LTL with the Freeze Quantifier and Register Automata

Stéphane Demri\(^1\)  Ranko Lazić\(^2\)

\(^1\)LSV, CNRS & ENS Cachan & INRIA Futurs

\(^2\)University of Warwick

LICS, August 2006, Seattle
Overview

LTL with freeze
- Models
- Syntax
- Example

Register automata
- Introduction
- Alternation

Expressiveness
- FO over data words
- Register automata

Complexity of satisfiability
- From RA to ICA
- From ICA to LTL with freeze and RA
- Complexity results
- Characterisation of languages
Data words

Finite alphabet & infinite domain:
Infinite-state computations, XML documents, Timed words, …

\[ a \quad a \quad b \quad b \quad a \quad b \]
\[ URL_1 \quad URL_2 \quad URL_1 \quad URL_2 \quad URL_3 \quad URL_3 \]
\[ a \quad a \quad b \quad b \quad a \quad b \]
\[ 3 \quad 2.5 \quad 3 \quad 2.5 \quad 4 \quad 4 \]
Data words

Finite alphabet & infinite domain:
Infinite-state computations, XML documents, Timed words, ...
LTL with the Freeze Quantiﬁer

\[ \phi ::= \top \mid a \mid \uparrow r \sim \mid \neg \phi \mid \phi \land \phi \mid O(\phi, \ldots, \phi) \mid \downarrow r \phi \]

\[ r \in \{1, \ldots, n\} \]

\[ O \in \{X, X^{-1}, F, F^{-1}, U, U^{-1}, \ldots\} \]
## Freeze quantifier

<table>
<thead>
<tr>
<th>Logic</th>
<th>Symbol</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timed logics</td>
<td>$x \cdot \phi(x)$</td>
<td>[Alur &amp; Henzinger, JACM '94]</td>
<td></td>
</tr>
<tr>
<td>Hybrid logics</td>
<td>$\downarrow p \phi(p)$</td>
<td>[Goranko, JoLLI '96]</td>
<td></td>
</tr>
<tr>
<td>Modal logics</td>
<td>$\langle \lambda x \cdot \phi(x) \rangle(c)$</td>
<td>[Fitting, JLC '02]</td>
<td></td>
</tr>
</tbody>
</table>
Example

\[ F(a \land \downarrow_1 XF(a \land \uparrow_1 \sim)) \]

\[ a \quad a \quad b \quad b \quad a \quad b \]

Stephane Demri, Ranko Lazic
1-way or 2-way, Nondeterministic or Alternating, $n$ registers

Over infinite data words: weak parity acceptance.
Special case of Büchi and co-Büchi acceptance.
Register automata  [Kaminski & Francez, TCS '94]
Pebble automata  [Neven, Schwentick & Vianu, ToCL '04]
Timed automata  [Alur & Dill, TCS '94]
Data automata  [Bouyer, Petit & Thérien, I & C '03]
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

Introduction
Alternation

\[
G (a \Rightarrow \downarrow_1 X((a \Rightarrow \neg \uparrow_1 \sim) \cup (b \land \uparrow_1 \sim)))
\]

\[
\begin{array}{c}
\text{4} \\
\text{end} \\
\text{3}
\end{array}
\]

\[
\begin{array}{c}
a \Rightarrow \downarrow_1 \\
a \Rightarrow \uparrow_1 \sim \\
b \land \uparrow_1 \sim
\end{array}
\]

Stéphane Demri, Ranko Lazić
[Kaminski & Francez, TCS ’94],
[Neven, Schwentick & Vianu, ToCL ’04]
[Kaminski & Francez, TCS ’94], [Neven, Schwentick & Vianu, ToCL ’04]

Provided $\text{LogSpace} \subseteq \text{NLogSpace} \subseteq \text{PTime}$. 
$\text{FO}(\sim, <, +1)$

Theorem

\[
\text{Simple } LTL_1^\downarrow(x, x^{-1}, F, F^{-1}) \overset{\text{LOGSPACE}}{\longrightarrow} \text{PSPACE} \overset{\Longleftarrow}{\longrightarrow} \text{FO}^2(\sim, <, +1)
\]

[Etessami, Vardi & Wilke, I & C '02]

$LTL(x, x^{-1}, F, F^{-1})$ is equivalent to $\text{FO}^2(<, +1)$.

[Bojańczyk et al., LICS '06]

$\text{FO}^2(\sim, <, +1)$ SAT is as hard as Petri net Reachability.
Simple $\text{LTL}_1^f(X, X^{-1}, F, F^{-1})$

\begin{align*}
F(a \land \downarrow_1 XF(a \land \uparrow_1 \sim)) \\
\exists x (a(x) \land \exists y (x < y \land a(y) \land x \sim y)) \\
F(a \land \downarrow_1 XF(b \land XF(a \land \uparrow_1 \sim)))
\end{align*}
From LTL$\downarrow$ to register automata

**Theorem**

$LTL_n(x, x^{-1}, u, u^{-1}) \xrightarrow{\text{LogSpace}} 2ARA_n$. 

Stéphane Demri, Ranko Lazić
[Kaminski & Francez, TCS ’94]
‘... it is very likely that [decidability of language containment of 1NRA in 1NRA₁] can be extended to infinite data words.’

[French, TIME ’03], [Demri, Lazić & Nowak, TIME ’05], [Lisitsa & Potapov, TIME ’05]:

<table>
<thead>
<tr>
<th>Registers</th>
<th>SAT&lt;ω</th>
<th>SATω</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTL↓(X, F)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LTL↓(X, U)</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>LTL↓(X, F, F⁻¹)</td>
<td>Σ₁&lt;comp.</td>
<td>Σ₁&lt;comp.</td>
</tr>
</tbody>
</table>

Stéphane Demri, Ranko Lazić  LTL with the Freeze Quantifier and Register Automata
\[
\text{LogSpace} \downarrow \text{LTL}_1^{(x, u)} \downarrow \text{SAT} \downarrow \text{1ARA}_1 \uparrow \neg \text{EMP}
\]
Incrementing Counter Automata

\[
\text{inc}.c \\;
\text{ifzero}.c \\;
\varepsilon \\;
\text{dec}.c \\;
\text{ifzero}.c
\]

\[
a \\;
\text{ifzero}.c \\;
b
\]

Stephane Demri, Ranko Lazic

LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

$LTL_1(x, u)$ SAT

$\text{LogSpace}$

$1\text{ARA}_1 \neg \text{EMP}$

$\text{PSPACE}$

$\text{ICA} \neg \text{EMP}$
Proof.

1. Quotient runs by $\sim$.
2. Represent levels and steps using counters.
3. For infinite data words, use [Miyano & Hayashi, TCS '84]: weak parity alternating $\leftrightarrow$ Büchi nondeterministic.
4. Incrementing errors cannot cause a false acceptance.
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

$q$

end

$q'$

$b \land \uparrow_1 \beta$

$a \Rightarrow \downarrow_1$

$q$

$q$

$LTL$ with the Freeze Quantifier and Register Automata

Stéphane Demri, Ranko Lazić

LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

4
q
\[ q \rightarrow 3 \]
a \Rightarrow \downarrow 1

3
\begin{align*}
q' & \leftarrow b \land \uparrow 1 \sim \\
\end{align*}
a \Rightarrow \uparrow 1 \neg

T

\begin{align*}
\end{align*}

q
\begin{align*}
q' & \leftarrow q', 0 \\
q & \leftarrow q \\
\end{align*}

a
1

Stéphane Demri, Ranko Lazić
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

\[ q \xrightarrow{a} 1 \]

\[ q' \xrightarrow{b \land 1 \sim} T \]

\[ a \Rightarrow 1 \gamma \]

\[ q, 0 \]

\[ q', 0 \xrightarrow{q'} 0 \]

\[ q', 1 \]

\[ q, 0 \]

\[ q, 1 \]

\[ a \]

\[ \begin{align*}
q & \xrightarrow{a} 1 \\
1 & \xrightarrow{a} q
\end{align*} \]

Stéphane Demri, Ranko Lazić
LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

\[ q \xrightarrow{a} 4 \]
\[ q' \xrightarrow{b \land \uparrow_1 \sim} 3 \]
\[ q' \xrightarrow{a} \top \]

\[ q \xrightarrow{a} q \]
\[ q \xrightarrow{b} q' \]
\[ q \xrightarrow{a} q' \]

Stéphane Demri, Ranko Lazic

LTL with the Freeze Quantifier and Register Automata
LTL with the Freeze Quantifier and Register Automata

Stephane Demri, Ranko Lazic

From RA to ICA

From ICA to LTL with freeze and RA

Complexity results

Characterisation of languages
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

Stéphane Demri, Ranko Lazić
LTL with the Freeze Quantifier and Register Automata
LTL with the Freeze Quantiﬁer and Register Automata

- Expressiveness
- Complexity of satisﬁability
- From RA to ICA
- From ICA to LTL with freeze and RA
- Complexity results
- Characterisation of languages

Stephane Demri, Ranko Lazic

Graphical representation of a deterministic automaton with transitions for symbols $a$, $b$, and $\mathbf{end}$. Transitions include: $q \xrightarrow{a} 1$, $q' \xrightarrow{b} \sim 1$, $q \xrightarrow{a} 1$, and $q \xrightarrow{b} q'$. The automaton transitions through states $q$, $q'$, and $q''$, with an end state $T$.
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

\begin{itemize}
\item \textbf{Stephane Demri, Ranko Lazic}
\item \textbf{LTL with the Freeze Quantifier and Register Automata}
\end{itemize}
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability
From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

Stephane Demri, Ranko Lazic
LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

Stéphane Demri, Ranko Lazic
LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

Stéphane Demri, Ranko Lazić

LTL with the Freeze Quantifier and Register Automata

Graphical representation of a state transition diagram with states and transitions labeled with formulas and actions.
Stephane Demri, Ranko Lazic

LTL with the Freeze Quantifier and Register Automata
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

\[
\begin{array}{c}
q \\
4 \\
\text{end} \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
a \Rightarrow \downarrow 1 \\
3 \\
\end{array} \quad \begin{array}{c}
b \wedge \uparrow 1 \sim \\
q' \\
\end{array} \quad \rightarrow \quad \begin{array}{c}
a \Rightarrow \uparrow 1 \checkmark \\
T \\
\end{array}
\]

\[
\begin{array}{cccccccc}
0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 1 & \rightarrow & 0 & \rightarrow & 1 & \rightarrow & 0 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
a & a & b & b & a & b \\
\end{array}
\]

Stéphane Demri, Ranko Lazić
LTL with the Freeze Quantifier and Register Automata
LTL with the Freeze Quantifier and Register Automata

Stéphane Demri, Ranko Lazic

0 → 1 → 2 → 1 → 0 → 1 → 0

a a b b a b
Infinite words

\[ G(a \Rightarrow \downarrow_1 X((a \Rightarrow \neg \uparrow_1 \sim) U (b \land \uparrow_1 \sim))) \]

We should accept:

\[ a\ a\ b\ a\ b\ a\ b\ a\ b\ \ldots \]

but reject:

\[ a\ a\ b\ a\ b\ a\ a\ a\ a\ a\ \ldots \]
Stephane Demri, Ranko Lazic

LTL with the Freeze Quantifier and Register Automata
\[
\begin{align*}
\text{LTL}^1_1(X, \mathcal{U}) \text{ SAT} & \xrightarrow{\text{LogSpace}} \text{1ARA}_1 \xrightarrow{\neg\text{EMP}} \text{ICA} \xrightarrow{\neg\text{EMP}} \\
\text{LTL}^1_1(X, F) \text{ SAT} & \xrightarrow{\text{LogSpace}} \text{1NRA}_1 \xrightarrow{\neg\text{UNI}} \\
\text{LogSpace} & \xrightarrow{\text{PSPACE}} \\
\end{align*}
\]
Proof.
Encode computations of ICA as data words:

\[
\begin{array}{c}
\text{inc.c inc.c dec.c inc.c dec.c dec.c iszero.c} \\
\end{array}
\]
Proof.
Encode computations of ICA as data words:

\[
\begin{array}{ccccccc}
\text{inc.c} & \text{inc.c} & \text{dec.c} & \text{inc.c} & \text{dec.c} & \text{dec.c} & \text{iszero.c} \\
\hline
\end{array}
\]
Proof.
Encode computations of ICA as data words:

\[
\text{inc.c inc.c dec.c inc.c dec.c dec.c iszero.c}
\]
LTL with freeze
Register automata
Expressiveness
Complexity of satisfiability

From RA to ICA
From ICA to LTL with freeze and RA
Complexity results
Characterisation of languages

\[
\begin{align*}
\text{LTL}_1^{1}(X, U) \text{ SAT} & \quad \text{LogSpace} \quad \text{LTL}_1^{1}(X, F) \text{ SAT} \\
1\text{ARA}_1 \neg\text{EMP} & \quad \text{LogSpace} \quad 1\text{NRA}_1 \neg\text{UNI} \\
\text{LogSpace} & \quad \text{PSpace} \quad \text{LogSpace} \\
\text{ICA} \neg\text{EMP} & 
\end{align*}
\]
### Theorem

<table>
<thead>
<tr>
<th></th>
<th>Minsky CA</th>
<th>Incrementing CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg EMP_{&lt;\omega}^\omega$</td>
<td>$\Sigma^0_1$-complete</td>
<td>$R$ (a), not $PR$ (b)</td>
</tr>
<tr>
<td>$\neg EMP_{\omega}$</td>
<td>$\Sigma^1_1$-complete</td>
<td>$\Pi^0_1$-complete (c)</td>
</tr>
</tbody>
</table>

### Proof.

(a) Reverse computations, and obtain Reset Petri net Coverability [Dufourd, Finkel & Schnoebelen, ICALP ’98].

(b) Reverse computations, and adapt [Schnoebelen, IPL ’02]: Reachability is not $PR$ for Lossy Channel Systems.

(c) Adapt [Ouaknine & Worrell, FoSSaCS ’06]: Recur. State for Insertion Chan. Mach. with Empt. Test. □
[Kaminski & Francez, TCS ’94]

‘…it is very likely that
[decidability of language containment of 1NRA in 1NRA₁]
can be extended to infinite data words.’

[French, TIME ’03], [Demri, Lazić & Nowak, TIME ’05],
[Lisitsa & Potapov, TIME ’05]:

<table>
<thead>
<tr>
<th>Registers</th>
<th>SAT$_{&lt;\omega}$</th>
<th>2</th>
<th>SAT$_{\omega}$</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTL$(X, F)$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>LTL$(X, U)$</td>
<td>$\Sigma^0_1$-comp.</td>
<td></td>
<td>$\Sigma^1_1$-comp.</td>
<td></td>
</tr>
<tr>
<td>LTL$(X, F, F^{-1})$</td>
<td></td>
<td></td>
<td>$\Sigma^1_1$-comp.</td>
<td></td>
</tr>
</tbody>
</table>
[Demri & Lazić, LICS ’06]
$1_{\text{NRA}_1 \neg \text{UNI}^\omega}$ is $\Pi_1^0$-complete.

[French, TIME ’03], [Demri, Lazić & Nowak, TIME ’05], [Lisitsa & Potapov, TIME ’05]:

<table>
<thead>
<tr>
<th>Registers</th>
<th>SAT$^{&lt;\omega}$</th>
<th>2</th>
<th>1</th>
<th>SAT$^\omega$</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{LTL}^\downarrow(X, F)$</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$\text{LTL}^\downarrow(X, U)$</td>
<td>$\Sigma_1^0$-comp.</td>
<td></td>
<td></td>
<td>$\Sigma_1^1$-comp.</td>
<td></td>
</tr>
<tr>
<td>$\text{LTL}^\downarrow(X, F, F^{-1})$</td>
<td></td>
<td></td>
<td></td>
<td>$\Sigma_1^1$-comp.</td>
<td></td>
</tr>
</tbody>
</table>
[Demri & Lazić, LICS '06]

\( 1\text{NRA}_1 \sim \text{UNI}^\omega \) is \( \Pi^0_1 \)-complete.

[French, TIME '03], [Demri, Lazić & Nowak, TIME '05], [Lisitsa & Potapov, TIME '05], [Demri & Lazić, LICS '06]:

<table>
<thead>
<tr>
<th>Registers</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 1 )</th>
<th>( 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{LTL}^\downarrow(X, F) )</td>
<td>( R \setminus \text{PR} )</td>
<td>( \Sigma^0_1 )-comp.</td>
<td>( \Pi^0_1 )-comp.</td>
<td>( \Sigma^1_1 )-comp.</td>
</tr>
<tr>
<td>( \text{LTL}^\downarrow(X, U) )</td>
<td>( R \setminus \text{PR} )</td>
<td>( \Sigma^0_1 )-comp.</td>
<td>( \Pi^0_1 )-comp.</td>
<td>( \Sigma^1_1 )-comp.</td>
</tr>
<tr>
<td>( \text{LTL}^\downarrow(X, F, F^{-1}) )</td>
<td>( \Sigma^0_1 )-comp.</td>
<td>( \Sigma^0_1 )-comp.</td>
<td>( \Sigma^1_1 )-comp.</td>
<td>( \Sigma^1_1 )-comp.</td>
</tr>
</tbody>
</table>
Characterisation of languages

Corollary

\[
\{ L(C) : C \text{ an Incrementing CA} \} = \{ f(\text{str}(L(\phi))) : f \text{ a homomorphism, } \phi \text{ in } \text{LTL}^1_1(X,F) \} = \{ f(\text{str}(L(\phi))) : f \text{ a homomorphism, } \phi \text{ in } \text{LTL}^1_1(X,U) \}
\]