Büchi Automata & Model checking

Tomáš Babiak

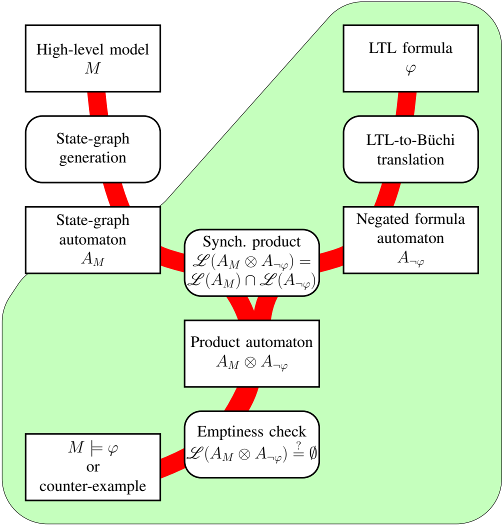
DTEDI

November 11, 2010

Outline

Introduction to Model checking

The automata-theoretic approach to LTL model checking.



Model checking - pros & cons

Advantages:

- General technique applicable on hardware and software.
- Decision process can be fully automatized. (Tools are available.)
- Soundness is proven:
  - If $\mathcal{M} \models \varphi$ then system has the given property.
  - If $\mathcal{M} \not\models \varphi$ then system can violate the given property.
- A counterexample is generated when the property is violated.

Model checking - pros & cons

Advantages:

- General technique applicable on hardware and software.
- Decision process can be fully automatized. (Tools are available.)
- Soundness is proven:
  - If $\mathcal{M} \models \varphi$ then system has the given property.
  - If $\mathcal{M} \not\models \varphi$ then system can violate the given property.
- A counterexample is generated when the property is violated.

Disadvantages:

- Only a model of a system is verified.
- Applicable only on finite state systems.
- Number of states of $\mathcal{A}_{\mathcal{M}}$ is often exponential in the size of implicit description of the system - state explosion problem.

Model checking - partial solutions of the state explosion problem

- abstraction
- partial order reduction
- symetry reduction
- on-the-fly algorithms
- symbolic model checking
- distributed algorithms
- ...

Syntax and Semantics of LTL

Linear Temporal Logic (LTL) is defined by

$$\varphi ::= tt \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2\ \mathsf{X}\varphi \mid \varphi_1\, \mathsf{U}\, \varphi_2$$

where *tt* stands for true and *a* ranges over a countable set *AP* of atomic propositions.

Syntax and Semantics of LTL

Linear Temporal Logic (LTL) is defined by

$$\varphi \; ::= \; \mathit{tt} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \; \mathsf{X}\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

where tt stands for true and a ranges over a countable set AP of atomic propositions.

Abbreviations: $\mathit{ff} \equiv \neg\mathit{tt}$ $\quad \mathsf{F}\varphi \equiv \mathit{tt}\,\mathsf{U}\,\varphi$ $\quad \mathsf{G}\varphi \equiv \neg\mathsf{F}\neg\varphi$

Syntax and Semantics of LTL

Linear Temporal Logic (LTL) is defined by

$$\varphi \ ::= \ tt \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \ \mathsf{X}\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2$$

where tt stands for true and a ranges over a countable set AP of atomic propositions.

Abbreviations: $ff \equiv \neg tt$ $\quad \mathsf{F}\varphi \equiv tt\,\mathsf{U}\,\varphi$ $\quad \mathsf{G}\varphi \equiv \neg\mathsf{F}\neg\varphi$

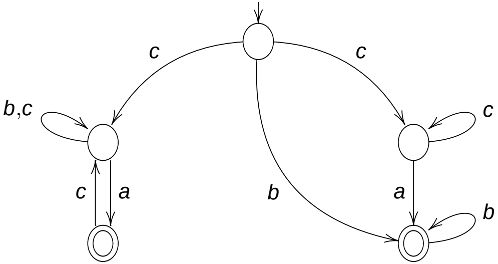
We interpret LTL on infinite words $w \in (2^{AP})^\omega$.

Semantics of modal operators:

$\mathsf{X}\varphi$	**next**	$\bullet\ \varphi\ \bullet\ \bullet\ \bullet\ \ldots$
$\varphi\,\mathsf{U}\,\psi$	**until**	$\varphi\ \varphi\ \ldots\ \varphi\ \psi\ \bullet\ \bullet\ \bullet\ \ldots$
$\mathsf{F}\varphi$	**eventually**	$\bullet\ \bullet\ \ldots\ \bullet\ \varphi\ \bullet\ \bullet\ \bullet\ \ldots$
$\mathsf{G}\varphi$	**always**	$\varphi\ \varphi\ \varphi\ \varphi\ \ldots$

Büchi automata (BA)

Similar to finite automata (FA), but interpreted over infinite words.
Accepts a word w if some accepting state is visited infinitely often during some computation over w.

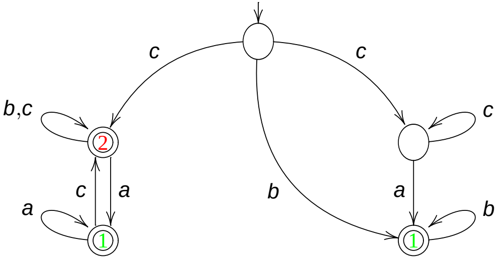


For example:

- Accepts infinite words $cca(b)^\omega$ or $ccc(ca)^\omega$.
- Does not accept infinite word $cacac(c)^\omega$.

Generalized Büchi automata (GBA)

Several sets of accepting states. Accepts a word w if some accepting state of each set is visited infinitely often.

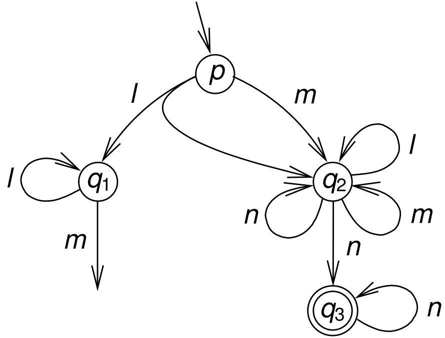


For example:

- Accepts infinite word $cbb(ac)^\omega$.
- Does not accept infinite words $cacac(c)^\omega$ and $cca(b)^\omega$.

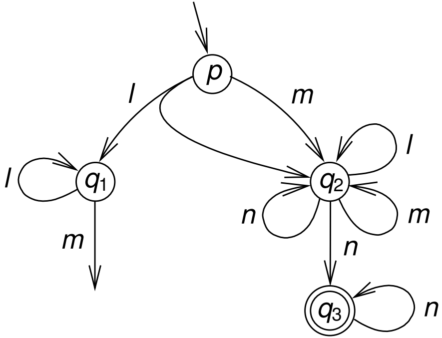
Alternating Büchi automata (GBA)

A run of an alternating BA $\mathcal{A}$ on an infinite word w is a tree. A run is accepting if along any infinite branch some accepting state occurs infinitely often.



Alternating Büchi automata (GBA)

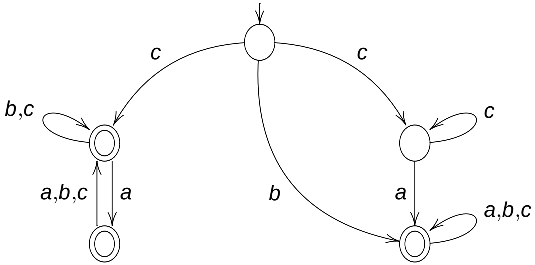
A run of an alternating BA $\mathcal{A}$ on an infinite word w is a tree. A run is accepting if along any infinite branch some accepting state occurs infinitely often.



Accepts the language $l^* m(l + m + n)^* n^\omega$.

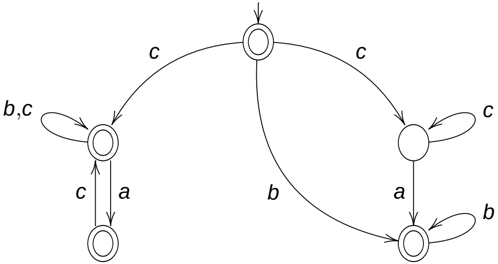
Büchi automata classes

terminal BA = each accepting state have transitions under each input symbol and there is no transition leading from an accepting state to a non-accepting one



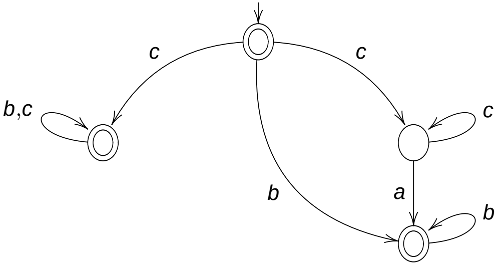
Büchi automata classes

weak BA = each SCC contains only accepting states or only non-accepting states

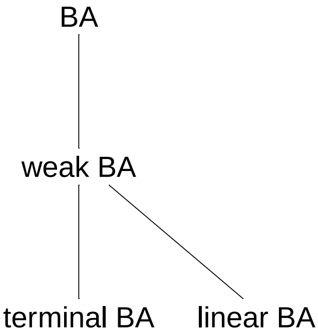


Büchi automata classes

linear BA = **1-weak BA** = **very weak BA** = each SCC contains just one state



Hierarchy of Büchi automata classes



Connections between LTL and BA

Each LTL formula φ can be translated into language equivalent BA $\mathcal{A}_\varphi$ such that the number of states of $\mathcal{A}_\varphi$ is $2^{\mathcal{O}(|\varphi|)}$.

(Wolper, Vardi & Sistla '83)

Connections between LTL and BA

Each LTL formula φ can be translated into language equivalent BA $\mathcal{A}_\varphi$ such that the number of states of $\mathcal{A}_\varphi$ is $2^{\mathcal{O}(|\varphi|)}$.

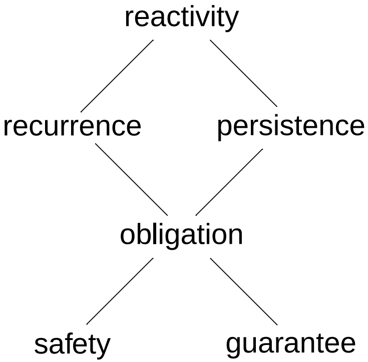
(Wolper, Vardi & Sistla '83)

Several translations of LTL to BA using different intermediate formalisms were developed:

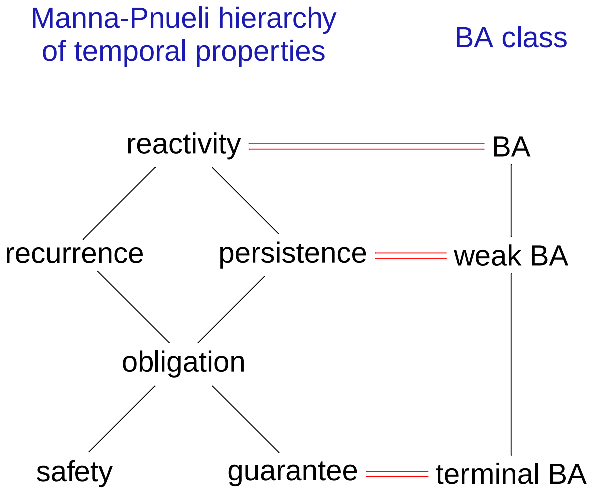
- LTL $\rightarrow$ VWAA $\rightarrow$ BA (Vardi '94)
- LTL $\rightarrow$ GBA $\rightarrow$ BA (Gerth, Peled, Vardi & Wolper '95)
- LTL $\rightarrow$ VWAA $\rightarrow$ TGBA $\rightarrow$ BA (Gastin & Oddoux '01)
- LTL $\rightarrow$ TGBA $\rightarrow$ BA (Giannakopoulou & Lerda '02)

Connections between LTL and BA [ČP2003]

Manna-Pnueli hierarchy
of temporal properties

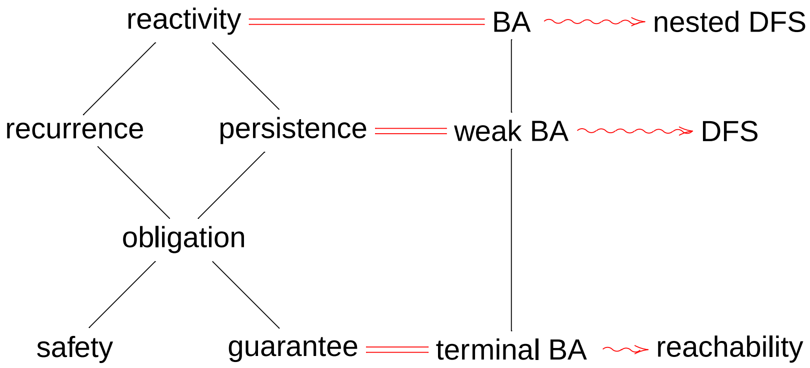


Connections between LTL and BA [ČP2003]



Connections between LTL and BA [ČP2003]

Manna-Pnueli hierarchy of temporal properties | BA class | accepting cycle detection algorithm



Motivation for further LTL→BA transl. improvements

- Contemporary translations are far from perfect.
  (Rozier & Vardi ’07)
- For specific formulae, translation itself may take a significant time of the whole model checking process.
- Quality (i.e. size, determinism) of resulting automaton has impact on the overall model checking performance.
- In past the focus was on the size of the produced automaton. Todays research indicate that determinism of produced automaton has bigger impact on model checking performance than its size.
  (Sebastiani & Tonetta ’03)
  (Geldenhuys & Hansen ’06)

Other types of automata in model checking?

Traditional approach uses BA for LTL model checking. During the translation of LTL$\rightarrow$BA several intermediate automata are produced.

Other types of automata in model checking?

Traditional approach uses BA for LTL model checking. During the translation of LTL$\rightarrow$BA several intermediate automata are produced.

Natural question arises:

Can those automata be directly used for model checking?

Other types of automata in model checking?

Traditional approach uses BA for LTL model checking. During the translation of LTL$\rightarrow$BA several intermediate automata are produced.

Natural question arises:

Can those automata be directly used for model checking?

Example of using TGBA instead of BA: SPOT

(Couvreur '99)

(Couvreur, Duret-Lutz & Poitrenaud '05)

Thank you for your attention.