Temporal Logics over Presburger Constraints

Stéphane Demri

Laboratoire Spécification et Vérification

CNRS & INRIA Project SECSI & ENS de Cachan
Temporal logics

- Aspects of temporality in Computer Science
  - Specification and verification of concurrent and reactive systems.
  - Real-time processes and systems.
  - Temporal databases.

- Logics as formal specification languages
  - To define mathematically the correctness of programs and systems.
  - To express properties without ambiguities.
  - To make formal proofs.
Labeled transition systems

• An LTS is a structure \( \langle S, (\xrightarrow{a})_{a\in Act} \rangle \) where
  – \( S \) is a non-empty set of states, \( Act \) is a non-empty set of actions,
  – \( \xrightarrow{a} \) is a binary relation in \( S \times S \).

• Examples: programs or processes run concurrently on the same computing device, finite automata, coffee machines, Kripke frames.
Finite representations of LTS
Presburger Arithmetic

- First-order theory of $\langle \mathbb{Z}, 0, + \rangle$.

- Decidability shown in [Presburger 29].

- Quantifier elimination in presence of modulo constraints.

- Satisfiability in $3\text{EXPTIME}$.

- Presburger formulae define exactly semilinear sets.
Presburger constraints on LTS

- Constraints on the number of occurrences [Bouajjani & Echahed & Habermehl 95].

- Constraints on the number of children (for semistructured data) [Seidl et al 04, Lugiez & Dal Zilio 05, Demri & Lugiez 06].

- Constraints on the values of variables in LTS in which the states are tuples of integers.
Constraint system

- Constraint system: \( \mathcal{D} = \langle D, (R_\alpha)_{\alpha \in I} \rangle \).

- Interpretation domains of program variables.

- Atomic constraint: \( R(x_1, \ldots, x_t), x_i \in \text{VAR} \).

- A \( \mathcal{D} \)-valuation \( v : \text{VAR} \rightarrow \mathcal{D} \).

- Examples: \( \langle \mathbb{N}, =, < \rangle, \langle \mathbb{N}, =, \text{succ} \rangle, \langle \mathbb{R}, =, < \rangle, \langle \mathbb{Z}, =, < \rangle, \langle \{0, 1\}^*, \subset, = \rangle, \langle \mathbb{Z}, (R_\phi(x_1, \ldots, x_n))_{\phi(x_1, \ldots, x_n) \in \text{Presburger}} \rangle \cdots \)
• Atomic term constraint $R(x_{n_1}x_1, \ldots, x_{n_t}x_t)$.

• $x^i x$ interpreted as the value of $x$ in the $i$th next state.

• $\phi ::= R(x_{n_1}x_1, \ldots, x_{n_t}x_t) \mid X\phi \mid \phi U \phi \mid \neg \phi \mid \ldots$

• Models: $\sigma : \mathbb{N} \rightarrow (\text{VAR} \rightarrow D)$.

• $\sigma, j \models R(x_{n_1}x_1, \ldots, x_{n_t}x_t)$ iff
  
  value of $x_1$ in the $j+n_1$th state
  
  
  \[
  (\sigma(j + n_1)(x_1), \ldots, \sigma(j + n_t)(x_t)) \in R
  \]

  i.e. values at different states can be compared.
**Linear-time temporal operators**

\[ X\phi : \text{next-time } \phi \]

\[ \phi_1 U \phi_2 : \phi_1 \text{ until } \phi_2 \]

\[ F\phi : \text{sometimes } \phi \]
CLTL(\(\mathcal{D}\)) problems

- Satisfiability problem for CLTL(\(\mathcal{D}\)):
  \textbf{instance:} a CLTL(\(\mathcal{D}\)) formula \(\phi\),
  \textbf{question:} is there a model \(\sigma\) such that \(\sigma \models \phi\)?

- Model-checking problem for CLTL(\(\mathcal{D}\)):
  \textbf{instance:} A \(\mathcal{D}\)-automaton \(A\) and a CLTL(\(\mathcal{D}\)) formula \(\phi\),
  \textbf{question:} are there a symbolic \(\omega\)-word \(v = \phi_0, \phi_1, \ldots\) accepted
  by \(A\), a model \(\sigma\) (a realization of \(v\)) such that \(\sigma \models \phi\) and for
  every \(i \geq 0\), \(\sigma, i \models \phi_i\)?
Existential model-checking

\[ Xx = 2 \]

\[ Xx = x - 1 \quad Xx = x + 1 \quad \models (x = 0) \wedge GF(x = 0) \]
About plain LTL

- **Formulae:** $\phi ::= p \mid \neg \phi \mid \phi \land \psi \mid \phi U \psi \mid X \phi$.

- **Models:** $\sigma : \mathbb{N} \rightarrow 2^{AP}$, $\sigma, i \models p$ iff $p \in \sigma(i)$.

- $L(\phi) = \{ \sigma \in (2^{AP})^\omega : \sigma, 0 \models \phi \}$.

- $\phi \leadsto$ Büchi automaton $A_\phi$ [Vardi & Wolper 86] s.t. $L(\phi) = L(A_\phi)$.

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

- Model-checking and satisfiability are PSPACE-complete [Sistla & Clarke 85].
Complexity of CLTL($D$, $<$, $=$)

- Symbolic model: sequence of maximally consistent sets of constraints.

- Every model of CLTL($D$, $<$, $=$) has a unique symbolic model (its abstraction).

- Given a formula $\phi$ in either CLTL($\mathbb{R}$, $<$, $=$) or CLTL($\mathbb{Q}$, $<$, $=$), the abstractions of the models for $\phi$ form an $\omega$-regular set.

- **Theorem.** Satisfiability and model-checking for either CLTL($\mathbb{R}$, $<$, $=$) or CLTL($\mathbb{Q}$, $<$, $=$) are PSPACE-complete.

- For CLTL($\mathbb{N}$, $<$, $=$), $\omega$-regularity is not systematic but

- **Theorem.** [Demri & D’Souza 02] Satisfiability and MC for CLTL($\mathbb{N}$, $<$, $=$) are PSPACE-complete.
Integer periodicity constraints

- Fragments of Presburger arithmetic with quantitative/qualitative constraints of the form

  \[ x \equiv_k y + c, \quad x \equiv_k c \quad + \quad \neg, \wedge, \exists \ldots \]

- Such constraints are used in many formalisms:
  - DATALOG with integer periodicity constraints.
  - Formalisms dealing with calendars.
  - Temporal reasoning in database access control.
  - Periodic time in generalized databases.

- Can we plug such quantitative temporal constraints in plain LTL
  - preserving decidability (full Presburger LTL is undecidable)
  - without increasing the computational complexity?
Constraint language $\text{IPC}^{++}$

- $p ::= x \equiv_k y + c \mid x \equiv_k c \mid p \land p \mid \exists x \ p \mid (\text{IPC})$
  
  \[ x \equiv_k y + [c_1, c_2] \mid x = y \mid x \sim d \mid \neg p \]

- with
  - $x, y \in \text{VAR}$,
  - $k, c_1, c_2 \in \mathbb{N} (0 \leq c_1 \leq c_2 \leq k - 1)$,
  - $\sim \in \{<, >, =\}$,
  - $d \in \mathbb{Z}$.

- Extension of IPC [Toman&Chomicki98].

- $\text{IPC}^+ = \text{IPC}^{++}$ minus the clause ’$x = y$’.

- Interpretation $v : \text{VAR} \rightarrow \mathbb{Z}$. 
Semantics

- \( v \models x \sim d \overset{\text{def}}{\iff} v(x) \sim d \text{ with } \sim \in \{<, >, =\}. \)

- \( v \models x = y \overset{\text{def}}{\iff} v(x) = v(y). \)

- \( v \models x \equiv_k c \overset{\text{def}}{\iff} \exists \alpha \in \mathbb{Z} \text{ s.t. } v(x) = \alpha \times k + c \) (\( 0 \leq c \leq k - 1 \)).

- \( v \models x \equiv_k y + [c_1, c_2] \overset{\text{def}}{\iff} \exists \alpha \in \mathbb{Z}, \exists c \in [c_1, c_2] \text{ s.t. } v(x) - v(y) = \alpha \times k + c. \)

- \( v \models \exists x \ p \overset{\text{def}}{\iff} \text{there is } z \in \mathbb{Z} \text{ s.t. } v[x \leftarrow z] \models p \) where \( v[x \leftarrow z](x') = v(x') \) if \( x' \neq x \), and \( v[x \leftarrow z](x) = z. \)
Past LTL over IPC$^{++}$: PLTL$^{\text{mod}}$

- Syntax:
  \[ \phi ::= p[x_1 \leftarrow X^{i_1}x_{j_1}, \ldots, x_k \leftarrow X^{i_k}x_{j_k}] \mid \neg \phi \mid \phi \land \phi \mid X\phi \mid \phi U \phi \mid X^{-1}\phi \mid \phi U^{-1}\phi \]

  with $x_1, \ldots, x_k$ free variables of $p \in \text{IPC}^{++}$

- Model: $\sigma : \mathbb{N} \times \text{VAR} \rightarrow \mathbb{Z}$.

- $X^i x_j$ interpreted as the value of $x_j$ in the $i$th next state.

- Example: $x \equiv_2 0 \land \mathcal{G}(X x \equiv_2 x + 1)$.
  Size in $\mathcal{O}(n)$. 

Temporal Logics over Presburger Constraints – p. 17
Semantics of PLTL$^\text{mod}$

- $\sigma, i \models p[x_1 \leftarrow x^{i_1} x_{j_1}, \ldots, x_k \leftarrow x^{i_k} x_{j_k}]$ iff $[x_1 \leftarrow \sigma(i + i_1, x_{j_1}), \ldots, x_k \leftarrow \sigma(i + i_k, x_{j_k})] \models p$ (for IPC$^{++}$).

- Future-time operators:
  - $\sigma, i \models x\phi$ iff $\sigma, i + 1 \models \phi$.
  - $\sigma, i \models \phi U \phi'$ iff there is $j \geq i$ s.t. $\sigma, j \models \phi'$ and for every $i \leq k < j$, we have $\sigma, k \models \phi$.

- Past-time operators:
  - $\sigma, i \models x^{-1}\phi$ iff $i > 0$ and $\sigma, i - 1 \models \phi$.
  - $\sigma, i \models \phi U^{-1}\phi'$ iff there is $0 \leq j \leq i$ s.t. $\sigma, j \models \phi'$ and for every $j < k \leq i$, we have $\sigma, k \models \phi$. 
Complexity of IPC++

- Complexity:

**Theorem.** IPC++-satisfiability is PSPACE-complete.

- Quantifier elimination:

**Theorem.** Given a constraint \( p \) in IPC++, one can compute an equivalent quantifier-free \( p' \) in polynomial space in \( |p| \) (but \( |p'| \) is in \( O(2^{|p|}) \)).
Equivalence to satisfiability

Theorem. The model-checking and satisfiability problems for $\text{PLTL}^{\text{mod}}$ are inter-reducible with respect to logspace transformations.

- $\phi \in \text{PLTL}^{\text{mod}}$, $d_0, \ldots, d_{n+1}$, $k_1, \ldots, k_u$, $K$.

- $l : 1 + \text{greatest } i \text{ such that } x^i x \text{ occurs in } \phi$.

- $k = s \times l$ with free variables $x_1, \ldots, x_s$ in $\phi$. 
Shift of \( l \)-pack of states

- \( \Sigma_\phi = (\{0, \ldots, n + 1\} \times \{0, \ldots, K - 1\})^k \times 2^{\{1, \ldots, k\}} \).

- Shift of \( l \)-pack of states:

![Diagram of states](image)

- Concrete model: \( \sigma : \mathbb{N} \times \{x_1, \ldots, x_s\} \rightarrow \mathbb{Z} \).

- Symbolic model: \( \sigma' : \mathbb{N} \rightarrow \Sigma_\phi \).
Büchi automata

\( A_\phi \): intersection of the following Büchi automata

- **\( A_1 \) (realization)**
  \[ a \xrightarrow{a} a' \text{ iff } a, a' \in \Sigma_\phi \text{ and } a \text{ has a realization.} \]
  \[ a \xrightarrow{a} a'? \text{ can be checked in polynomial-time.} \]

- **\( A_2 \) (shift)**
  \[ a \xrightarrow{a} a' \text{ iff } a' \text{ is obtained from } a \text{ by a single shift.} \]
  \[ a \xrightarrow{a} a'? \text{ can be checked in polynomial-time.} \]

- **\( A_3 \) (PLTL)**
  \[ X \xrightarrow{a} Y \text{ implies each atomic formula of } X \text{ is satisfied by } a. \]
  \[ X \xrightarrow{a} Y'? \text{ can be checked in PSPACE.} \]
**Complexity of PLTL\textsuperscript{mod}**

- $\phi$ is satisfiable iff $L(A_\phi) \neq \emptyset$.

- Emptiness of $A_\phi$ can be checked in polynomial space in $|\phi|$.

- Such a decomposition with three automata fails with the constraints of the form $x < y$.

**Theorem.** [Demri 06] Model-checking and satisfiability for PLTL\textsuperscript{mod} are PSPACE-complete.

**Corollary.** Adding a finite amount of MSO-definable operators preserves the PSPACE upper bound. (consequence of [Gastin&Kuske03])
Extended single string automata

- Automata recognizing a unique $\omega$-sequence to define time granularities [Wijsen00,DalLago&Montanari01].
  Time granularity $\mathbb{N} \rightarrow 2^{\text{TimeDomain}}$.

- Business week: $(\square\square\square\square\square\square\square\square\square\square\square)^\omega$.

- Example:
  $\square, \neg x \equiv_{2^n} 2^n - 1, x := x + 1$

- Associated $\omega$-word: $\square^{2^n} \cdot \square^\omega$. 
Equivalence problem

• Equivalence problem for ESSA:
  input: two ESSA $A$ and $A'$.
  question: Is $w_A = w_{A'}$?

• The equivalence problem is in PSPACE.
  Construction of a $\text{PLTL}^{\text{mod}}$-automaton $B$ such that

  $$B \models \top \iff w_A = w_{A'}.$$

• The equivalence problem is PSPACE-hard.
  Reduction of QBF by simulating the algorithm for solving QBF.

• PSPACE-hardness holds true for many strict subproblems.
### A summary

<table>
<thead>
<tr>
<th></th>
<th>LTL/PLTL</th>
<th>LTL/PLTL + (\downarrow)</th>
<th>LTL/PLTL + (\exists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>({x &lt; y, x = y})</td>
<td>PSPACE</td>
<td>(\Sigma_1^1)</td>
<td>(\Sigma_1^1)</td>
</tr>
<tr>
<td>({x - y = c, x = c})</td>
<td>(\Sigma_1^1)</td>
<td>(\Sigma_1^1)</td>
<td>(\Sigma_1^1)</td>
</tr>
<tr>
<td>IPC + ({x &lt; y, x = y})</td>
<td>PSPACE</td>
<td>(\Sigma_1^1)</td>
<td>(\Sigma_1^1)</td>
</tr>
<tr>
<td>IPC(^+)</td>
<td>PSPACE</td>
<td>EXPSPACE</td>
<td>EXPSPACE</td>
</tr>
<tr>
<td>IPC(^{++})</td>
<td>PSPACE</td>
<td>(\Sigma_1^1)</td>
<td>(\Sigma_1^1)</td>
</tr>
</tbody>
</table>
**CLTL(DL)**

- **Constraint language** $\text{DL}$

\[
\phi ::= x \sim y + d \mid x \sim d \mid \phi \land \phi \mid \neg \phi
\]

- $x \equiv_k c$ and $x + y + z < 5$ are not in $\text{DL}$.

- Satisfiability problem for $\text{CLTL}(\text{DL})$ is $\Sigma_1^1$-complete. By reduction from recurrent reachability for non-deterministic Minsky machines (easy).

- $\text{CLTL}_{k}^{l}(\text{DL})$: restriction of $\text{CLTL}(\text{DL})$ to formulae with at most $k$ variables and $x$-length less or equal to $l$. 

Temporal Logics over Presburger Constraints – p. 27
Two main undecidable fragments

- Satisfiability for $\text{CLTL}_2(\text{DL})$ and $\text{CLTL}_1(\text{DL})$ are $\Sigma_1^1$-complete.

- **Corollary.** Counter logic $\mathcal{L}_p$ restricted to two variables [Comon & Cortier 00] is highly undecidable.

- **Corollary.** Satisfiability for $\text{CLTL}(\mathbb{N}, =, +1)$ restricted to a unique variable is highly undecidable.

- Model-checking for $\text{CLTL}_2(\text{DL})$ and $\text{CLTL}_1(\text{DL})$ are $\Sigma_1^1$-complete.
A PSPACE-complete fragment

- **Theorem.** [Demri & Gascon 06] Model-checking and satisfiability for $\text{CLTL}^1_1(\text{DL})$ are PSPACE-complete.

- Symbolic models of an $\text{CLTL}^1_1(\text{DL})$ formula can be recognized by one-counter automata where
  - the counter is interpreted in $\mathbb{Z}$,
  - there are zero tests and sign tests,
  - accepted words are $\omega$-sequences (Büchi acceptance condition),
  - updates of the counter are among 0,-1,1.

- Nonemptiness problem for this class of one-counter automata is NLOGSPACE-complete.
Model-checking one-counter automata

- Quantifier-free Presburger arithmetic \( \text{QFP} \):

\[
\phi ::= \sum_{i \in I} a_i x_i = d \mid \sum_{i \in I} a_i x_i < d \mid \sum_{i \in I} a_i x_i \equiv_k c \mid \neg \phi \mid \phi \land \phi
\]

- Satisfiability for \( \text{CLTL}^1_1(\text{QFP}) \) is known to be \( \Sigma^1_1 \)-complete.

- **Theorem.** Model-checking for \( \text{CLTL}^\omega_1(\text{QFP}) \) over one-counter automata with updates in \( \mathbb{Z} \) is PSPACE-complete.

- Symbolic runs of the one-counter automata satisfying a \( \text{CLTL}^\omega_1(\text{QFP}) \) formula are recognizable by one-counter automata.

- **Open problem:** Complexity of nonemptiness problem for one-counter automata with updates in \( \mathbb{Z} \).
Adding the freeze operator

- \( \text{VAR} = \text{FV} \) (flexible variables) \( \cup \) \( \text{RV} \) (rigid variables).

- Unary \( \downarrow_{y=x_j x} \) with \( y \in \text{RV}, x \in \text{FV} \).

- Environment \( \rho: \text{RV} \to D \).

- Models \( \sigma: \mathbb{N} \to (\text{FV} \to D) \).

- \( \sigma, j \models_{\rho} \downarrow_{y=x^n x} \phi \) iff \( \sigma, j \models_{\rho[y \mapsto \sigma(j+n)x]} \phi \).

- \( \sigma, j \models_{\rho} R(t_1, \ldots, t_n) \) iff \( (\llbracket t_1 \rrbracket_{\sigma, \rho, j}, \ldots, \llbracket t_2 \rrbracket_{\sigma, \rho, j}) \in R \) with

\[
\llbracket X^n x \rrbracket_{\sigma, \rho, j} = \sigma(j+n)(x) \quad \text{if } x \text{ is in } \text{FV} \\
\llbracket y \rrbracket_{\sigma, \rho, j} = \rho(y) \quad \text{if } y \text{ is in } \text{RV}
\]
Freeze quantifier in hybrid logics

- $\downarrow x \phi$: $\phi$ holds true in the variant model where $x$ is true only at the current state [Blackburn&Seligman95, Goranko96].

- $x$: pointer to a state.

- Every reachable state can be visited inf. often: $\forall G \downarrow x \exists F x$.

\[ \forall G \downarrow x \exists F x \]
Predicate $\lambda$-abstraction

- How to interpret constants in first-order modal logics?

- Current value of the constant $c$ satisfies the predicate $P$ in the future [Fitting02]
  \[ \langle \lambda x \cdot FP(x) \rangle (c) \]

- $\text{LTL}_{\lambda=}$ with $x$, $u$, and 3 registers is undecidable [Lisitsa&Potapov05].
Main undecidability results

- **Theorem.** [Demri & Lazić & Nowak 05] Satisfiability for $\text{CLTL}^\downarrow(\mathbb{N}, =)$ restricted to two rigid variables is $\Sigma^1_1$-complete.

  By reduction from the recurrent reachability problem for nondeterministic Minsky machines.

- The above problem is also undecidable with finite models.

- **Theorem.** [Demri & Lazić 06] Satisfiability for $\text{CLTL}^\downarrow(\mathbb{N}, =)$ restricted to one rigid variable is $\Pi^0_1$-complete.

  By reduction from infinitary nonemptiness for incrementing counter automata.
A decidability result

- **Theorem.** [Demri & Lazić 06] Satisfiability for $\text{CLTL}^\downarrow(\mathbb{N}, =)$ restricted to one rigid variable over finite models is decidable but not primitive recursive.

- Decidability proof in two steps:
  1. From formulae to alternating register automata.
  2. From alternating register automata with a unique register to incrementing counter automata.

- Non primitive recursiveness is also proved in two steps
  1. Finitary nonemptiness for incrementing counter automata is non PR by adapting [Schnoebelen02].
  2. This problem can be reduced in logspace to satisfiability in $\text{CLTL}^\downarrow(\mathbb{N}, =)$ restricted to one rigid variable.
Branching-time extensions

- CTL variant of $\text{CLTL}(\mathbb{N}, <, =) +$ constants has an undecidable model-checking problem [Cerans94].

- Quantification over paths can simulate quantification over integer.

- Existential-CTL* is decidable [Cerans94] using a sophisticated argument based on well-structured systems.

- Existential-CTL* variant of $\text{PLTL}^{\text{mod}}$ without past is decidable [Bozzelli & Gascon 06].
Other related topics

- Model-checking flat counter systems with finite monoid [Boigelot 98, Finkel & Leroux 02].

- Model-checking Petri nets, reversal-bounded counter systems, etc.

- Description logics over concrete domains [Lutz02].
Perspectives

• New constraint systems
  – Strings with prefix, subword, factor etc ... relation
  – Heterogeneous domains

Open problem: Decidability status of CLTL(\{0,1\}^*, =, ⊆).

• Decidability status when restricted use of freeze as for \( x = F y \).

• Other branching-time extensions.

• Extended versions of LTL for reasoning about programs with
  pointers (cf Rémi Brochenin’s master thesis 06).