Expressiveness of Temporal Logics

François Laroussinie and Nicolas Markey

Lab. Specification et Verification
ENS Cachan & CNRS, France

August 1, 2006
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus
2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past
3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle
4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Introduction to the second lecture

- In this second course, we focus on discrete time.

- as is usual in that case, we use the following definitions:
  - $LTL = \mathcal{L}(U, X)$,
  - $LTL + \text{Past} = \mathcal{L}(U, S, X, X^{-1})$.

where $U$ and $S$ have their *non-strict* meaning, e.g.:

\[
\varphi U \psi \text{ holds iff }
\begin{align*}
&\text{either } \psi \text{ holds,} \\
&\text{or } \varphi \text{ holds, and } \varphi U \psi \text{ holds in the next location,}
\end{align*}
\]

In other words, we have the following equivalence:

\[
\varphi U \psi \equiv \psi \lor (\varphi \land X(\varphi U \psi)).
\]

- linear structures are seen as (infinite) words over the alphabet $2^{AP}$. 

Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus
2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past
3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle
4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
LTL+Past and the $\mu$-calculus

Definition

The *linear-time $\mu$-calculus* is an extension of $L(X)$ with fixpoint operators:

$$\mu\text{-calculus} \ni \phi, \psi ::= \top \mid p \mid \neg p \mid Z \mid \phi \lor \psi \mid \phi \land \psi \mid X \phi \mid X^{-1} \phi \mid \mu Z \phi \mid \nu Z \phi$$

where $p$ ranges over AP and $Z$ ranges over a finite set of variables.
LTL+Past and the $\mu$-calculus

**Definition**

The *linear-time $\mu$-calculus* is an extension of $\mathcal{L}(X)$ with fixpoint operators:

\[
\mu\text{-calculus} \ni \varphi, \psi ::= \top \mid p \mid \neg p \mid Z \mid \varphi \lor \psi \mid \varphi \land \psi \mid X\psi \mid X^{-1}\psi \mid \mu Z \varphi \mid \nu Z \varphi
\]

where $p$ ranges over AP and $Z$ ranges over a finite set of variables.

**Example**

\[
\mu Z (\text{green} \lor XZ)
\]
LTL++Past and the $\mu$-calculus

Theorem (Knaster, 1928 & Tarski, 1955)

Let $T$ be a linear structure. Given a formula $\varphi(Z)$, the set of positions satisfying $\mu Z \varphi(Z)$ is

$$\bigcap \{U \subseteq T \mid \varphi(U) \subseteq U\}$$
LTL+Past and the $\mu$-calculus

**Theorem (Knaster, 1928 & Tarski, 1955)**

Let $T$ be a linear structure. Given a formula $\varphi(Z)$, the set of positions satisfying $\mu Z \varphi(Z)$ is

$$\bigcap \{U \subseteq T \mid \varphi(U) \subseteq U\}$$

Moreover, fixpoints can be computed iteratively:

**Theorem (Knaster, 1928 & Tarski, 1955)**

The set of positions satisfying $\mu Z \varphi(Z)$ is the limit of the following sequence:

$$\llbracket \mu Z \varphi(Z) \rrbracket_0 = \emptyset$$

$$\llbracket \mu Z \varphi(Z) \rrbracket_{i+1} = \{t \in T \mid \langle T, t \rangle \models \varphi(\llbracket \mu Z \varphi(Z) \rrbracket_i)\}$$
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$$T = \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{green}{g} \textcolor{green}{g} \textcolor{red}{b} \textcolor{red}{r} \textcolor{red}{b} \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} ...$$

and the following formula:

$$\mu Z (\textcolor{green}{\text{green}} \lor \textcolor{red}{X} Z).$$
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$$T = g \ r \ r \ g \ g \ b \ r \ b \ g \ r \ r \ g \ r \ r \ r \ r \ r \ r \ r \ r \ r \ r \ r \ r \ r \ ...$$

and the following formula:

$$\mu Z \ (\text{green} \lor X Z).$$

Then:

$$\llbracket \mu Z \ (g \lor X Z) \rrbracket_0 = \emptyset$$
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$$T = \text{g r r g g b r b g r r g r r r r r r r r ...}$$

and the following formula:

$$\mu Z (\text{green} \lor \mathbf{X} Z).$$

Then:

$$\mu Z (\text{g} \lor \mathbf{X} Z) \equiv_0 \bot$$
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$$T = \textbf{g} \textbf{r} \textbf{r} \textbf{g} \textbf{g} \textbf{b} \textbf{r} \textbf{b} \textbf{g} \textbf{r} \textbf{r} \textbf{g} \textbf{r} \textbf{r} \textbf{r} \textbf{r} \textbf{r} \textbf{r} \textbf{r} \ldots$$

and the following formula:

$$\mu Z (\text{green} \lor X Z).$$

Then:

$$\llbracket \mu Z (g \lor X Z) \rrbracket_1 = \{ t \in T \mid \langle T, t \rangle \models g \lor X \bot \}$$
Example

Consider the following linear structure (represented as a word):

\[ T = \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{green}{g} \textcolor{green}{g} \textcolor{blue}{b} \textcolor{red}{r} \textcolor{blue}{b} \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{green}{g} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \textcolor{red}{r} \ldots \]

and the following formula:

\[ \mu Z \left( \textcolor{green}{\text{green}} \lor X Z \right). \]

Then:

\[ \mu Z \left( g \lor X Z \right) \equiv_1 g \]
Example

Consider the following linear structure (represented as a word):

\[ T = \text{g r r g g b r b g r r g r r r r r r ...} \]

and the following formula:

\[ \mu Z (\text{green} \lor X Z). \]

Then:

\[ \llbracket \mu Z (g \lor X Z) \rrbracket_2 = \{ t \in T \mid \langle T, t \rangle \models g \lor X g \} \]
Example

Consider the following linear structure (represented as a word):

\[ T = \text{g} \text{r} \text{r} \text{g} \text{g} \text{b} \text{r} \text{b} \text{g} \text{r} \text{r} \text{g} \text{r} \text{r} \text{r} \text{r} \text{r} \text{r} \text{r} \ldots \]

and the following formula:

\[ \mu Z (\text{green} \lor X Z) \]

Then:

\[ \mu Z (g \lor X Z) \equiv_2 g \lor X g \]
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$$T = \text{g r r g g b r b b g g r r g r r r r r r ...}$$

and the following formula:

$$\mu Z (\text{green } \lor X Z).$$

Then:

$$\mu Z (g \lor X Z) \equiv_n g \lor X g \lor X X g \lor ... \lor X^{n-1} g$$
LTL+Past and the $\mu$-calculus

Example

Consider the following linear structure (represented as a word):

$T = \text{g r r g g b r b b g r r g g r r r r r r r r ...}$

and the following formula:

$\mu Z (\text{green } \lor X Z)$.

Then:

$\mu Z (g \lor X Z) \equiv F g$
LTL+Past and the $\mu$-calculus

$$\varphi \mathbf{U} \psi \equiv \psi \lor (\varphi \land \mathbf{X}(\varphi \mathbf{U} \psi)).$$

From this equivalence, we get:

$$\varphi \mathbf{U} \psi \equiv \mu Z (\psi \lor (\varphi \land \mathbf{X} Z))$$
LTL+Past and the \( \mu \)-calculus

\[ \varphi \mathbf{U} \psi \equiv \psi \lor (\varphi \land \mathbf{X}(\varphi \mathbf{U} \psi)) \].

From this equivalence, we get:

\[ \varphi \mathbf{U} \psi \equiv \mu Z (\psi \lor (\varphi \land \mathbf{X} Z)) \]

Thus:

**Proposition**

\( \mu \)-calculus is at least as expressive as LTL+Past.
LTL+Past and the $\mu$-calculus

In fact:

Proposition

$\mu$-calculus is strictly more expressive than LTL+Past.
LTL+Past and the $\mu$-calculus

In fact:

**Proposition**

$\mu$-calculus is strictly more expressive than LTL+Past.

*Proof.*

The negation of

"green occurs at every even position"

is

"$\neg$green occurs at some even position"

i.e.

$\neg$green $\lor$ $\mathbf{XX}$ $\neg$green $\lor$ $\mathbf{XXXX}$ $\neg$green $\lor$ ...
LTL+Past and the $\mu$-calculus

In fact:

**Proposition**

$\mu$-calculus is strictly more expressive than LTL+Past.

**Proof.**

The negation of

“green occurs at every even position”

is

“$\neg$ green occurs at some even position”

i.e.

$\neg$ green $\lor$ XX $\neg$ green $\lor$ XXXX $\neg$ green $\lor$ ...

It can be written as

$\mu Z (\neg$ green $\lor$ XX Z)
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
LTL+Past and Büchi automata

Finite-state automata are a powerful formalism for defining languages.

Example

G(green ⇒ F red)
LTL+Past and Büchi automata

Finite-state automata are a powerful formalism for defining languages.

Example

\[ G(\text{green} \Rightarrow F \text{red}) \]
LTL+Past and Büchi automata

Finite-state automata are a powerful formalism for defining languages.

Example

What is the relationship between automata (on words) and (linear-time) temporal logics?
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,
- \( \Sigma \) 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 $B = \langle Q, Q_0, \Sigma, \rightarrow, F \rangle$ where

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

Example

\[
\begin{align*}
Q &= \{q_0, q_1\}, \quad Q_0 = \{q_0\}, \\
\Sigma &= \{\text{green, red}\}, \\
\rightarrow &= \{(q_0, \text{green, } q_1), (q_1, \text{green, } q_1), \\
&\quad (q_1, \text{red, } q_0), (q_0, \text{red, } q_0)\}, \\
F &= \{q_0\}.
\end{align*}
\]
**Büchi automata**

**Definition**

An (infinite) word $w_0 \, w_1 \, ...$ is **accepted** by a Büchi automaton $B$ 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 $\pi$. 

<table>
<thead>
<tr>
<th>State Condition</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell_0 \in Q_0$</td>
<td>$\ell_0$ starts in $Q_0$</td>
</tr>
<tr>
<td>For each $i$, $(\ell_i, w_i, \ell_{i+1}) \in \rightarrow$</td>
<td>Transitions are defined</td>
</tr>
<tr>
<td>At least one state in $F$ occurs infinitely often in $\pi$</td>
<td>Infinitely many states in $F$ are visited</td>
</tr>
</tbody>
</table>
Büchi automata

Definition

An (infinite) word $w_0 w_1 \ldots$ is \textit{accepted} by a Büchi automaton $B$ 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 $\pi$. 
Büchi automata

Definition

An (infinite) word $w_0 w_1 \ldots$ is accepted by a Büchi automaton $B$ 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 $\pi$.

We write $L(B)$ for the set of words accepted by $B$. 
**Büchi automata**

**Definition**

An (infinite) word $w_0 \ w_1 \ \ldots$ is **accepted** by a Büchi automaton $B$ 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 $\pi$.

We write $L(B)$ for the set of words accepted by $B$.

**Example**

$$
\begin{aligned}
\text{green} \cdot \text{red}^\omega &\in L(B), \\
\text{green} \cdot \text{red} \cdot \text{green}^\omega &\notin L(B).
\end{aligned}
$$
Outline of today’s lecture

1. **LTL+Past and the \( \mu \)-calculus**

2. **LTL+Past and Büchi automata**
   - Büchi automata
   - From LTL+Past to Büchi automata
     - Büchi automata are more expressive
     - Alternating Büchi automata
     - Application: Succinctness of LTL+Past

3. **Stuttering**
   - The stuttering principle
   - The generalized stuttering principle

4. **Ehrenfeucht-Fraïssé games**
   - The rules of the game
   - EF games and the Until-Since hierarchy
Theorem (Lichtenstein, Pnueli, Zuck, 1985)

Let \( \varphi \) a formula in LTL+Past. There exists a Büchi automaton \( B_\varphi \) s.t.

\[
\forall w \in (2^{AP})^\omega. \quad w \in \mathcal{L}(B_\varphi) \iff w, 0 \models \varphi.
\]

Sketch of proof.

- each state of the automaton corresponds to a set of subformulas of \( \varphi \) (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 \( \varphi \) a formula in LTL+Past. There exists a Büchi automaton \( B_\varphi \) s.t.

\[
\forall w \in (2^{AP})^\omega. \quad w \in L(B_\varphi) \iff w, 0 \models \varphi.
\]

**Sketch of proof.**

- each state of the automaton corresponds to a set of subformulas of \( \varphi \) (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 $\varphi$ a formula in LTL+Past. There exists a Büchi automaton $B_{\varphi}$ s.t.
\[ \forall w \in (2^AP)^\omega. \quad w \in \mathcal{L}(B_{\varphi}) \iff w, 0 \models \varphi. \]

Sketch of proof.

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

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

- \( \top \) and \( \bot \) are in \( \text{Cl}(\varphi) \),
- \( \neg \psi \in \text{Cl}(\varphi) \) iff \( \psi \in \text{Cl}(\varphi) \) (identifying \( \neg \neg \psi \) with \( \psi \)),
- if \( \psi_1 \land \psi_2 \) or \( \psi_1 \lor \psi_2 \) is in \( \text{Cl}(\varphi) \), then \( \psi_1 \in \text{Cl}(\varphi) \) and \( \psi_2 \in \text{Cl}(\varphi) \),
- if \( X \psi_1 \) is in \( \text{Cl}(\varphi) \), then so \( \psi_1 \),
- if \( \psi_1 U \psi_2 \) is in \( \text{Cl}(\varphi) \), then so are \( \psi_1, \psi_2, \) and \( X(\psi_1 U \psi_2) \),
- if \( X^{-1} \psi_1 \) is in \( \text{Cl}(\varphi) \), then so \( \psi_1 \),
- if \( \psi_1 S \psi_2 \) is in \( \text{Cl}(\varphi) \), then so are \( \psi_1, \psi_2, \) and \( X^{-1}(\psi_1 S \psi_2) \).
Proposition

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

The size of $\mathcal{C}_I(\varphi)$ is at most $4 |\varphi|$. 

Proof.
By induction of the structure of $\varphi$:
- clear if $\varphi$ is an atomic formula,
Proposition

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

Proof.
By induction of the structure of $\varphi$:

- clear if $\varphi$ is an atomic formula,
- if $\varphi = \psi_1 \land \psi_2$ or $\varphi = \psi_1 \lor \psi_2$, then

  $\text{Cl}(\varphi) = \text{Cl}(\psi_1) \cup \text{Cl}(\psi_2) \cup \{\varphi, \neg \varphi\}$. 
From LTL+Past to Büchi automata

**Proposition**

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

**Proof.**

By induction of the structure of \( \varphi \):

- clear if \( \varphi \) is an atomic formula,
- if \( \varphi = \psi_1 \land \psi_2 \) or \( \varphi = \psi_1 \lor \psi_2 \), then
  \[
  \text{Cl}(\varphi) = \text{Cl}(\psi_1) \cup \text{Cl}(\psi_2) \cup \{\varphi, \neg \varphi\}.
  \]
- if \( \varphi = \psi_1 \mathbf{U} \psi_2 \), then
  \[
  \text{Cl}(\varphi) = \text{Cl}(\psi_1) \cup \text{Cl}(\psi_2) \cup \{\varphi, \neg \varphi, \mathbf{X} \varphi, \neg \mathbf{X} \varphi\}.
  \]
Proposition

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

Proof.

By induction of the structure of $\varphi$:

- Clear if $\varphi$ is an atomic formula,
- If $\varphi = \psi_1 \land \psi_2$ or $\varphi = \psi_1 \lor \psi_2$, then

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

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

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

- The other cases are similar.
From LTL+Past to Büchi automata

Example

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

$$Cl(\varphi) = \{\varphi, \neg \varphi, \quad \text{green} \Rightarrow (F\ red \lor G^{-1}\ green), \quad \neg (green \Rightarrow (F\ red \lor G^{-1}\ green)), \quad F\ red \lor G^{-1}\ green, \quad \neg (F\ red \lor G^{-1}\ green), \quad F\ red, \neg F\ red, X\ F\ red, \neg X\ F\ red, \quad G^{-1}\ green, \neg G^{-1}\ green, \quad X^{-1}\ G^{-1}\ green, \neg X^{-1}\ G^{-1}\ green, \quad \text{green}, \neg \text{green}, \text{red}, \neg \text{red}, \top, \bot \}.$$
From LTL+Past to Büchi automata

Definition

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

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

The set

\[
\{ \varphi, \neg (\text{green} \Rightarrow (F \text{ red} \lor G^{-1} \text{ green})), \\
\neg (F \text{ red} \lor G^{-1} \text{ green}), \\
\neg F \text{ red}, \neg X F \text{ red}, \neg G^{-1} \text{ green}, \neg X^{-1} G^{-1} \text{ green}, \\
\text{green}, \neg \text{red} \}
\]

is maximal consistent.
From LTL+Past to Büchi automata

Example

The set

\[
\{ \varphi, \neg (\text{green} \Rightarrow (F \text{red} \lor G^{-1} \text{green})), \\
\neg (F \text{red} \lor G^{-1} \text{green}), \\
\neg F \text{red}, \neg X F \text{red}, \neg G^{-1} \text{green}, \neg X^{-1} G^{-1} \text{green}, \\
\text{green}, \neg \text{red} \}
\]

is maximal consistent.

Proposition

There are at most \(2^{4|\varphi|}\) maximal consistent subsets of \(\text{Cl}(\varphi)\).
From LTL+Past to Büchi automata

Example

The set

\{\varphi, \neg (\text{green} \Rightarrow (F \text{ red} \lor G^{-1} \text{ green})),\}
\neg (F \text{ red} \lor G^{-1} \text{ green}),
\neg F \text{ red}, \neg X F \text{ red}, \neg G^{-1} \text{ green}, \neg X^{-1} G^{-1} \text{ green},
green, \neg \text{ red}\}

is maximal consistent.

Proposition

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

Maximal consistent subsets are the states of our Büchi automaton.
Given two maximal consistent subsets \( S \) and \( T \) of \( \text{Cl}(\varphi) \), and a “letter” \( \sigma \subseteq \text{AP} \), there is a transition \((S, \sigma, T)\) iff:

- for any \( p \in \text{AP} \), we have \( p \in S \) iff \( p \in \sigma \),
- for any subformula \( X \varphi_1 \in \text{Cl}(\varphi) \):
  \[ X \varphi_1 \text{ is in } S \text{ iff } \varphi_1 \in T, \]
- for any subformula \( X^{-1} \varphi_1 \in \text{Cl}(\varphi) \):
  \[ \varphi_1 \text{ is in } S \text{ iff } X^{-1} \varphi_1 \in T. \]
From LTL+Past to Büchi automata

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

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

**Example**

\[ S: \]
\[
\begin{align*}
  \ldots \\
  \neg X \, F \, \text{red} \\
  \neg G^{-1} \, \text{green} \\
  \text{green}
\end{align*}
\]

\[ T: \]
\[
\begin{align*}
  \ldots \\
  \neg F \, \text{red} \\
  \neg X^{-1} \, G^{-1} \, \text{green} \\
  \neg \text{red}
\end{align*}
\]
Example

For formula $G(\text{green} \Rightarrow (F \text{red} \lor G^{-1}\text{green}))$, we get (only "useful" states are displayed):
From LTL+Past to Büchi automata

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

- For each subformula $\psi = \varphi_1 U \varphi_2$, we write
  $$F_\psi = \{ l \in Q \mid \varphi_2 \in l \lor \psi \in l \}$$

- a word is accepted if it has a trajectory whose repeated states intersect $F_\psi$ for each $U$-subformula $\psi$. 
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
  
  $$F_\psi = \{ l \in Q \mid \varphi_2 \in l \text{ or } \psi \in l \}$$

- A word is accepted if it has a trajectory whose repeated states intersect $F_\psi$ for each $\mathbf{U}$-subformula $\psi$.

- Initial states are those where all $\mathbf{X}^{-1}$-subformulas are false.
From LTL+Past to Büchi automata

Example

For formula $G(\text{green } \Rightarrow (F \text{ red } \lor G^{-1} \text{ green}))$, we get:
From LTL+Past to Büchi automata

Theorem

For any LTL+Past formula \( \varphi \), there exists a Büchi automaton \( A \) s.t.
- a word is accepted by \( A \) iff if satisfies \( \varphi \);
- \( A \) has at most \( 2^{4|\varphi|} \) states.

This result is extremely important in computer science: it provides a nice way of verifying that an automaton satisfies an LTL+Past formula. In particular:

Theorem

Satisfiability of an LTL+Past formula is PSPACE-complete.
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Büchi automata are more expressive

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Büchi automata are strictly more expressive than $\text{LTL}+\text{Past}$.</td>
</tr>
</tbody>
</table>
Büchi automata are more expressive

**Theorem**

*Büchi automata are strictly more expressive than LTL+Past.*

**Proof.**

The following Büchi automaton accepts words where *green* holds (at least) at even positions:

![Automaton Diagram](image)
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example

\[\text{\begin{tikzpicture}[->,>=stealth',auto,node distance=2cm,every state/.style={thick},shorten >=0pt]
  \node[initial,state] (1) {1};
  \node[state] (2) [right of=1] {2};
  \node[state] (3) [below of=1] {3};
  \node[state] (4) [right of=3] {4};

  \path
  (1) edge node {green} (2)
  (1) edge node {red} (3)
  (1) edge [loop above] node {\ast} (1)
  (2) edge node {\ast} (3)
  (3) edge node {blue} (4)
  (4) edge [loop below] node {\ast} (4);
\end{tikzpicture}}\]
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example

```
1 ←→ 2 ←→ 4 ←→ 3
1 ←→ 2 ←→ 4 ←→ 3
```

```
1  2  1  2  2
3  4  3  4  1
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```

```
1  2  1  2  2
3  2  3  2  2
```
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example

```
1 — green — 2 — * — 3
| | red | | blue |
| |    |    |      |
| v blue | v |
3 — * — 4
```

```
1 — green — 1 — green — 1 — red — 2
| | green | | green | | red |
| |      | |      | |    |
| v blue | v blue |
3 — red — 2 — blue — 2 — 2
```
Alternating Büchi automata

Alternation

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

Example
Alternating Büchi automata

**Alternation**

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

**Example**

![Diagram of an alternating Büchi automaton]

- States: 1, 2, 3, 4
- Transitions:
  - From 1 to 2: green
  - From 1 to 3: red
  - From 2 to 3: blue
  - From 3 to 4: blue
  - From 4 to 3: blue
  - From 1 to 1: green
  - From 2 to 2: red
  - From 3 to 3: red
  - From 4 to 4: blue
Alternating Büchi automata

**Alternation**

Alternating automata are automata in which non-deterministic “choices” can be both disjunctive (as usual) and conjunctive.

**Example**

![Diagram of Alternating Büchi automata]
Alternating Büchi automata

Definition

An alternating automaton is 1-weak if there exists a total order on its set of states such that transitions are always “decreasing”.
Alternating Büchi automata

**Definition**
An alternating automaton is *1-weak* if there exists a total order on its set of states such that transitions are always “decreasing”.

**Example**

![Diagram of an alternating Büchi automaton](image-url)
Alternating Büchi automata

Definition
An alternating automaton is \textit{1-weak} if there exists a total order on its set of states such that transitions are always “decreasing”.

Example
\[
\begin{array}{c}
\text{1} \rightarrow \text{2} \\
\text{4} \rightarrow \text{3}
\end{array}
\]
\[
\begin{array}{c}
\text{red} \\
\text{green} \\
\text{blue} \\
\text{F blue}
\end{array}
\]
\[
(\text{green} \land \text{F blue}) \cup \text{red}
\]
Alternating Büchi automata

**Theorem (Vardi, 1994)**

Any LTL formula $\varphi$ can be transformed into a 1-weak alternating Büchi automaton $B_\varphi$ s.t.

- a word is accepted by $B_\varphi$ iff it satisfies $\varphi$,
- $B_\varphi$ has at most $|\varphi|$ states.
### Alternating Büchi automata

#### Theorem (Vardi, 1994)

Any LTL formula $\varphi$ can be transformed into a 1-weak alternating Büchi automaton $B_\varphi$ s.t.
- a word is accepted by $B_\varphi$ iff it satisfies $\varphi$,
- $B_\varphi$ has at most $|\varphi|$ states.

Conversely:

#### Theorem (Rohde, 1997)

A 1-weak alternating Büchi automaton $B$ can be transformed into an LTL formula $\varphi_B$ s.t.
- a word is accepted by $\varphi$ iff it satisfies $\varphi_B$,
- $\varphi_B$ has size exponential in the size of $B$. 
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Succinctness of LTL+Past

Consider the following property, built on $\text{AP} = \{p_0, \ldots, p_n\}$:

$(\mathcal{P})$: any two states that agree on propositions $p_1$ to $p_n$ also agree on proposition $p_0$. 

Succinctness of LTL+Past

Consider the following property, built on \( \text{AP} = \{ p_0, \ldots, p_n \} \):

\((P)\): any two states that agree on propositions \( p_1 \) to \( p_n \) also agree on proposition \( p_0 \).

It can be expressed in LTL by enumerating the possible valuations for \( p_0 \) to \( p_n \):

\[
\bigwedge_{(b_0, \ldots, b_n) \in \{ \top, \bot \}^{n+1}} \left( \mathbf{F} \left( \bigwedge_{i \geq 0} p_i = b_i \right) \Rightarrow \mathbf{G} \left( \left( \bigwedge_{i \geq 1} p_i = b_i \right) \Rightarrow p_0 = b_0 \right) \right)
\]

The size of this formula is exponential in \( n \).
Succinctness of LTL+Past

(\(\mathcal{P}\)): any two states that agree on propositions \(p_1\) to \(p_n\) also agree on proposition \(p_0\).

Let \(\mathcal{A}\) be a Büchi automaton corresponding to property (\(\mathcal{P}\)).

Let \(\Sigma = \{a_0, a_1, ..., a_{2^n-1}\}\) be the subsets of \(\{p_1, ..., p_n\}\).
Succinctness of LTL+Past

\((\mathcal{P})\): any two states that agree on propositions \(p_1\) to \(p_n\) also agree on proposition \(p_0\).

For each \(K \subseteq \{0, \ldots, 2^n - 1\}\), we define \(w_K = b_0 \ldots b_{2^n-1}\) with

\[
\begin{align*}
b_i &= \begin{cases} a_i & \text{if } i \in K \\ a_i \cup \{p_0\} & \text{otherwise} \end{cases}
\end{align*}
\]
Succinctness of LTL+Past

\((\mathcal{P})\): any two states that agree on propositions \(p_1\) to \(p_n\) also agree on proposition \(p_0\).

For each \(K \subseteq \{0, ..., 2^n - 1\}\), we define \(w_K = b_0...b_{2^n - 1}\) with

\[
\begin{align*}
  b_i &= \begin{cases}
    a_i & \text{if } i \in K \\
    a_i \cup \{p_0\} & \text{otherwise}
  \end{cases}
\end{align*}
\]

Lemma

There are \(2^{2^n}\) different such words.
Succinctness of LTL+Past

\((P)\): any two states that agree on propositions \(p_1\) to \(p_n\) also agree on proposition \(p_0\).

For each \(K \subseteq \{0, \ldots, 2^n - 1\}\), we define \(w_K = b_0 \ldots b_{2^n-1}\) with

\[
b_i = \begin{cases} a_i & \text{if } i \in K \\ a_i \cup \{p_0\} & \text{otherwise} \end{cases}
\]

Lemma

For any \(K \subseteq \{0, \ldots, 2^n - 1\}\), the word \(w_K^\omega\) is accepted by \(A\).
Succinctness of LTL+Past

(\mathcal{P})$: any two states that agree on propositions $p_1$ to $p_n$ also agree on proposition $p_0$.

For each $K \subseteq \{0, ..., 2^n - 1\}$, we define $w_K = b_0...b_{2^n-1}$ with

$$b_i = \begin{cases} a_i & \text{if } i \in K \\ a_i \cup \{p_0\} & \text{otherwise} \end{cases}$$

Lemma

For any $K \subseteq \{0, ..., 2^n - 1\}$, the word $w_K^\omega$ is accepted by $A$.

Lemma

For any $K \neq K'$, the word $w_{K'} \cdot w_K^\omega$ is not accepted by $A$. 
(P): any two states that agree on propositions \( p_1 \) to \( p_n \) also agree on proposition \( p_0 \).

Lemma

For any \( K \subseteq \{0, \ldots, 2^n - 1\} \), the word \( w^\omega_K \) is accepted by \( A \).

Lemma

For any \( K \neq K' \), the word \( w_{K'} \cdot w^\omega_K \) is not accepted by \( A \).

For any \( K \neq K' \), the states reached after reading \( w_K \) and after reading \( w_{K'} \) must be different.
Succinctness of LTL+Past

\((\mathcal{P})\): any two states that agree on propositions \(p_1\) to \(p_n\) also agree on proposition \(p_0\).

**Lemma**

For any \(K \subseteq \{0, ..., 2^n - 1\}\), the word \(w_K^\omega\) is accepted by \(A\).

**Lemma**

For any \(K \neq K'\), the word \(w_{K'} \cdot w_K^\omega\) is not accepted by \(A\).

For any \(K \neq K'\), the states reached after reading \(w_K\) and after reading \(w_{K'}\) must be different.

**Theorem**

Any Büchi automaton \(A\) characterizing property \((\mathcal{P})\) has at least \(2^{2^n}\) states.
Succinctness of LTL+Past

(\mathcal{P})$: any two states that agree on propositions $p_1$ to $p_n$ also agree on proposition $p_0$.

**Theorem**

Any Büchi automaton $A$ characterizing property (\mathcal{P}) has at least $2^{2^n}$ states.

**Corollary**

Any LTL formula expressing property (\mathcal{P}) has size at least $2^{n-1}$. 
Succinctness of LTL+Past

Consider now the following property, slightly different:

\((\mathcal{P}'):\) any state that agrees on propositions \(p_1\) to \(p_n\) with the initial state also agrees on proposition \(p_0\).
Succinctness of LTL+Past

Consider now the following property, slightly different:

\( (P') \): any state that agrees on propositions \( p_1 \) to \( p_n \) with the initial state also agrees on proposition \( p_0 \).

This can be expressed in LTL+Past by the following (polynomial-size) formula:

\[
G \left( \bigwedge_{i \geq 1} p_i \Leftrightarrow F^{-1} G^{-1} p_i \right) \Rightarrow (p_0 \Leftrightarrow F^{-1} G^{-1} p_0)
\]
Succinctness of LTL+Past

Consider now the following property, slightly different:

\((P')\): any state that agrees on propositions \(p_1\) to \(p_n\) with the initial state also agrees on proposition \(p_0\).

This can be expressed in LTL+Past by the following (polynomial-size) formula:

\[
\mathbf{G} \left( \left( \bigwedge_{i \geq 1} p_i \iff \mathbf{F}^{-1} \mathbf{G}^{-1} p_i \right) \Rightarrow (p_0 \iff \mathbf{F}^{-1} \mathbf{G}^{-1} p_0) \right).
\]

Let \(\varphi\) be an LTL formula expressing property \((P')\). Then \(\mathbf{G} \varphi\) precisely expresses property \((P)\), and thus has size at least \(2^{n-1}\).
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
The stuttering principle

Definition

Let \( w = w_0 w_1 \ldots \) be a word over AP. A letter \( a = w_i \) is stuttering in \( w \) if it appears several consecutive times, i.e., if \( a = w_i = w_{i+1} \).
The stuttering principle

**Definition**

Let $w = w_0 w_1 \ldots$ be a word over AP. A letter $a = w_i$ is **stuttering** in $w$ if it appears several consecutive times, i.e., if $a = w_i = w_{i+1}$.

**Example**

\[
    w = g \cdot r \cdot \underbrace{g \cdot g \cdot g \cdot g \cdot g}_{b} \cdot b \cdot r \ldots
\]
The stuttering principle

**Definition**

Let \( w = w_0 w_1 \ldots \) be a word over AP. A letter \( a = w_i \) is *stuttering* in \( w \) if it appears several consecutive times, i.e., if \( a = w_i = w_{i+1} \).

**Definition**

The relation \( \preccurlyeq \) is defined as follows:

\[
\begin{align*}
  w \preccurlyeq w' & \iff w \text{ is obtained from } w' \text{ by removing} \\
  & \text{one copy of a stuttering letter.}
\end{align*}
\]
The stuttering principle

Definition
Let \( w = w_0 w_1 \ldots \) be a word over AP. A letter \( a = w_i \) is stuttering in \( w \) if it appears several consecutive times, i.e., if \( a = w_i = w_{i+1} \).

Definition
The relation \( \preceq \) is defined as follows:

\[ w \preceq w' \iff w \text{ is obtained from } w' \text{ by removing one copy of a stuttering letter.} \]

Example
\[
\begin{array}{c}
g \cdot r \cdot \underbrace{g \cdot g \cdot g \cdot b \cdot r \ldots} \preceq g \cdot r \cdot \underbrace{g \cdot g \cdot g \cdot g \cdot b \cdot r \ldots}
\end{array}
\]
Definition

*Stuttering equivalence* is the least equivalence relation that subsumes $\ll$. 
The stuttering principle

**Definition**

*Stuttering equivalence* is the least equivalence relation that subsumes $\preceq$.

**Example**

The words $g \cdot (b \cdot b \cdot r)\omega$ and $g \cdot (b \cdot r \cdot r)\omega$ are stuttering equivalent.
The stuttering principle

**Definition**

*Stuttering equivalence* is the least equivalence relation that subsumes $\subseteq$.

**Theorem**

Two words are stuttering-equivalent iff they cannot be distinguished by any formula of $\mathcal{L}(U)$. 
The stuttering principle

**Definition**

*Stuttering equivalence* is the least equivalence relation that subsumes $\prec$.

**Theorem**

*Two words are stuttering-equivalent iff they cannot be distinguished by any formula of $\mathcal{L}(U)$.*

**Corollary**

$\mathcal{L}(U, X)$ has strictly more distinguishing power than $\mathcal{L}(U)$. 
A subword $w[i, j]$ of a word $w$ is $(m, n)$-redundant if the subword $w[i + j, mj - m + 1 + n]$ is a prefix of $w[i, j]^{\omega}$. 
The generalized stuttering principle

Definition
A subword $w[i, j]$ of a word $w$ is $(m, n)$-redundant if the subword $w[i + j, mj - m + 1 + n]$ is a prefix of $w[i, j]^\omega$.

Example

```
  r r g g r g r r g r r g r r g r r g g r
```
The generalized stuttering principle

Definition
A subword $w[i, j]$ of a word $w$ is $(m, n)$-redundant if the subword $w[i + j, mj - m + 1 + n]$ is a prefix of $w[i, j]^\omega$.

Example

$r$ $r$ $g$ $g$ $g$ $g$ $g$ $r$ $r$ $g$ $r$ $r$ $g$ $r$ $r$ $g$ $r$ $g$ $g$ $g$ $g$ $r$
The generalized stuttering principle

Definition

A subword $w[i, j]$ of a word $w$ is $(m, n)$-redundant if the subword $w[i + j, mj - m + 1 + n]$ is a prefix of $w[i, j]^\omega$.

Example

$r \ r \ g \ g \ g \ g \ r \ g \ r \ g \ g \ r \ g \ r \ g \ r \ g \ g \ g \ r$

$w[5, 3]^\omega = \begin{array}{cccccccccccc}
    r & g & r & r & g & r & r & g & r & r & g & r & r & g & r \\
\end{array}$
The generalized stuttering principle

Definition
A subword $w[i,j]$ of a word $w$ is $(m,n)$-redundant if the subword $w[i+j, mj - m + 1 + n]$ is a prefix of $w[i,j]^{\omega}$.

Example

$w[5, 3]^{\omega} = \text{r g r r g r r g r r g r r g r r g r}$

$w[5, 3]$ is (0,8)-redundant, but also (3,2)-redundant.
The generalized stuttering principle

Definition

Given two words $w$ and $w'$, and two integers $m$ and $n$, we define:

$$w \preceq_{m,n} w' \iff w \text{ is obtained from } w' \text{ by removing } (m, n)\text{-redundant subwords.}$$
The generalized stuttering principle

**Definition**

Given two words $w$ and $w'$, and two integers $m$ and $n$, we define:

$$w \preceq_{m,n} w' \iff w \text{ is obtained from } w' \text{ by removing } (m, n)\text{-redundant subwords.}$$

**Definition**

$(m, n)$-stuttering equivalence is the least equivalence relation that subsumes $\preceq_{m,n}$. 
The generalized stuttering principle

**Definition**

Given two words $w$ and $w'$, and two integers $m$ and $n$, we define:

$$w \preceq_{m,n} w' \iff w \text{ is obtained from } w' \text{ by removing } (m, n)\text{-redundant subwords.}$$

**Definition**

$(m, n)$-stuttering equivalence is the least equivalence relation that subsumes $\preceq_{m,n}$.

**Example**

$$rrgg \ rgr \ rgr \ ggr \ldots \preceq_{3,2} rrgg \ rgr \ rgr \ rgr \ rgr \ ggr \ldots$$
The generalized stuttering principle

Definition

We write $\mathcal{L}(U^m, X^n)$ for the fragment of LTL where nesting identical modalities is bounded (by $m$ and $n$ for $U$ and $X$, respectively).
The generalized stuttering principle

Example

The following formula is in $\mathcal{L}(U^2, X^4)$:

$$\text{green } U (X(\text{red } U \text{ green } \land X \text{ blue})) \lor X X X X X \text{ red}$$
The generalized stuttering principle

Example

The following formula is in $\mathcal{L}(U^2, X^4)$:

$$\text{green } U (X(\text{red } U \text{ green } \land X \text{ blue})) \lor X X X X X \text{ red}$$
The generalized stuttering principle

**Theorem (Kučera, Strejček, 2005)**

If two words are \((m, n)\)-stuttering equivalent, then they can’t be distinguished by formulas of \(\mathcal{L}(U^m, X^n)\).
The generalized stuttering principle

**Theorem (Kučera, Strejček, 2005)**

*If two words are \((m, n)\)-stuttering equivalent, then they can’t be distinguished by formulas of \(L(U^m, X^n)\).*

**Corollary**

*For any \(m\) and \(n\), there exists formulas in \(L(U^m, X^n)\) that can be expressed neither in \(L(U^{m-1}, X)\) nor in \(L(U^m, X^{n-1})\).*
The generalized stuttering principle

Example

We illustrate the case where $m = 3$ and $n = 2$: let

$$\varphi = F(green \land F(red \land F(blue \land XX green))).$$

and

$$w = brg brg brg \ y^\omega$$
$$w' = brg brg \ y^\omega$$

Then $w, 0 \models \varphi$ and $w', 0 \not\models \varphi$.

$w$ and $w'$ are $(2, 2)$- and $(3, 1)$-stutter equivalent, and thus can be distinguished neither by formulas in $L(U^2, X^2)$ nor by formulas in $L(U^3, X^1)$. 
The generalized stuttering principle

Theorem

The family $\mathcal{L}(U^m, X^n)$ form a strict hierarchy w.r.t. expressive power.
The generalized stuttering principle

Theorem

The family $\mathcal{L}(U^m, X^n)$ form a strict hierarchy w.r.t. expressive power.

⚠️ It seems natural to conjecture that

if a property can be expressed in $\mathcal{L}(U^{m+1}, X^n)$ and in $\mathcal{L}(U^m, X^{n+1})$, then it can be expressed in $\mathcal{L}(U^m, X^n)$.

This is false: for instance, $F(green \land F \neg green)$ and $F(green \land X \neg green)$ are equivalent, but cannot be expressed in $\mathcal{L}(U^1)$. 
Outline of today’s lecture

1. LTL+Past and the $\mu$-calculus

2. LTL+Past and Büchi automata
   - Büchi automata
   - From LTL+Past to Büchi automata
   - Büchi automata are more expressive
   - Alternating Büchi automata
   - Application: Succinctness of LTL+Past

3. Stuttering
   - The stuttering principle
   - The generalized stuttering principle

4. Ehrenfeucht-Fraïssé games
   - The rules of the game
   - EF games and the Until-Since hierarchy
Ehrenfeucht-Fraïssé games

EF games are 2-player games used to show that two linear structures $T$ and $U$ can (or cannot) be distinguished by a logic.
Ehrenfeucht-Fraïssé games

EF games are 2-player games used to show that two linear structures $T$ and $U$ can (or cannot) be distinguished by a logic.

**Definition**

A *configuration* of the game is a couple $\langle t, u \rangle$ where $t \in T$ and $u \in U$. 
Ehrenfeucht-Fraïssé games

EF games are 2-player games used to show that two linear structures $T$ and $U$ can (or cannot) be distinguished by a logic.

Definition
A configuration of the game is a couple $\langle t, u \rangle$ where $t \in T$ and $u \in U$.

Definition
From a configuration $\langle t_0, u_0 \rangle$, the rules of a $k$-round EF game is defined recursively as follows:

- when $k = 0$, player $A$ wins if $t_0$ and $u_0$ are labeled by exactly the same atomic propositions;
- when $k \geq 1$, two cases may arise:
  - if $t_0$ and $u_0$ are not labeled by the same atomic propositions, then the game stops and player $A$ is declared the winner;
  - otherwise, player $A$ plays an $U$-move or a $S$-move.
Ehrenfeucht-Fraïssé games

EF games are 2-player games used to show that two linear structures $T$ and $U$ can (or cannot) be distinguished by a logic.

**Definition**

An $U$-move from configuration $\langle t, u \rangle$ is played as follows:

- player $A$ selects the structure he wants to play on (say $T$), and an element $t'$ of that structure s.t. $t \leq t'$;
- player $B$ responds by choosing an element $u'$ in the other structure s.t. $u \leq u'$;
- player $A$ has then two choices:
  - either he sets the new configuration to $\langle t', u' \rangle$;
  - or he selects a position $u''$ in $U$ s.t. $u \leq u'' < u'$, player $B$ chooses $t''$ in $T$ with $t \leq t'' < t'$, and the new configuration is $\langle t'', u'' \rangle$.

$S$-moves are symmetric.
Ehrenfeucht-Fraïssé games

Example

Initial configuration
Ehrenfeucht-Fraïssé games

Example

Player $\mathcal{A}$ plays an $\textbf{U}$-move on the first structure.
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds on the other structure.
Ehrenfeucht-Fraïssé games

Example

This is the new configuration.
Ehrenfeucht-Fraïssé games

Example

Player $A$ plays an $U$-move on the first structure.
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds on the other structure...
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds on the other structure... and loses.
Ehrenfeucht-Fraïssé games

Example

Initial configuration
Ehrenfeucht-Fraïssé games

Example

Player $A$ plays an $U$-move on the first structure.
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds on the other structure.
Ehrenfeucht-Fraïssé games

Example

This is the new configuration.
Ehrenfeucht-Fraïssé games

Example

Player $A$ plays a $S$-move on the second structure.
Ehrenfeucht-Fraïssé games

Example

Player $\mathcal{B}$ responds on the other structure.
Ehrenfeucht-Fraïssé games

Example

Player A picks an intermediary position on the first structure.
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds...
Ehrenfeucht-Fraïssé games

Example

Player $B$ responds... and loses.
EF games and the Until-Since hierarchy

Definition

We write $\mathcal{L}(\{\textbf{U}, \textbf{S}\}^k)$ for the fragment of $\mathcal{L}(\textbf{U}, \textbf{S})$ where the temporal height is bounded by $k$. 
EF games and the Until-Since hierarchy

Definition

We write $\mathcal{L}(\{U, S\}^k)$ for the fragment of $\mathcal{L}(U, S)$ where the temporal height is bounded by $k$.

Example

$\text{green } U ((\text{green } S \text{ red}) S (\text{red } U \text{ green } \land (\text{green } U \text{ blue})))$
Definition

We write $\mathcal{L}(\{U, S\}^k)$ for the fragment of $\mathcal{L}(U, S)$ where the temporal height is bounded by $k$.

Example

$$\text{green } U ((\text{green } S \text{ red}) \ S (\text{red } U \text{ green } \land (\text{green } U \text{ blue})))$$
EF games and the Until-Since hierarchy

**Definition**

We write $\mathcal{L}(\{\text{U}, \text{S}\}^k)$ for the fragment of $\mathcal{L}(\text{U}, \text{S})$ where the temporal height is bounded by $k$.

**Example**

$\text{green U (\text{green S red) S (red U green } \land (\text{green U blue}))}$

![Diagram representing the example formula]
EF games and the Until-Since hierarchy

**Definition**

We write $\mathcal{L}(\{U, S\}^k)$ for the fragment of $\mathcal{L}(U, S)$ where the temporal height is bounded by $k$.

**Theorem (Etessami & Wilke, 2000)**

Player $B$ has a winning strategy in the $k$-round game from a configuration $\langle t, u \rangle$ iff, for any formula $\varphi \in \mathcal{L}(\{U, S\}^k)$, we have

$$\langle T, t \rangle \models \varphi \iff \langle U, u \rangle \models \varphi.$$
EF games and the Until-Since hierarchy

Now, consider the following two structures:

\[ T_k = ((r b)^{k-1} g b)^{k-1} (r b)^k g b ((r b)^{k-1} g b)^{k-1} y^\omega \]

\[ U_k = ((r b)^{k-1} g b)^{k-1} (r b)^{k-1} g b ((r b)^{k-1} g b)^{k-1} y^\omega \]
EF games and the Until-Since hierarchy

Now, consider the following two structures:

\[ T_k = ((r \ b)^{k-1} g \ b)^{k-1} (r \ b)^k \ g \ b ((r \ b)^{k-1} g \ b)^{k-1} y^\omega \]
\[ U_k = ((r \ b)^{k-1} g \ b)^{k-1} (r \ b)^{k-1} g \ b ((r \ b)^{k-1} g \ b)^{k-1} y^\omega \]

For instance:

\[ T_3 = rbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgby^\omega \]
\[ U_3 = rbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgby^\omega \]
EF games and the Until-Since hierarchy

Now, consider the following two structures:

\[
T_k = ((r b)^{k-1} g b)^{k-1} (r b)^k g b ((r b)^{k-1} g b)^{k-1} y^\omega
\]
\[
U_k = ((r b)^{k-1} g b)^{k-1} (r b)^{k-1} g b ((r b)^{k-1} g b)^{k-1} y^\omega
\]

For instance:

\[
T_3 = rbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgby^\omega
\]
\[
U_3 = rbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgby^\omega
\]

**Lemma**

*Player B has a winning strategy in the k-round game based on* \(T_k\) *and* \(U_k\) *from configuration* \(\langle 0, 0 \rangle\).
EF games and the Until-Since hierarchy

Now, consider the following two structures:

\[ T_k = ((rb)^{k-1} gb)^{k-1} (rb)^k gb ((rb)^{k-1} gb)^{k-1} y^\omega \]
\[ U_k = ((rb)^{k-1} gb)^{k-1} (rb)^{k-1} gb ((rb)^{k-1} gb)^{k-1} y^\omega \]

For instance:

\[ T_3 = rbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrbgbbrb
EF games and the Until-Since hierarchy

Now, consider the following two structures:

\[ T_k = ((r \ b)^{k-1} g b)^{k-1} (r \ b)^k g b ((r \ b)^{k-1} g b)^{k-1} y^\omega \]
\[ U_k = ((r \ b)^{k-1} g b)^{k-1} (r \ b)^{k-1} g b ((r \ b)^{k-1} g b)^{k-1} y^\omega \]

For instance:

\[ T_3 = rbrbgbrbrbgbrbrbrbgbrbrbgbrbrbgbrbrbgby^\omega \]
\[ U_3 = rbrbgbrbrbgbrbrbgbrbrbgbrbrbgbrbrbgby^\omega \]

**Lemma**

There exists a formula \( \varphi \in L(\{U, S\}^{k+1}) \) that distinguishes between \( \langle T_k, 0 \rangle \) and \( \langle U_k, 0 \rangle \).

For instance:

\[ \top \ U (b \land (b \ S (r \land r \ S b)) \land (b \ U (r \land r \ U (b \land b \ U r)))) \]
Theorem (Etessami & Wilke, 2000)

The hierarchy $\mathcal{L}(\{U, S\}^k)$ is strict (w.r.t. distinguishing power).
EF games and the Until-Since hierarchy

**Theorem (Etessami & Wilke, 2000)**

*The hierarchy* $\mathcal{L}(\{U, S\}^k)$ *is strict (w.r.t. distinguishing power).*

Similar techniques can be used to prove that:

**Theorem (Etessami & Wilke, 2000)**

*The hierarchy* $\mathcal{L}(\{U, S\}^k, \{X, F, X^{-1}, F^{-1}\}^r)$ *is strict (w.r.t. distinguishing power).*