On the Relationship Between LTL Normal Form and Büchi Automata

Jianwen Li

Software Engineering Institute,
East China Normal University

28 April 2013

Background (1)

- Question: How to verify the system has the desired property ?
- Model Checking
- Automata theory
- Linear Temporal Logic (LTL)

Background (2)

- The translation from LTL to Büchi automata is the key issue in LTL model checking.

Outline

- Preliminaries
- Motivation
- LTL Normal Form
- LTL Transition System (LTS)
- Obligation Set
- Büchi Construction
- Experiments
- Conclusion

Preliminaries (1)

Let AP be a set of atomic properties, then the syntax of LTL formulas is defined by:

$$\phi \ ::= \ \mathsf{tt} \mid \mathsf{ff} \mid a \mid \neg a \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi U \phi \mid \phi R \ \phi \mid X \phi$$

The semantics of temporal operators with respect to the run ξ is given by:

- $\xi \models \alpha$ iff $\xi^1 \models \alpha$, here α is an propositional formula;
- $\xi \models \phi_1 \ U \ \phi_2$ iff there exists $i \geqslant 0$ such that $\xi_i \models \phi_2$ and for all $0 \leqslant j < i, \xi_j \models \phi_1$;
- $\xi \models \phi_1 \ R \ \phi_2$ iff either $\xi_i \models \phi_2$ for all $i \geq 0$, or there exists $i \geq 0$ with $\xi_i \models \phi_1 \wedge \phi_2$ and $\xi_j \models \phi_2$ for all $0 \leq j < i$;
- $\xi \models X \ \phi$ iff $\xi_1 \models \phi$.

Preliminaries (2)

Definition (Büchi Automata)

A Büchi automaton is a tuple $\mathcal{A} = (S, \Sigma, \delta, S_0, F)$, where S is a finite set of states, Σ is a finite set of alphabet symbols , $\delta : S \times \Sigma \to 2^S$ is the transition relation, S_0 is a set of initial states, and $F \subseteq S$ is a set of accepting states of $\mathcal{A}$.
An infinite word $\xi = \omega_0\omega_1\dots$ is accepted by $\mathcal{A}$ iff it will run across one of the accepting states in F infinitely often.

Preliminaries (2)

Definition (Büchi Automata)

A Büchi automaton is a tuple $\mathcal{A} = (S, \Sigma, \delta, S_0, F)$, where S is a finite set of states, Σ is a finite set of alphabet symbols , $\delta : S \times \Sigma \rightarrow 2^S$ is the transition relation, S_0 is a set of initial states, and $F \subseteq S$ is a set of accepting states of $\mathcal{A}$.
An infinite word $\xi = \omega_0\omega_1 \ldots$ is accepted by $\mathcal{A}$ iff it will run across one of the accepting states in F infinitely often.

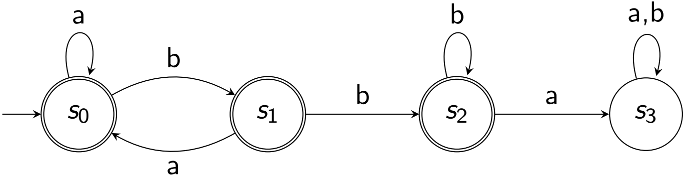


Figure: An Büchi automaton example.

Motivation

All start from the two equations below:

$$\phi U \psi \equiv \psi \vee \phi \wedge X(\phi U \psi);$$
$$\phi R \psi \equiv (\phi \wedge \psi) \vee \psi \wedge X(\phi R \psi).$$

LTL Normal Form (1)

Definition (Normal Form Expansion)

The *normal form* of an LTL formula ϕ, denoted as $NF(\phi)$, is :

1. $NF(\phi) = \{\phi \wedge X(\mathrm{tt})\}$ if $\phi \not\equiv \mathrm{ff}$ is a propositional formula. If $\phi \equiv \mathrm{ff}$, we define $NF(\mathrm{ff}) = \emptyset$;
2. $NF(X\phi) = \{\mathrm{tt} \wedge X(\psi) \mid \psi \in DF(\phi)\}$;
3. $NF(\phi_1 U \phi_2) = NF(\phi_2) \cup NF(\phi_1 \wedge X(\phi_1 U \phi_2))$;
4. $NF(\phi_1 R \phi_2) = NF(\phi_1 \wedge \phi_2) \cup NF(\phi_2 \wedge X(\phi_1 R \phi_2))$;
5. $NF(\phi_1 \vee \phi_2) = NF(\phi_1) \cup NF(\phi_2)$;
6. $NF(\phi_1 \wedge \phi_2) = \{(\alpha_1 \wedge \alpha_2) \wedge X(\psi_1 \wedge \psi_2) \mid \forall i = 1, 2.\ \alpha_i \wedge X(\psi_i) \in NF(\phi_i)\}$;

LTL Normal Form (2)

Example

- $NF(aUb) = \{b \wedge X\text{tt}, a \wedge X(aUb)\}$ // $aUb \equiv b \vee a \wedge X(aUb)$;
- Let $\phi_1 = G(bUc \wedge dUe)$, then

  $NF(\phi_1) = \{c \wedge e \wedge X\phi_1, b \wedge e \wedge X\phi_2, c \wedge d \wedge X\phi_3, b \wedge d \wedge X\phi_4\}$:

  here $\phi_2 = bUc \wedge \phi_1$, $\phi_3 = dUe \wedge \phi_1$, and $\phi_4 = bUc \wedge dUe \wedge \phi_1$.

LTL Transition System (LTS) (1)

Definition (LTL Transition System)

The labelled transition system T_ϕ generated from the formula ϕ is a tuple $\langle \Sigma, S_\phi, \rightarrow, \phi \rangle$ where ϕ is the initial state, and:

1. the transition relation $\rightarrow$ is defined by: $\psi_1 \xrightarrow{\alpha} \psi_2$ iff there exists $\alpha \wedge X(\psi_2) \in NF(\psi_1)$;
2. S_ϕ is the smallest set of formulas such that $\phi \in S_\phi$, and inductively $\psi_1 \in S_\phi$ and $\psi_1 \xrightarrow{\alpha} \psi_2$ implies $\psi_2 \in S_\phi$.

LTL Transition System (LTS) (2)

Example

- aUb:
  1. $NF(aUb) = \{b \wedge X\text{tt}, a \wedge X(aUb)\}$;
  2. $NF(\text{tt}) = \text{tt} \wedge X(\text{tt})$.

LTL Transition System (LTS) (2)

Example

- $\phi_1 = G(bUc \wedge dUe)$:
  1. $NF(\phi_1) = \{c \wedge e \wedge X\phi_1, b \wedge e \wedge X\phi_2, c \wedge d \wedge X\phi_3, b \wedge d \wedge X\phi_4\}$: here $\phi_2 = bUc \wedge \phi_1$, $\phi_3 = dUe \wedge \phi_1$, and $\phi_4 = bUc \wedge dUe \wedge \phi_1$.
  2. $NF(\phi_2) = NF(\phi_3) = NF(\phi_4)$.

Obligation Set (1)

Definition (Obligation Set)

For a formula ϕ, we define its obligation set, denoted by $Olg(\phi)$, as follows:

1. $Olg(\text{tt}) = \{\emptyset\}$ and $Olg(\text{ff}) = \{\{\text{ff}\}\}$;
2. If ϕ is a literal, $Olg(\phi) = \{\{\phi\}\}$;
3. If $\phi = X\psi$, $Olg(\phi) = Olg(\psi)$;
4. If $\phi = \psi_1 \vee \psi_2$, $Olg(\phi) = Olg(\psi_1) \cup Olg(\psi_2)$;
5. If $\phi = \psi_1 \wedge \psi_2$, $Olg(\phi) = \{O_1 \cup O_2 \mid O_1 \in Olg(\psi_1) \wedge O_2 \in Olg(\psi_2)\}$;
6. If $\phi = \psi_1 U \psi_2$ or $\psi_1 R \psi_2$, $Olg(\phi) = Olg(\psi_2)$;

For $O \in Olg(\phi)$, we refer to it as an *obligation* of ϕ.

Obligation Set (2)

Example

- $Olg(aUb) = \{\{b\}\}$;
- $Olg(G(bUc \wedge dUe)) = \{\{c, e\}\}$;
- $Olg(G(bUc \vee dUe)) = \{\{c\}, \{e\}\}$.

Büchi Construction (1)

Definition ($\mathcal{A}_\phi$ for Release/Until-free formulas)

For a Release/Until-free formula ϕ, we define the Büchi automaton $\mathcal{A}_\phi = (S, \Sigma, \rho, S_0, F)$ where $T_\phi = \langle \Sigma, S, \delta, S_0 \rangle$, and

- $s_2 \in \rho(s_1, \omega)$ iff there exists $s_2 \in \delta(s_1, \alpha)$ and $\omega \models \alpha$;
- The set F is defined by: $F = \{\text{true}\}$ if ϕ is Release-free while $F = S$ if ϕ is Until-free.

Büchi Construction (2)

Definition (Büchi Automaton $\mathcal{A}_\phi$)

The Büchi automaton for the formula ϕ is defined as $\mathcal{A}_\phi = (\Sigma, S, \delta, S_0, \mathcal{F})$, where $\Sigma = 2^{AP}$ and:

- $S = \{\langle \psi, P \rangle \mid \psi \in S_\phi\}$ is the set of states;
- $S_0 = \{\langle \phi, \emptyset \rangle\}$ is the set of initial states;
- $\mathcal{F} = \{\langle \psi, \emptyset \rangle \mid \psi \in S_\phi\}$ is the set of accepting states;
- Let states s_1, s_2 with $s_1 = \langle \psi_1, P_1 \rangle$, $s_2 = \langle \psi_2, P_2 \rangle$ and $w \subseteq 2^{AP}$. Then, $s_2 \in \delta(s_1, \omega)$ iff there exists $\psi_1 \xrightarrow{\alpha} \psi_2$ with $\omega \models \alpha$ such that the corresponding P_2 is updated by:
  1. $P_2 = \emptyset$ if $\exists O \in Olg_{\psi_2} \cdot O \subseteq P_1 \cup CF(\alpha)$,
  2. $P_2 = P_1 \cup CF(\alpha)$ otherwise.

Examples (1)

- aUb

Examples (1)

- *aUb*

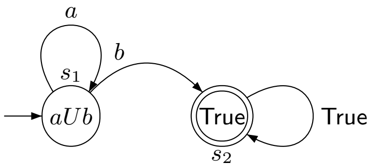


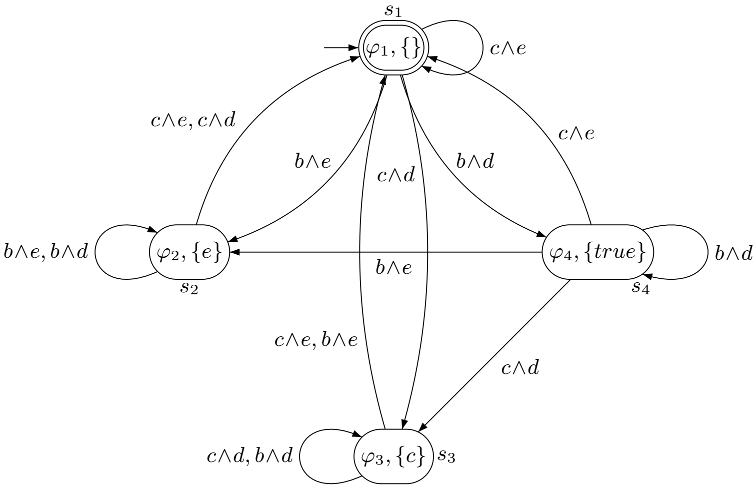
Figure: The Büchi automaton for *aUb*

Examples (2)

- $\mathrm{G}(bUc \wedge dUe)$

Examples (2)

- $\mathrm{G}(bUc \wedge dUe)$



Experiments

Formula Length	States	Transitions	Nondet-States	Nondet	Time	Product States
10	4.32 3.44	17.99 18.22	2.69 1.77	0.75 0.74	0.14 0.03	706 538
20	23.30 6.67	146.73 56.22	4.43 2.84	0.82 0.76	0.14 0.05	4467 1145
30	41.90 10.52	259.15 113.27	16.32 7.62	0.85 0.78	0.14 0.10	8183 1857
40	45.76 20.55	296.05 323.20	20.26 16.84	0.83 0.80	0.06 0.27	8909 3857
50	167.13 43.34	1161.11 744.53	69.52 36.87	0.91 0.86	0.12 2.80	33225 8420

Table: Comparison results between SPOT and *Aalta*. In each formula group (with the same length) the first line displays the results from SPOT while the second from *Aalta*.

Conclusion

Under the LTL Transition System (LTS) framework, we achieve to propose:

1. A new LTL-to-Büchi translation;

Co-authors

- Geguang Pu, East China Normal University;
- Lijun Zhang, Technical University of Denmark;
- Jefing He, East China Normal University;
- Moshe Y. Vardi, Rice University.

Thank you !