Fondements pour la vérification des systèmes temps-réel et concurrents

Lecture 3
Alternation and LTL extensions

Stéphane Demri

October 8th, 2007
Summary from previous lecture

\[ \mathcal{B}_\phi = (\Sigma, S, S_0, \rho, F_1, \ldots, F_k) \]

- \( S \) is the set of maximally consistent sets wrt \( \phi \),

- \( \Sigma = \mathcal{P}(\text{PROP}) \),

- \( S_0 = \{ X \in S : \phi \in X \} \),

- \( Y \in \rho(X, a) \) iff
  - \( X \cap \text{PROP} = a \),
  - for \( X\psi \in \text{cl}(\phi) \), \( X\psi \in X \) iff \( \psi \in Y \),

- If \( \psi_1 \cup \psi_1', \ldots, \psi_k \cup \psi_k' \) occurs in \( \phi \), then
  \[ F_i \overset{\text{def}}{=} \{ X \in S : \text{either } \psi_i \cup \psi_i' \notin X \text{ or } \psi_i' \in X \} \]

- If \( U \) does not occur in \( \phi \), then \( k = 1 \) and \( F_1 = S \).
Simple complexity properties

- $L(\mathcal{B}_\phi) = \text{Models}(\phi)$.

- Checking whether $\mathcal{X} \subseteq \text{cl}(\phi)$ belongs to $S$ [resp. $S_0$, $F_1$, $\ldots$, $F_k$] can be done in polynomial-time in $|\phi|$.

- Checking whether $\mathcal{Y} \in \rho(\mathcal{X}, a)$ can be done in polynomial-time in $|\phi|$.

- $|S|$ is in $2^\mathcal{O}(|\phi|)$.

- Elements in $S$ can be encoded in polynomial-space in $|\phi|$.
**NPSPACE algorithm**

1. Guess \( s_0 \in S_0, s_1 \in F_1, \ldots, s_k \in F_k; \)

2. \( i := 0; s := s_0 \) (current state);

3. While \( s \neq s_1 \) and \( i < |S| \) do
   
   3.1 Guess \( s' \) such that \( s \xrightarrow{a} s' \) for some \( a \in \Sigma; \)
   
   3.2 \( i := i + 1; s := s'. \)

4. If \( s \neq s_1 \), then abort otherwise
   
   4.1 \( i := 0; j := 2; \)
   
   4.2 While \( i := 0 \) or \((j \neq 1 \text{ and } i < |S| \times k)\) do
       
       4.2.1 Guess \( s' \) such that \( s \xrightarrow{a} s' \) for some \( a \in \Sigma; \)
       
       4.2.2 \( i := i + 1; s := s'. \)
       
       4.2.3 If \( s' \in F_j \) then nondeterministically choose either
              
              \( j := (j \mod k) + 1 \) or skip;
       
   4.3 If \( s = s_1 \), then accept, otherwise abort.
Complexity

- $\mathcal{B}_\phi$ is in exponential size in $|\phi|$.

- Testing on-the-fly the nonemptiness of $\mathcal{B}_\phi$ can be done in $\text{NPSpace}$.

- By Savitch’s theorem: $\text{NPSpace} = \text{PSpace}$.

- Satisfiability for LTL is in $\text{PSpace}$. 
What about model-checking?

- Let $\mathcal{M} = (W, R, L)$ be a finite and total Kripke structure and $s_0 \in W$.

- $L(\mathcal{A}_\mathcal{M}, s_0) = \text{Paths}(\mathcal{M}, s_0)$:

  $$\mathcal{A}_\mathcal{M}, s_0 = (\mathcal{P}(\text{PROP}), W, \{s_0\}, \rho, W)$$

  where $\rho(s, a) \overset{\text{def}}{=} \{s' : (s, s') \in R, \ a = L(s)\}$ for all $s \in W$ and $a \subseteq \text{PROP}$.

- $\mathcal{M}, s_0 \models \exists \phi$ iff $L(\mathcal{A}_\mathcal{M}, s_0) \cap L(\mathcal{B}_\phi) \neq \emptyset$.

- LTL model-checking is in $\text{PSPACE}$. 

Stéphane Demri Fondements pour la vérification des systèmes temps-réel et co
Exercise (bis)

- Adapt the automata-based approach to deal with $X^{-1}$:
  \[ \sigma, i \models X^{-1}\phi \iff i > 0 \text{ and } \sigma, i - 1 \models \phi. \]

- Adapt the automata-based approach to deal with $S$:
  \[ \sigma, i \models \phi S\psi \iff \text{there is } j \leq i \text{ such that } \sigma, j \models \psi \text{ and for } j < k \leq i, \text{ we have } \sigma, k \models \phi. \]

- Characterize the complexity of model-checking and satisfiability problems for $\text{LTL}(U, X, X^{-1}, S)$. 
LTL and alternating Büchi automata
Positive Boolean formulae

Given a finite set $\mathcal{X}$, $\mathcal{B}^+(\mathcal{X})$ denotes the set of positive Boolean formulae built over $\mathcal{X} \cup \{\bot, \top\}$.

Example: $(s \lor s') \land s'' \in \mathcal{B}^+(\{s, s', s''\})$.

Each subset $\mathcal{Y} \subseteq \mathcal{X}$ can be viewed as a propositional valuation: $s \in \mathcal{Y}$ iff $s$ is interpreted as true.

$\mathcal{Y} \models \phi \in \mathcal{B}^+(\mathcal{X}) \iff \phi$ holds true in the interpretation $\mathcal{Y}$.

Example: $\{s, s''\} \models (s \lor s') \land s''$. 
Alternating Büchi automata

- $A = (\Sigma, S, s_0, \rho, F)$ with
  - $\Sigma$: finite alphabet,
  - $S$: finite set of states,
  - $s_0 \in S$: initial state,
  - $\rho: S \times \Sigma \rightarrow B^+(S)$: transition relation,
  - $F \subseteq S$: set of accepting states.

Encoding nondeterministic BA in alternating BA:

$$\rho(s, a) \mapsto \bigvee_{s' \in \rho(s, a)} s'$$
Accepting runs

- A run $r$ on the $\omega$-sequence $a_0 a_1 a_2 \ldots \in \Sigma^\omega$ is a (possibly infinite) tree whose nodes are labelled by states in $S$ and s.t.
  - $r = (T, \mathcal{T})$ where $T$ is a tree and $\mathcal{T} : T \rightarrow S$,
  - Root of $T$ is labelled by $s_0$ (i.e. $\mathcal{T}(\epsilon) = s_0$),
  - For $x \in T$, if $|x| = i$ (depth in $T$) and $\mathcal{T}(x) = s$ then
    \begin{align*}
    \{\mathcal{T}(x_1), \ldots, \mathcal{T}(x_k)\} \models \rho(s, a_i) \text{ where } x_1, \ldots, x_k \text{ are the children of } x.
    \end{align*}

- A run is accepting $\overset{\text{def}}{\iff}$ for every infinite branch of $T$, an accepting state is repeated infinitely often.

- $L(\mathcal{A})$: set of $\omega$-sequences in $\Sigma^\omega$ for which there is an accepting run.
Properties

- ABA are closed under intersection, union and complementation (with quadratic blow-up).

- Nonemptiness problem for ABA is $\text{PSPACE}$-complete [Chandra & Kozen & Stockmeyer, JACM 81].
From ABA to NBA

Given an ABA $\mathcal{A} = (\Sigma, S, s_0, \rho, F)$, there is a NBA $\mathcal{A}_n = (\Sigma, S', s'_0, \rho', F')$ s.t. $L(\mathcal{A}) = L(\mathcal{A}_n)$.

Idea of the proof: $\mathcal{A}_n$ guesses the set of states at each level of an accepting run of $\mathcal{A}$.

A state of $\mathcal{A}_n$ is a set of states from $\mathcal{A}$.

One needs to encode which states are visited infinitely often on each branch of the accepting run of $\mathcal{A}$.

A state of $\mathcal{A}$ is divided in two subsets in order to distinguish branches that visit recently an accepting state.
\( S' \overset{\text{def}}{=} \mathcal{P}(S) \times \mathcal{P}(S) \)

if \((X, \mathcal{Y}) \in S\) then \(\mathcal{Y}\) is the set of states on branches that visit recently an accepting state,

\( S'_0 \overset{\text{def}}{=} \{ (\{s_0\}, \emptyset) \} \);

\( F' \overset{\text{def}}{=} \emptyset \times \mathcal{P}(S) \);

Transition relation \( \rho' \) (2 subcases):

\[\begin{align*}
\rho'((\emptyset, X'), a) & \overset{\text{def}}{=} \{ (\mathcal{Y}, \mathcal{Y}') : \exists Z \models \bigwedge_{s \in X'} \rho(s, a), \mathcal{Y} = Z \setminus F, \mathcal{Y}' = Z \cap F \} \\
\rho'((X, X'), a) & \overset{\text{def}}{=} \{ (\mathcal{Y}, \mathcal{Y}') : \exists Z, Z' \text{ such that} \} \\
& \quad Z \models \bigwedge_{s \in X} \rho(s, a), Z' \models \bigwedge_{s \in X'} \rho(s, a), \mathcal{Y} = Z \setminus F, \mathcal{Y}' = Z' \cup (Z \cap F) \}.
\end{align*}\]
Negative normal form

- \( \phi R \psi \overset{\text{def}}{=} \neg (\neg \phi U \neg \psi) \).

- A formula built over \( \lor, \land, X, U, R, \neg \) and PROP in which negation occurs only in front of propositional variables is said to be in negative normal form.

- Every formula in LTL is equivalent to a formula in negative normal form (reduction in polynomial-time).

- Some essential properties:
  - \( \neg X \phi \) is equivalent to \( X \neg \phi \),
  - \( \neg (\phi U \psi) \) is equivalent to \( (\neg \phi R \neg \psi) \),
  - \( \neg (\phi \land \psi) \) is equivalent to \( (\neg \phi \lor \neg \psi) \).
From LTL formulae to ABA

\[ A = (\Sigma, S, s_0, \rho, F) \]

- \( S \) is the set of subformulae of \( \phi \),

- \( s_0 \overset{\text{def}}{=} \phi \),

- \( \Sigma \overset{\text{def}}{=} \mathcal{P}(\text{PROP}) \),

- \( F \) is equal to \( S \) restricted to formulae whose outermost connective is not \( \text{U} \).

- Transition relation:
  - \( \rho(p, a) \overset{\text{def}}{=} \top \) if \( p \in a \); \( \rho(\neg p, a) \overset{\text{def}}{=} \top \) if \( p \not\in a \),
  - \( \rho(p, a) \overset{\text{def}}{=} \bot \) if \( p \not\in a \); \( \rho(\neg p, a) \overset{\text{def}}{=} \bot \) if \( p \in a \),
  - \( \rho(\psi \land \psi', a) \overset{\text{def}}{=} \rho(\psi, a) \land \rho(\psi', a) \),
  - \( \rho(X\psi, a) \overset{\text{def}}{=} \psi \),
  - \( \rho(\psi \text{U} \varphi, a) \overset{\text{def}}{=} \rho(\varphi, a) \lor (\rho(\psi, a) \land (\psi \text{U} \varphi)) \),
  - \( \rho(\psi \text{R} \varphi, a) \overset{\text{def}}{=} \rho(\varphi, a) \land (\rho(\psi, a) \lor (\psi \text{R} \varphi)) \).
Example

- Extensions:
  - $\rho(\top, a) \overset{\text{def}}{=} \top$; $\rho(\bot, a) \overset{\text{def}}{=} \bot$;
  - $\rho(F\psi, a) \overset{\text{def}}{=} \rho(\psi, a) \lor F\psi$;
  - $\rho(G\psi, a) \overset{\text{def}}{=} \rho(\psi, a) \land G\psi$.

- Transition relation for $FGp$:

<table>
<thead>
<tr>
<th>$s$</th>
<th>$\rho(s, \emptyset)$</th>
<th>$\rho(s, {p})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FGp$</td>
<td>$FGp$</td>
<td>$Gp \lor FGp$</td>
</tr>
<tr>
<td>$Gp$</td>
<td>$\bot$</td>
<td>$Gp$</td>
</tr>
<tr>
<td>$p$</td>
<td>$\bot$</td>
<td>$\top$</td>
</tr>
</tbody>
</table>
Summary

- \( L(\mathcal{A}_\phi) \) is the set of models for \( \phi \).

- The number of states of \( \mathcal{A}_\phi \) is polynomial in \( |\phi| \).

- The difficulty is in the nonemptiness test for ABA.

- Corollary: When PROP is finite and fixed, satisfiability for LTL is in \( \text{PSPACE} \).

- NB: LTL satisfiability/model-checking can be reduced in logspace to LTL satisfiability/model-checking with at most 2 propositional variables.
Exercise

- Construct the ABA for $FGp \land FGq$ with the previous systematic construction and compare it with a direct construction.

- Represent an accepting run for $\{p\}{q}\{q\}{p}\{p, q\}^\omega$. 
Wolper’s automata-based operators
From BA to LTL formulae?

- $\phi \mapsto A\phi$ [Büchi, 62; Wolper & Vardi, IC 94].

- BA $A$ over $P(\text{PROP}) \mapsto$ LTL formula $\phi_A$?

- $\omega$-sequences accepted by the BA below are exactly the sequences with $\{p\}$ on even positions:

![Diagram](attachment:image.png)

- What about

  1. $G(p \Leftrightarrow XXp) \land p \land X\neg p$,
  2. $p \land G(p \Rightarrow XXp)$,
  3. $q \land X\neg q \land G(q \Leftrightarrow XXq) \land G(q \Rightarrow p)$

  ?
Expressive power

- By Kamp’s theorem, LTL\((U^s, S^s)\) is as expressive as first-order theory on \((\mathbb{N}, <)\).

- LTL is as expressive as first-order theory on \((\mathbb{N}, <)\) with respect to initial equivalence.

- Büchi automata are as expressive as monadic second-order theory on \((\mathbb{N}, <)\).

- **Proposition** [Wolper, IC 83]
  There is no LTL formula \(\phi\) built over the unique propositional variable \(p\) such that \(\text{Models}(\phi)\) is exactly the set of LTL models such that \(p\) holds true on every even position (on odd positions, \(p\) may hold true or not).
Proof of proposition

- Suppose that there is a formula $\phi$ built over $p$ and $|\phi|_X$ be the number of $X$ occurrences in $\phi$.

- $\mathbb{N}(\phi) \overset{\text{def}}{=} \{ i \in \mathbb{N} : \{p\}^i \cdot \emptyset \cdot \{p\}^\omega \models \phi \}$

- We shall show that for $n \geq |\phi|_X + 1$, $n \in \mathbb{N}(\phi)$ iff $n + 1 \in \mathbb{N}(\phi)$.

- Consequently, $|\phi|_X + 1 \in \mathbb{N}(\phi)$ iff $|\phi|_X + 2 \in \mathbb{N}(\phi)$.

- However, exactly one structure among $\{p\}^{|\phi|_X+1} \cdot \emptyset \cdot \{p\}^\omega$ and $\{p\}^{|\phi|_X+2} \cdot \emptyset \cdot \{p\}^\omega$ is a model for $\phi$, a contradiction.
Induction

- Base case: $\phi = p$
  - $\mathbb{N}(\phi) = \mathbb{N} \setminus \{0\}$ and $|\phi|_X = 0$.
  - For $n \geq 1$, $n \in \mathbb{N}(\phi)$ iff $n + 1 \in \mathbb{N}(\phi)$.

- Induction hypothesis: for $\phi$ s.t. $|\phi| \leq N$, for $n \geq |\phi|_X + 1$, $n \in \mathbb{N}(\phi)$ iff $n + 1 \in \mathbb{N}(\phi)$.
Case $\phi = \phi_1 \land \phi_2$

- Cases for $\neg$ and $\lor$ are analogous.

- $|\phi|_X = |\phi_1|_X + |\phi_2|_X$.

- Equivalence between the propositions below ($n \geq |\phi|_X + 1$):
  - $n \in \mathbb{N}(\phi)$,
  - $n \in \mathbb{N}(\phi_1)$ and $n \in \mathbb{N}(\phi_2)$ ($\land$ semantics);
  - $n + 1 \in \mathbb{N}(\phi_1)$ and $n + 1 \in \mathbb{N}(\phi_2)$
    (by (IH) since $|\phi_1|, |\phi_2| \leq N$ and $n \geq |\phi_1|_X + 1, |\phi_2|_X + 1$),
  - $n + 1 \in \mathbb{N}(\phi)$ ($\land$ semantics).
\[ \phi = X\psi \]

\[ |\phi|_X = 1 + |\psi|_X. \]

- Equivalence between the propositions below \((n \geq |\phi|_X + 1 \geq 2)\):
  - \(n \in \mathbb{N}(\phi)\),
  - \(n - 1 \in \mathbb{N}(\psi)\) (X semantics),
  - \(n \in \mathbb{N}(\psi)\) (by (IH) since \(|\psi| \leq N\) and \(n - 1 \geq |\psi|_X + 1\)),
  - \(n + 1 \in \mathbb{N}(\phi)\) (X semantics).
\[ \phi = \phi_1 \cup \phi_2 \]

\[ |\phi|_X = |\phi_1|_X + |\phi_2|_X. \]

Let \( n \geq |\phi|_X + 1 \) and suppose \( \sigma^n = \{p\}^n \emptyset \{p\}^\omega \models \phi. \)

There exists \( j \geq 0 \) such that \( \sigma^n, j \models \phi_2 \) and for \( 0 \leq k < j \), we have \( \sigma^n, k \models \phi_1. \)

We shall show that \( \sigma^{n+1} = \{p\}^{n+1} \emptyset \{p\}^\omega \models \phi \), i.e. \( n + 1 \in \mathbb{N}(\phi). \).
First part of the until case

- Subcase $j = 0$
  - $|\phi_2| \leq N$ and $n \geq |\phi_2|X + 1$.
  - By (IH), $n + 1 \in \mathbb{N}(\phi_2)$, whence $n + 1 \in \mathbb{N}(\phi)$.

- Subcase $j \geq 1$
  - $|\phi_1| \leq N$ and $n \geq |\phi_1|X + 1$.
  - $n \in \mathbb{N}(\phi_1)$.
  - By (IH), $n + 1 \in \mathbb{N}(\phi_1)$.
  - Hence $n + 1 \in \mathbb{N}(\phi)$.
Second part of until case

- Now suppose that $\sigma^{n+1} = \{p\}^{n+1} \emptyset \{p\}^\omega \models \phi$.

- There exists $j \geq 0$ s.t. $\sigma^{n+1}, j \models \phi_2$ and for $0 \leq k < j$, we have $\sigma^{n+1}, k \models \phi_1$.

- If $j = 0$, then since $|\phi_2| \leq N$ and $n \geq |\phi_2| \chi + 1$, by (IH) $n \in \mathbb{N}(\phi_2)$, whence $n \in \mathbb{N}(\phi)$.

- If $j \geq 1$, then $\sigma^{n+1}, 1 \models \phi$, whence $n \in \mathbb{N}(\phi)$. 
Extended temporal logic ETL

- FSA $\mathcal{A} = (\Sigma, S, S_0, \rho, F)$ with $\Sigma = a_1 < \ldots < a_k$.

- ETL = LTL + all formulae $\mathcal{A}(\phi_1, \ldots, \phi_k)$.

- $\sigma, i \models \mathcal{A}(\phi_1, \ldots, \phi_k)$ $\overset{\text{def}}{\iff}$
  - either $S_0 \cap F \neq \emptyset$ ($\epsilon \in L(\mathcal{A})$),
  - or there is a finite word $a_{i_1}a_{i_2} \ldots a_{i_n} \in L(\mathcal{A})$ such that for every $1 \leq j \leq n$, $\sigma, i + (j - 1) \models \phi_{i_j}$.

- If $S_0 \cap F \neq \emptyset$, then $\mathcal{A}(\phi_1, \ldots, \phi_k)$ is equivalent to $\top$.

- $L(\mathcal{A}) = \{ab^i a : i \geq 0\}$ and $a < b$: $\mathcal{A}(p, q)$

Stéphane Demri
Fondements pour la vérification des systèmes temps-réel et complémentaire
Define $X$ and $U$ with automata-based operators.

Define a formula $\phi$ in ETL built over $p$ whose models are exactly those in which $p$ holds true at least on even positions.

Model-checking and satisfiability problems for ETL are $\text{PSPACE}$-complete [Vardi & Wolper, IC 94].

ETL has the same expressive power as Büchi automata:
- For any BA $A$ over $\Sigma = \{a_1, \ldots, a_k\}$, for any map $l : \Sigma \rightarrow \mathcal{X}$ where $\mathcal{X}$ is a set of finite subsets of PROP, there is a formula $\phi$ in ETL built over $\bigcup_i l(a_i)$ s.t. $L(A) = \text{Models}(\phi)$. 

Stéphane Demri  
Fondements pour la vérification des systèmes temps-réel et concurr...
Expressive power

The class of languages defined by ETL formulae is equal to the class of languages defined by

- Büchi automata,
- formulae from monadic second-order theory for \((\omega, <)\), also known as S1S,
- \(\omega\)-regular expressions (or by finite union of sets \(U \cdot V^\omega\) with regular \(U, V \subseteq \Sigma^*\)),
- formulae from LTL with second-order quantification.
  - \(\sigma, \sigma' : \mathbb{N} \to \mathcal{P}(\text{PROP}), p \in \text{PROP.}\)
    \[
    \sigma \approx_p \sigma' \overset{\text{def}}{\iff} \text{for } i \in \mathbb{N}, \sigma(i) \setminus \{p\} = \sigma'(i) \setminus \{p\}.
    \]
  - LTL with second-order quantification: \(\sigma, i \models \forall p \phi \overset{\text{def}}{\iff} \text{for } \sigma' \)
    \[
    \text{s.t. } \sigma \approx_p \sigma', \text{ we have } \sigma', i \models \phi.
    \]
- formulae from LTL with fixed-point operators [Vardi, POPL 88].
Consiseness

ETL is a powerful and concise extension of LTL:

- the nonemptiness problem for Büchi automata is \( \text{NLOGSPACE} \)-complete,

- \( \text{MC}^\exists (\text{ETL}) \) and \( \text{SAT} (\text{ETL}) \) are \( \text{PSPACE} \)-complete,

- satisfiability for LTL with fixed-point operators is \( \text{PSPACE} \)-complete [Vardi, POPL 88],

- satisfiability for S1S is non-elementary (time complexity is not bounded by any tower of exponentials of fixed height).
Extension with context-free languages

- $C$: class of languages of finite words.

- $\text{LTL} + C$: extension of LTL with formulae $L(\phi_1, \ldots, \phi_n)$ for some $L \in C$.

- ETL = LTL + REG where REG is the class of regular languages represented by finite-state automata.

- Context-free languages (in CF) represented by context-free grammars.

- SAT(LTL + CF) is undecidable.
Proof

- Language equality between context-free grammars is undecidable.

- Reduction to SAT(LTL + CF).

- $G_1, G_2$: CF grammars over $\Sigma = \{a_1, \ldots, a_n\}$.

- $G_1^+, G_2^+$: CF grammars over $\Sigma^+ = \{a_1, \ldots, a_n, a_{n+1}\}$ s.t. $L(G_1^+) = L(G_1) \cdot \{a_{n+1}\}$ and $L(G_2^+) = L(G_2) \cdot \{a_{n+1}\}$.

- $G_1^+$ and $G_2^+$ can be effectively computed from $G_1$ and $G_2$, respectively.

- $L(G_1) = L(G_2)$ iff $L(G_1^+) = L(G_2^+)$.

- We shall construct $\phi_{G_1, G_2}$ in LTL + CF s.t. $\phi_{G_1, G_2}$ is satisfiable iff $L(G_1) \neq L(G_2)$. 

Stéphane Demri
Building $\phi_{G_1,G_2}$

- $\phi_{G_1,G_2}$ is built over $p_1, \ldots, p_{n+1}$ and holds true only in structures s.t.
  - exactly one variable from $p_1, \ldots, p_{n+1}$, holds true at each position,
  - $p_{n+1}$ holds true at a unique position (end marker).

- UNI encodes these properties:

$$\text{UNI} \overset{\text{def}}{=} G\left( \bigvee_{1 \leq i \leq n+1} p_i \right) \land G\left( \bigwedge_{1 \leq i \leq n+1} (p_i \Rightarrow \bigwedge_{1 \leq j \neq i \leq n+1} \neg p_j) \right) \land$$

$$\left( (p_{n+1} \land XG\neg p_{n+1}) \lor \neg p_{n+1} U(p_{n+1} \land XG\neg p_{n+1}) \right)$$

- Equivalence between
  - $\text{L}(G_1^+) \neq \text{L}(G_2^+)$,
  - $\text{UNI} \land \neg (\text{L}(G_1^+)(p_1, \ldots, p_{n+1}) \Leftrightarrow \text{L}(G_2^+)(p_1, \ldots, p_{n+1}))$ is satisfiable.
Equivalence (I)

- Suppose $L(G_1^+) \neq L(G_2^+)$ with $a_{i_1}a_{i_2} \cdots a_{i_l}a_{n+1} \in L(G_1^+)$ and $a_{i_1}a_{i_2} \cdots a_{i_l}a_{n+1} \not\in L(G_2^+)$.  

- Wlog, $l \geq 1$.  

- $\sigma:\{p_{i_1}\} \cdot \{p_{i_2}\} \cdots \{p_{i_l}\} \cdot \{p_{n+1}\} \cdot \{p_1\}^\omega$.  

- We have  
  - $\sigma \models \text{UNI}$,  
  - $\sigma \models L(G_1^+) (p_1, \ldots, p_{n+1})$,  
  - $\sigma \not\models L(G_2^+) (p_1, \ldots, p_{n+1})$ since the only finite word ending by $\{p_{n+1}\}$ in $\sigma$ is $\{p_{i_1}\} \cdot \{p_{i_2}\} \cdots \{p_{i_l}\} \cdot \{p_{n+1}\}$ and $a_{i_1}a_{i_2} \cdots a_{i_l}a_{n+1} \not\in L(G_2^+)$.  

Suppose $\sigma, 0 \models \text{UNI} \land \neg (L(G_1^+)(p_1, \ldots, p_{n+1}) \Leftrightarrow L(G_2^+)(p_1, \ldots, p_{n+1}))$.

Assume $\sigma \models L(G_1^+)(p_1, \ldots, p_{n+1})$ and $\sigma \not\models L(G_2^+)(p_1, \ldots, p_{n+1})$.

A simple reasoning allows to deduce that $L(G_1^+) \neq L(G_2^+)$. 
Special context-free languages

- $L_0 = \{a_1^k \cdot a_2 \cdot a_1^{k-1} \cdot a_3 : k \geq 1\}$.

- $L_1 = \{a_1^k \cdot a_2 \cdot a_1^k \cdot a_3 : k \geq 0\}$ ($L_0 = \{a_1\} \cdot L_1$).

- Valid formulae in $LTL + \{L_0, L_1\}$.
  - $L_1(p, q, r) \iff (q \land Xr) \lor L_0(p, q, p \land Xr)$,
  - $L_0(p, q, r) \iff p \land XL_1(p, q, r)$,
  - $F \phi \iff L_1(\top, \phi, \top)$,
  - $X \phi \iff L_1(\bot, \top, \phi)$.
Undecidability

- SAT(LTL + {L₁}) is undecidable.

- Consequently, MC³(LTL + {L₁}) is undecidable. (satisfiability reduces to it by building a complete Kripke structure)

- Reduction from the recurrence problem for domino games [Harel, 85].
  
  **input:** a domino game Dom with a distinguished color c.
  
  **output:** 1, if Dom can pave \( \mathbb{N} \times \mathbb{N} \) where the color c occurs infinitely often.

- Let Dom = (C, D, Col) be a domino game with
  
  - C = \{1, \ldots, n\} and c = 1,
  
  - D = \{d₁, \ldots, dₘ\},
  
  - Col : D \times \{up, down, left, right\} \rightarrow \{1, \ldots, n\}
Syntactic resources

We use the following propositional variables:

- $in$ holds true when the state encodes a position in $\mathbb{N}^2$. There are states in the model that do not correspond to positions in $\mathbb{N}^2$. $out$ is equivalent to the negation of $in$.

- For $1 \leq j \leq m$, we introduce $j$: “position in $\mathbb{N}^2$ associated to the current position has domino type $d_j$”.

- For every $1 \leq i \leq n$, we use the variables $up_i$, $down_i$, $left_i$, $right_i$. 
Every state encoding a position in $\mathbb{N}^2$ is occupied by a unique domino:

$$G(in \Rightarrow \bigvee_{j=1}^{m} (j \land \bigwedge_{j' = 1, j' \neq j}^{m} \neg j'))$$

Propositional variables for colours are compatible with the definition of domino types:

$$G(in \Rightarrow \bigwedge_{j=1}^{m} j \implies \bigwedge_{side \in \{up, down, right, left\}} \bigwedge_{1 \leq j' \neq Col(d_j, side) \leq n} \neg side_{j'})$$

PAVE: conjunction of above formulae.
SNAKE formula

SNAKE: conjunction of following formulae:

- $G(in \Leftrightarrow \neg out)$,

- $in \land Xout \land XXin \land XXXin \land XXXXXout$,

- $G(out \Rightarrow XL_1(in, out, in \land Xout))$.
  ($L_1 = \{a_1^k \cdot a_2 \cdot a_1^k \cdot a_3 : k \geq 0\}$).

- Only structure (built over $in$ and $out$) satisfying SNAKE:
  $$\{in\} \cdot \{out\} \cdot \{in\}^2 \cdot \{out\} \cdot \{in\}^3 \cdot \{out\} \cdot \{in\}^4 \ldots$$
Stéphane Demri

Fondements pour la vérification des systèmes temps-réel et concurrents
Difficulty of the proof is not to design a path through $\mathbb{N}^2$ but to define a path on which it is easy to access to neighbours (right or top).

**DIRECTION**: conjunction of formulae:

- $G(\uparrow \iff \neg \downarrow)$,
- $\downarrow \land X \uparrow$,
- $G(in \land X_{in} \land \uparrow \Rightarrow X \uparrow)$
  ("we stay on ascending chain"),
- $G(in \land X_{in} \land \downarrow \Rightarrow X \downarrow)$
  ("we stay on descending chain"),
- $G(in \land X_{out} \land \uparrow \Rightarrow (X \downarrow \land XX \downarrow))$
  ("we pass from ascending to descending chain"),
- $G(in \land X_{out} \land \downarrow \Rightarrow (X \uparrow \land XX \uparrow))$
  ("we pass from descending to ascending chain").
Only structure (built over \textit{in}, \textit{out}, \textit{↑} and \textit{↓}) satisfying \textbf{SNAKE} \land \textbf{DIRECTION}:

\[
\{\textit{in}, \downarrow\}\{\textit{out}, \uparrow\} \cdot \{\textit{in}, \uparrow\}^2 \cdot \{\textit{out}, \downarrow\} \cdot \{\textit{in}, \downarrow\}^3 \cdot \{\textit{out}, \uparrow\} \cdot \{\textit{in}, \uparrow\}^4 \ldots
\]
The path allows to access to adjacent states as follows:
- in \( \{\text{in}, \uparrow\} \), we access to the right neighbour with \( L_1 \),
- in \( \{\text{in}, \uparrow\} \), we access to the up neighbour with \( L_0 \),
- in \( \{\text{in}, \downarrow\} \), we access to the right neighbour with \( L_0 \),
- in \( \{\text{in}, \downarrow\} \), we access to the up neighbour with \( L_1 \).

\textbf{CONSTRAINTS:} conjunction of formulae
- \[ G(\text{in} \land \uparrow \implies (\land_{1 \leq i \leq n} \text{right}_i \Rightarrow L_1(\text{in}, \text{out}, \text{left}_i))))), \]
- \[ G(\text{in} \land \uparrow \implies (\land_{1 \leq i \leq n} \text{up}_i \Rightarrow L_0(\text{in}, \text{out}, \text{down}_i))))), \]
- \[ G(\text{in} \land \downarrow \implies (\land_{1 \leq i \leq n} \text{right}_i \Rightarrow L_0(\text{in}, \text{out}, \text{left}_i))))), \]
- \[ G(\text{in} \land \downarrow \implies (\land_{1 \leq i \leq n} \text{up}_i \Rightarrow L_1(\text{in}, \text{out}, \text{down}_i))))). \]
High undecidability

- **REC**: $\text{GF}(\text{in} \land \bigvee_{\text{side} \in \{\text{left, right, up, down}\}} \text{side}_1)$.

- *Dom* can pave $\mathbb{N}^2$ by repeating infinitely often the colour 1 iff

  $\text{PAVE} \land \text{SNAKE} \land \text{DIRECTION} \land \text{CONSTRAINTS} \land \text{REC}$

  is satisfiable in $\text{LTL} + \{L_0, L_1\}$.

- Since $X$, $F$ and $L_0$ can expressed with $L_1$, satisfiability and model-checking problems for propositional calculus with the temporal operator defined with $L_1$ are highly undecidable.
Exercise (bis)

1. $\phi[\psi]\rho$ be an LTL formula with subformula $\psi$ at the occurrence $\rho$. Show that $\phi[\psi]\rho$ is satisfiable iff $\phi[p]\rho \land G(p \leftrightarrow \psi)$ is satisfiable where $p$ is a new propositional variable no occurring in $\phi[\psi]\rho$.

2. Conclude that there is a logarithmic space reduction from SAT(LTL) to SAT(LTL$_{2\omega}$).

3. Show that SAT(LTL$_1(X)$) is NP-complete (independent of 1. and 2.).