varphi)$:
  $\psi \in S$ iff $\psi_2 \in S$, or both ψ_1 and $\mathbf{X}(\psi_1 \mathbf{U} \psi_2)$ are in S,
- for any $\psi = \psi_1 \mathbf{S} \psi_2 \in \mathrm{Cl}(\varphi)$:
  $\psi \in S$ iff $\psi_2 \in S$, or both ψ_1 and $\mathbf{X}^{-1}(\psi_1 \mathbf{S} \psi_2)$ are in S.

From LTL+Past to Büchi automata

Example

The set

$$\{\varphi, \neg(\mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green})),$$
$$\neg(\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}),$$
$$\neg\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{X}\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{G}^{-1}\,\mathtt{green}, \neg\,\mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\mathtt{green},$$
$$\mathtt{green}, \neg\,\mathtt{red}\}$$

is maximal consistent.

From LTL+Past to Büchi automata

Example

The set

$$\{\varphi, \neg(\mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green})),$$
$$\neg(\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}),$$
$$\neg\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{X}\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{G}^{-1}\,\mathtt{green}, \neg\,\mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\mathtt{green},$$
$$\mathtt{green}, \neg\,\mathtt{red}\}$$

is maximal consistent.

Proposition

There are at most $2^{4|\varphi|}$ maximal consistent subsets of $\mathrm{Cl}(\varphi)$.

From LTL+Past to Büchi automata

Example

The set

$$\{\varphi, \neg(\mathtt{green} \Rightarrow (\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green})),$$
$$\neg(\mathbf{F}\,\mathtt{red} \vee \mathbf{G}^{-1}\,\mathtt{green}),$$
$$\neg\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{X}\,\mathbf{F}\,\mathtt{red}, \neg\,\mathbf{G}^{-1}\,\mathtt{green}, \neg\,\mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\mathtt{green},$$
$$\mathtt{green}, \neg\,\mathtt{red}\}$$

is maximal consistent.

Proposition

There are at most $2^{4|\varphi|}$ maximal consistent subsets of $\mathrm{Cl}(\varphi)$.

Maximal consistent subsets are the states of our Büchi automaton.

From LTL+Past to Büchi automata

Given two maximal consistent subsets S and T of $\mathrm{Cl}(\varphi)$, and a "letter" $\sigma \subseteq \mathrm{AP}$, there is a transition (S, σ, T) iff:

- for any $p \in \mathrm{AP}$, we have $p \in S$ iff $p \in \sigma$,
- for any subformula $\mathbf{X}\,\varphi_1 \in \mathrm{Cl}(\varphi)$:
  $$\mathbf{X}\,\varphi_1 \text{ is in } S \quad \text{iff} \quad \varphi_1 \in T,$$
- for any subformula $\mathbf{X}^{-1}\,\varphi_1 \in \mathrm{Cl}(\varphi)$:
  $$\varphi_1 \text{ is in } S \quad \text{iff} \quad \mathbf{X}^{-1}\,\varphi_1 \in T.$$

From LTL+Past to Büchi automata

Given two maximal consistent subsets S and T of $\mathrm{Cl}(\varphi)$, and a "letter" $\sigma \subseteq \mathsf{AP}$, there is a transition (S, σ, T) iff:

- for any $p \in \mathsf{AP}$, we have $p \in S$ iff $p \in \sigma$,
- for any subformula $\mathbf{X}\,\varphi_1 \in \mathrm{Cl}(\varphi)$:
$$\mathbf{X}\,\varphi_1 \text{ is in } S \quad \text{iff} \quad \varphi_1 \in T,$$
- for any subformula $\mathbf{X}^{-1}\,\varphi_1 \in \mathrm{Cl}(\varphi)$:
$$\varphi_1 \text{ is in } S \quad \text{iff} \quad \mathbf{X}^{-1}\,\varphi_1 \in T.$$

Example

S:
...
$\neg\, \mathbf{X}\,\mathbf{F}\,\texttt{red}$
$\neg\, \mathbf{G}^{-1}\,\texttt{green}$
$\texttt{green}$

$\longrightarrow$

T:
...
$\neg\, \mathbf{F}\,\texttt{red}$
$\neg\, \mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\texttt{green}$
$\neg\,\texttt{red}$

From LTL+Past to Büchi automata

We use (generalized) Büchi acceptance condition is used to enforce that eventualities eventually occur:

- For each subformula $\psi = \varphi_1 \ \mathbf{U} \ \varphi_2$, we write

$$F_\psi = \{I \in Q \mid \varphi_2 \in I \text{ or } \psi \in I\}$$

- a word is accepted if it has a trajectory whose repeated states intersect F_ψ for each **U**-subformula ψ.

From LTL+Past to Büchi automata

We use (generalized) Büchi acceptance condition is used to enforce that eventualities eventually occur:

- For each subformula $\psi = \varphi_1 \ \mathbf{U} \ \varphi_2$, we write

$$F_\psi = \{I \in Q \mid \varphi_2 \in I \text{ or } \psi \in I\}$$

- a word is accepted if it has a trajectory whose repeated states intersect F_ψ for each **U**-subformula ψ.

- initial states are those where all $\mathbf{X}^{-1}$-subformulas are false.

From LTL+Past to Büchi automata

We use (generalized) Büchi acceptance condition is used to enforce that eventualities eventually occur:

- For each subformula $\psi = \varphi_1 \mathbf{U} \varphi_2$, we write

$$F_\psi = \{I \in Q \mid \varphi_2 \in I \text{ or } \psi \in I\}$$

- a word is accepted if it has a trajectory whose repeated states intersect F_ψ for each **U**-subformula ψ.

- initial states are those where all $\mathbf{X}^{-1}$-subformulas are false.

Lemma

A word is accepted by this automaton if, and only if, it satisfies the initial LTL+Past formula.

From LTL+Past to Büchi automata

Theorem

For any LTL+Past formula φ, there exists a generalized Büchi automaton $\mathcal{A}$ s.t.

- *a word is accepted by $\mathcal{A}$ iff if satisfies φ;*
- *$\mathcal{A}$ has at most $2^{4|\varphi|}$ states.*

From LTL+Past to Büchi automata

Theorem

For any LTL+Past formula φ, there exists a generalized Büchi automaton $\mathcal{A}$ s.t.

- *a word is accepted by $\mathcal{A}$ iff if satisfies φ;*
- *$\mathcal{A}$ has at most $2^{4|\varphi|}$ states.*

Proposition

A generalized Büchi automaton $\mathcal{A}$ can be transformed in a (standard) Büchi automaton $\mathcal{B}$ s.t.

- *$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$,*
- *$|\mathcal{B}| \leq |\mathcal{A}|^2$.*

An algorithm for LTL+Past satisfiability

Theorem

LTL+Past satisfiability can be achieved in PSPACE.

Proof.

- we use the translation to Büchi automata, but not directly, as it would require exponential space...
- the algorithm non-deterministically guesses the accepting path as follows:
  - guess and store one repeated state;
  - guess, step by step, a path from an initial state to the repeated state;
  - guess, step by step, a path from the repeated state to itself.

  Each time, only a polynomial amount of information has to be stored. This algorithm is thus in PSPACE.

$\square$

An algorithm for LTL+Past model-checking

Theorem

LTL+Past model-checking can be achieved in PSPACE.

Proof.

- a Kripke structure can be seen as an automaton: it suffices to label each transition with the set of atomic propositions that hold in its source state;
- it then suffices to compute the product of this automaton with the automaton $\mathcal{A}_{\neg\varphi}$: the language of the resulting automaton is empty if, and only if, all the paths in the original Kripke structure satisfy formula φ.

$\square$