Deterministic ω-Automata for LTL: A safraless, compositional, and mechanically verified construction

Javier Esparza\textsuperscript{1}  Jan Křetínský\textsuperscript{2}  Salomon Sickert\textsuperscript{1}

\textsuperscript{1}Fakultät für Informatik, Technische Universität München, Germany

\textsuperscript{2}IST Austria

May 11, 2015
Deterministic $\omega$-Automata for LTL:
A safraless, compositional, and mechanically verified construction
Deterministic $\omega$-Automata for LTL:
A safraless, compositional, and mechanically verified construction
Deterministic $\omega$-Automata for LTL:
A safrless, compositional, and mechanically verified construction

System with stochasticity and non-determinism expressed as a
Markov decision process $\mathcal{M}$

Product $\mathcal{M} \times \mathcal{R}$ to be analysed

Linear time property expressed as an
LTL formula $\varphi$

Non-deterministic Büchi automaton $\mathcal{B}$

Deterministic Rabin automaton $\mathcal{R}$
Deterministic $\omega$-Automata for LTL:
A safrless, compositional, and mechanically verified construction

System with stochasticity and non-determinism expressed as a Markov decision process $M$

Linear time property expressed as an

- LTL formula $\varphi$
- Non-deterministic Büchi automaton $B$
- Deterministic Rabin automaton $R$

Product $M \times R$ to be analysed

Safra
Deterministic $\omega$-Automata for LTL:
A safrless, compositional, and mechanically verified construction

System with stochasticity and non-determinism
expressed as a

Markov decision process $\mathcal{M}$

Product $\mathcal{M} \times \mathcal{R}$ to be analysed

Linear time property expressed as an

LTL formula $\varphi$

Deterministic (transition-based) Generalised Rabin automaton $\mathcal{R}$
Deterministic $\omega$-Automata for LTL:
A sahraless, compositional, and mechanically verified construction
Deterministic $\omega$-Automata for LTL:
A safranless, compositional, and mechanically verified construction

- Directly yields a deterministic system
Deterministic $\omega$-Automata for LTL:
A sahraless, compositional, and mechanically verified construction

- Directly yields a deterministic system
- Product of several automata
Deterministic $\omega$-Automata for LTL:
A sahraless, compositional, and mechanically verified construction

- Directly yields a deterministic system
- Product of several automata
- Logical structure of the input formula is preserved
  - e.g.: “Which $\mathbf{G}$-subformulae are eventually true?”
Deterministic $\omega$- Automata for LTL:
A safraless, compositional, and mechanically verified construction

- Directly yields a deterministic system
- Product of several automata
- Logical structure of the input formula is preserved
  - e.g.: “Which $\mathbf{G}$-subformulae are eventually true?”
- Smaller Systems\(^1\)

\(^1\)In most cases according to our experimental data; compared to the standard approach
Deterministic \( \omega \)-Automata for LTL: 
A safraless, compositional, and mechanically verified construction

- Directly yields a deterministic system
- Product of several automata
- Logical structure of the input formula is preserved
  - e.g.: “Which \( G \)-subformulae are eventually true?”
- Smaller Systems\(^1\)
- Bonus: Construction and correctness theorem verified in Isabelle/HOL

\(^1\)In most cases according to our experimental data; compared to the standard approach
Deterministic $\omega$-Automata for LTL:
A safraless, compositional, and mechanically verified construction

- Directly yields a deterministic system
- Product of several automata
- Logical structure of the input formula is preserved
  - e.g.: “Which $\mathbf{G}$-subformulae are eventually true?”
- Smaller Systems\(^1\)
- Bonus: Construction and correctness theorem verified in Isabelle/HOL with code extraction 50% done

\(^1\)In most cases according to our experimental data; compared to the standard approach
Experimental Data

\[ \bigwedge_{i \in \{1, \ldots, n\}} GFa_i \Rightarrow GFb_i \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>NBA</th>
<th>DRA</th>
<th>DTGRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LTL2BA</td>
<td>ltl2dstar</td>
<td>Rabinizer 3</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Experimental Data

\[ \bigwedge_{i \in \{1, \ldots, n\}} GF a_i \Rightarrow GF b_i \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>NBA</th>
<th>DRA</th>
<th>DTGRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LTL2BA</td>
<td>ltl2dstar</td>
<td>Rabinizer 3</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>&gt; 10^4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>&gt; 10^6</td>
<td></td>
</tr>
</tbody>
</table>
\[ \bigwedge_{i \in \{1, \ldots, n\}} GFa_i \Rightarrow GFb_i \]

<table>
<thead>
<tr>
<th>( n )</th>
<th>NBA</th>
<th>DRA</th>
<th>DTGRA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>( &gt; 10^4 )</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>( &gt; 10^6 )</td>
<td>1</td>
</tr>
</tbody>
</table>
An ω-word is an infinite sequence: $w = a_0 a_1 a_2 a_3 \ldots$
An $\omega$-word is an infinite sequence: $w = a_0 a_1 a_2 a_3 \ldots$.

**Definition (LTL Semantics, Negation-Normal-Form)**

<table>
<thead>
<tr>
<th>$\square$</th>
<th>$\neg$</th>
<th>$\wedge$</th>
<th>$\vee$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\models$</td>
<td>$\models$</td>
<td>$\models$</td>
<td>$\models$</td>
</tr>
<tr>
<td>$\alpha$ set word</td>
<td>$\to$</td>
<td>$\alpha$ ltl</td>
<td>$\to$</td>
</tr>
<tr>
<td>$w$</td>
<td>$tt$</td>
<td>$= True$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$ff$</td>
<td>$= False$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$a$</td>
<td>$= a \in w_0$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$\neg a$</td>
<td>$= a \notin w_0$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$\phi \wedge \psi$</td>
<td>$= w \models \phi \wedge w \models \psi$</td>
<td></td>
</tr>
<tr>
<td>$w$</td>
<td>$\phi \vee \psi$</td>
<td>$= w \models \phi \vee w \models \psi$</td>
<td></td>
</tr>
</tbody>
</table>
**ω-Words and LTL**

An $\omega$-word is an infinite sequence: $w = a_0 a_1 a_2 a_3 . . . .$

**Definition (LTL Semantics, Negation-Normal-Form)**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Box$</td>
<td>$\Box$</td>
</tr>
<tr>
<td>$w \models tt$</td>
<td>$True$</td>
</tr>
<tr>
<td>$w \models ff$</td>
<td>$False$</td>
</tr>
<tr>
<td>$w \models a$</td>
<td>$a \in w_0$</td>
</tr>
<tr>
<td>$w \models \neg a$</td>
<td>$a \notin w_0$</td>
</tr>
<tr>
<td>$w \models \varphi \land \psi$</td>
<td>$w \models \varphi \land w \models \psi$</td>
</tr>
<tr>
<td>$w \models \varphi \lor \psi$</td>
<td>$w \models \varphi \lor w \models \psi$</td>
</tr>
<tr>
<td>$w \models F\varphi$</td>
<td>$\exists k.; w_{k\infty} \models \varphi$</td>
</tr>
<tr>
<td>$w \models G\varphi$</td>
<td>$\forall k.; w_{k\infty} \models \varphi$</td>
</tr>
<tr>
<td>$w \models \psi U \varphi$</td>
<td>$\exists k.; w_{k\infty} \models \varphi \land \forall j &lt; k.; w_{j\infty} \models \psi$</td>
</tr>
<tr>
<td>$w \models X\varphi$</td>
<td>$w_{1\infty} \models \varphi$</td>
</tr>
</tbody>
</table>
\(\omega\)-Words and LTL

An \(\omega\)-word is an infinite sequence: \(w = a_0a_1a_2a_3\ldots\).

**Definition (LTL Semantics, Negation-Normal-Form)**

\[
\begin{align*}
\Box \models \Box & \quad :: \; \alpha \text{ set word} \rightarrow \alpha \text{ ltl} \rightarrow \mathbb{B} \\
w \models \top & = \; \text{True} \\
w \models \bot & = \; \text{False} \\
w \models a & = \; a \in w_0 \\
w \models \neg a & = \; a \notin w_0 \\
w \models \varphi \land \psi & = \; w \models \varphi \land w \models \psi \\
w \models \varphi \lor \psi & = \; w \models \varphi \lor w \models \psi \\
w \models F\varphi & = \; \exists k. \, w_{k\infty} \models \varphi \; \checkmark \\
w \models G\varphi & = \; \forall k. \, w_{k\infty} \models \varphi \\
w \models \psi U\varphi & = \; \exists k. \, w_{k\infty} \models \varphi \land \forall j < k. \, w_{j\infty} \models \psi \; \checkmark \\
w \models X\varphi & = \; w_{1\infty} \models \varphi \; \checkmark 
\end{align*}
\]
An \( \omega \)-word is an infinite sequence: \( w = a_0a_1a_2a_3 \ldots \).

### Definition (LTL Semantics, Negation-Normal-Form)

\[
\begin{align*}
\Box & \models \Box \quad \therefore \quad \alpha \text{ set word} \rightarrow \alpha \text{ ltl} \rightarrow \mathbb{B} \\
w & \models \text{tt} \quad = \quad \text{True} \\
w & \models \text{ff} \quad = \quad \text{False} \\
w & \models a \quad = \quad a \in w_0 \\
w & \models \lnot a \quad = \quad a \notin w_0 \\
w & \models \varphi \land \psi \quad = \quad w \models \varphi \land w \models \psi \\
w & \models \varphi \lor \psi \quad = \quad w \models \varphi \lor w \models \psi \\
w & \models \mathbf{F} \varphi \quad = \quad \exists k. w_{k\infty} \models \varphi \quad \checkmark \\
w & \models \mathbf{G} \varphi \quad = \quad \forall k. w_{k\infty} \models \varphi \quad \times \\
w & \models \psi \mathbf{U} \varphi \quad = \quad \exists k. w_{k\infty} \models \varphi \land \forall j < k. w_{j\infty} \models \psi \quad \checkmark \\
w & \models \mathbf{X} \varphi \quad = \quad w_{1\infty} \models \varphi \quad \checkmark
\end{align*}
\]
Unfolding Modal Operators

\[
\begin{align*}
F\varphi & \equiv XF\varphi \lor \varphi \\
G\varphi & \equiv XG\varphi \land \varphi \\
\psi U\varphi & \equiv \varphi \lor (\psi \land X(\psi U\varphi))
\end{align*}
\]
Co-Büchi Automata for $G$-free $\varphi$

$$\varphi = a \lor (b \mathbf{U} c)$$
Co-Büchi Automata for $G$-free $\varphi$

$$\varphi = a \lor (b \mathbf{U} c)$$
Co-Büchi Automata for $G$-free $\varphi$

$$\varphi = a \lor (b \mathbin{U} c)$$

$$\varphi \rightarrow a \lor c \lor (b \land \mathbf{X}(b \mathbin{U} c))$$
Co-Büchi Automata for $\mathbf{G}$-free $\varphi$

\[ \varphi = a \lor (b \mathbf{U} c) \]

\[ \varphi \rightarrow a \lor c \lor (b \land X(b \mathbf{U} c)) \rightarrow \bar{a}b\bar{c} \ b\mathbf{U}c \]
Co-Büchi Automata for \( G \)-free \( \varphi \)

\[
\varphi = a \lor (b \mathbf{U} c)
\]

\[
\varphi \rightarrow a \lor c \lor (b \land X(b \mathbf{U} c)) \rightarrow \bar{a}b\bar{c} b\mathbf{U}c
\]
Relaxed case: $FG\varphi$

- $w \models FG\varphi$ iff $w_{i\infty} \models \varphi$ for almost all $i$

Reason: $G$-subformulae may be nested inside $X, F, U$. 
Automata for $FG_\varphi$ where $\varphi$ is $G$-free

$W = \ldots$

$\begin{array}{c}
q_2 \\
\downarrow b\bar{c} \\
q_3 \\
\downarrow c \\
\end{array}$

$\begin{array}{c}
q_2 \\
\downarrow \bar{a}c \bar{b}c \\
\end{array}$

$\begin{array}{c}
q_4 \\
\downarrow \bar{a}b\bar{c} \\
\end{array}$

$\begin{array}{c}
a + \bar{a}c \\
\end{array}$
Automata for $\mathbf{FG}_\varphi$ where $\varphi$ is $\mathbf{G}$-free

$w = abc \ldots$
Automata for $\mathbf{FG}_\varphi$ where $\varphi$ is $\mathbf{G}$-free

\[ w = abc \ldots \]
Automata for $\mathbf{FG}_\varphi$ where $\varphi$ is $\mathbf{G}$-free

$$w = abc \bar{a}b\bar{c} \ldots$$
Automata for $\text{FG}_\varphi$ where $\varphi$ is $\text{G}$-free

$$w = abc \bar{a}b\bar{c} \bar{a}b\bar{c} \ldots$$
Automata for $FG_\varphi$ where $\varphi$ is $G$-free

$$w = abc \bar{a}b\bar{c} \bar{a}b\bar{c} \ldots$$
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function. Deterministic Mojmir automata are “blind” to events that only happen finitely often.
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function.
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function.

Deterministic
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function.

- Deterministic
- Accepts an \( \omega \)-word \( w \) iff almost all tokens reach the final states
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function.

Deterministic

Accepts an $\omega$-word $w$ iff almost all tokens reach the final states

- Mojmir automata are “blind” to events that only happen finitely often
In every step a new token is placed in the initial state and all other tokens are moved according to the transition function.

Deterministic

Accepts an $\omega$-word $w$ iff almost all tokens reach the final states

Mojmir automata are “blind” to events that only happen finitely often
From Mojmir to Rabin Automata
Going Further

- From Mojmir to Rabin Automata
  - Unbounded number of tokens?
From Mojmir to Rabin Automata

- Unbounded number of tokens?
  Abstraction with ranking functions for states and tokens
Going Further

- From Mojmir to Rabin Automata
  - Unbounded number of tokens?
    - Abstraction with ranking functions for states and tokens
  - Mojmir acceptance ($\forall\infty$) vs. Rabin acceptance (finite, $\exists\infty$)?
From Mojmir to Rabin Automata

- Unbounded number of tokens?
  Abstraction with ranking functions for states and tokens
- Mojmir acceptance (\(\forall\)) vs. Rabin acceptance (finite, \(\exists\))? Alternative definition for Mojmir acceptance
Going Further

- From Mojmir to Rabin Automata
  - Unbounded number of tokens?
    Abstraction with ranking functions for states and tokens
  - Mojmir acceptance ($\forall\infty$) vs. Rabin acceptance (finite, $\exists\infty$)?
    Alternative definition for Mojmir acceptance

- Mojmir Automata for $\text{FG}\varphi$ for arbitrary $\varphi$
Going Further

- From Mojmir to Rabin Automata
  - Unbounded number of tokens?
    Abstraction with ranking functions for states and tokens
  - Mojmir acceptance ($\forall^\infty$) vs. Rabin acceptance (finite, $\exists^\infty$)?
    Alternative definition for Mojmir acceptance

- Mojmir Automata for $\text{FG}\varphi$ for arbitrary $\varphi$
  - Divide-and-conquer approach
  - Construct for every $G$-subformula a separate automaton
  - Instead of expanding $G$’s rely on the other automata
  - Intersection and Union of several Mojmir Automata
Overview of the Construction

- LTL
  - Master-Transition-System
    - Acceptance:
      1. Guess the set of eventually true G-subformulae
      2. Verify this guess using the Mojmir automata
      3. Accept iff almost all the time this guess entails the current state of the master-transition-system

- Mojmir
  - G-subformulae
  - Product
    - Generalised Rabin

- Rabin
The Master-Transition-System tracks a finite prefix of the $\omega$-word.
The Master-Transition-System tracks a finite prefix of the $\omega$-word.

Acceptance:
1. Guess the set of eventually true $G$-subformulae
2. Verify this guess using the Mojmir automata
3. Accept iff almost all the time this guess entails the current state of the master-transition-system
The presented translation . . .

- preservers the logical structure of the formula
- is compositional
  - Aggressive optimization can lead to huge space savings
  - Some optimizations are already verified
- yields small deterministic $\omega$-automata
Conclusion and Future Work

The presented translation . . .
- preservers the logical structure of the formula
- is compositional
  - Aggressive optimization can lead to huge space savings
  - Some optimizations are already verified
- yields small deterministic $\omega$-automata

Open Problems:
- Explore and formalize further optimizations
- Adapt construction to support:
  - Alternation-free linear-time $\mu$-calculus (contains LTL)
  - Parity automata

Isabelle/HOL Formalisation
- To be submitted to the “Archive of Formal Proofs” - afp.sourceforge.net
- Available on request: sickert@in.tum.de

From LTL to Deterministic Automata: A Safraless Compositional Approach

Javier Esparza and Jan Křetínský†
Institut für Informatik, Technische Universität München, Germany
IST Austria
Getting More Information

- Isabelle/HOL Formalisation
  - To be submitted to the “Archive of Formal Proofs” - afp.sourceforge.net
  - Available on request: sickert@in.tum.de

Thank you for your attention!