Tool Support for Learning Büchi Automata and Linear Temporal Logic

Yih-Kuen Tsay
Dept. of Information Management
National Taiwan University

Joint work with Yu-Fang Chen & Kang-Nien Wu

Background

- **Büchi automata** and **linear temporal logic** are two fundamental components of *model checking*, in particular, the automata-theoretic approach:
  - The (finite-state) system is modeled as a Büchi automaton A.
  - A desired behavioral property of the system is given by a linear temporal formula f.
  - Let B_f ($B_{\sim f}$) denote a Büchi automaton equivalent to f ($\sim f$).
  - The model checking problem is essentially asking whether

$$L(A) \subseteq L(B_f) \text{ or equivalently } L(A) \cap L(B_{\sim f}) = \emptyset .$$

- The well-used model checker SPIN, for example, adopted the automata-theoretic approach.

Motivation

- Model checking has proven to be very useful and the number of courses covering related topics appears to be increasing.
- Understanding the correspondence between Büchi automata and linear temporal logic is not easy.
- A graphical interactive tool may be helpful for the learner (and the teacher).
- Tools exist for learning *classic* automata and formal languages, e.g., JFLAP (which inspired our tool GOAL and provided some of its basic building blocks).

Büchi Automata

- Büchi automata (BA) are a variant of omega-automata, which are finite automata operating on *infinite* words.
- A Büchi automaton is given, as in finite automata, by a 5-tuple $(\Sigma, Q, \delta, Q_0, F)$, where $F \subseteq Q$ is the set of accepting states.
- An infinite word $\alpha \in \Sigma^\omega$ is accepted by a Büchi automaton B if there exists a run ρ of B on α satisfying the condition: $Inf(\rho) \cap F \neq \emptyset$
  where $Inf(\rho)$ denotes the set of states occurring infinitely many times in ρ.

Generalized Büchi Automata

- A generalized Büchi automaton (GBA) is like a BA but with $F \subseteq 2^Q$, i.e., $F = \{F_1,\ldots,F_k\}$ where $F_i \subseteq Q$.
- A word $\alpha \in \Sigma^\omega$ is accepted by a generalized Büchi automaton $\mathcal{B}$ if there exists a run ρ of $\mathcal{B}$ on α satisfying the condition:

$$\forall F_i \in F:\ \mathrm{Inf}(\rho) \cap F_i \neq \emptyset$$

About the Alphabet

- To link Büchi automata to temporal formulae, we will consider automata with an alphabet like:
  - {p,~p}
  - {pq,p~q,~pq,~p~q}

Propositional Linear Temporal Logic (PTL)

- A subset of linear temporal logic (LTL).
- PTL formulae are interpreted over an infinite sequence of states, which can be seen as an infinite word over a suitable alphabet like {p,~p} or {pq,p~q,~pq,~p~q}.
- *Every PTL formula is equivalent to some Büchi automaton, but not vice versa.*

Note: Quantified PTL (QPTL) are as expressive as Büchi automata.

Temporal Operators in PTL

- Future temporal operators:
  - next: ( ) or X
  - eventually (sometime): <> or F
  - hence-forth (always): [ ] or G
  - wait-for (unless): $\mathcal{W}$
  - until: $\mathcal{U}$
- Past temporal operators:
  - previous: (-) or Y
  - before: (~) or Z
  - once: <-> or O
  - so-far: [-] or H
  - back-to: $\mathcal{B}$
  - since: $\mathcal{S}$

Semantics of Future Operators

Let π be an infinite sequence of states.

- $(\pi, i) \models (\,)f$ iff $(\pi, i+1) \models f$
- $(\pi, i) \models <>f$ iff $(\pi, j) \models f$ for some $j \geq i$
- $(\pi, i) \models [\,]f$ iff $(\pi, j) \models f$ for all $j \geq i$
- $(\pi, i) \models f\,\mathcal{U}\,g$ iff $(\pi, k) \models g$ for some $k \geq i$ and $(\pi, j) \models f$ for all j, $i \leq j < k$
- $(\pi, i) \models f\,\mathcal{W}\,g$ iff $(\pi, i) \models [\,]f$ or $(\pi, i) \models f\,\mathcal{U}\,g$

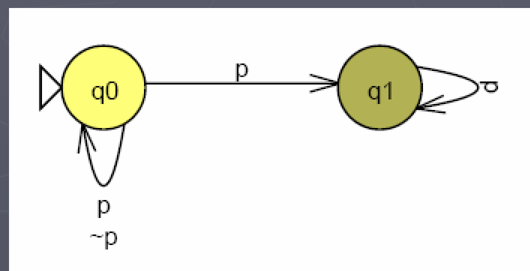
Semantics of Past Operators

- $(\pi, i) \models (-)f$ iff $i \geq 1$ and $(\pi, i-1) \models f$
- $(\pi, i) \models (\sim)f$ iff $i = 0$ or $(\pi, i-1) \models f$
- $(\pi, i) \models <->f$ iff $(\pi, j) \models f$ for some j, $0 \leq j \leq i$
- $(\pi, i) \models [-]f$ iff $(\pi, j) \models f$ for all j, $0 \leq j \leq i$
- $(\pi, i) \models f\,\mathcal{S}\,g$ iff $(\pi, k) \models g$ for some $k \leq i$, and $(\pi, j) \models f$ for all j, $k < j \leq i$
- $(\pi, i) \models f\,\mathcal{B}\,g$ iff $(\pi, i) \models [-]f$ or $(\pi, i) \models f\,\mathcal{S}\,g$

Example 1: <>[]p

- Meaning: p always holds after some time
- Satisfying models:
  - $(p)^{\omega}$, i.e., ppp…
  - $p{\sim}p{\sim}pp{\sim}p(p)^{\omega}$
- Unsatisfying models:
  - $p{\sim}p{\sim}pp({\sim}pp)^{\omega}$

<>[]p as a Büchi Automaton

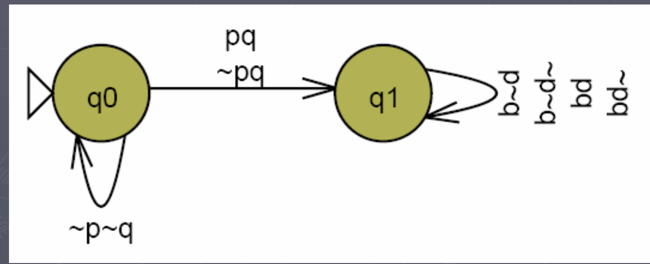


F = {q1}

Example 2: [](p --> <->q)

- Meaning: Every p is preceded by q.
- Satisfying models:
  - (~p~q)$^{\omega}$
  - (~p~q)(~pq) (~p~q) (p~q)$^{\omega}$
- Unsatisfying models:
  - (~p~q)(p~q)…

[](p --> <->q) as a Büchi Automaton

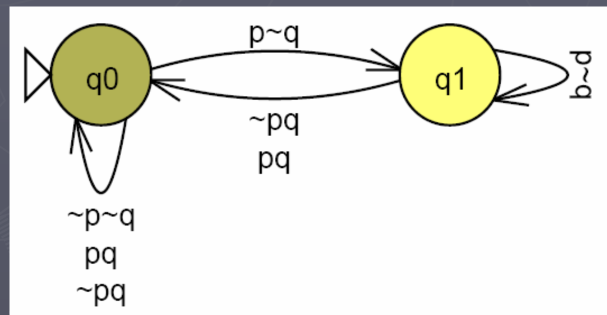


F = {q0,q1}

Example 3: [](p --> p *U* q)

- Meaning: Once p becomes true, it will remain true continuously until q becomes true, and q does become true.
- Satisfying models:
  - (~p~q)$^{\omega}$
  - (~p~q)(p~q)(p~q)(p~q)(~pq)(~p~q)$^{\omega}$
- Unsatisfying models:
  - (~p~q)(p~q)(~p~q)…

[](p --> p *U* q) as a Büchi Automaton

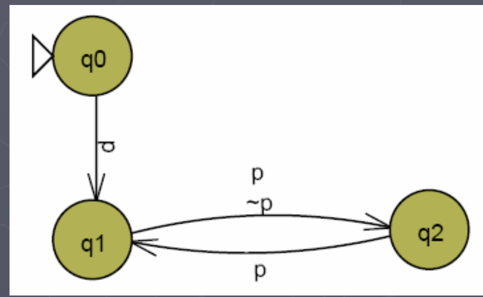


F = {q0}

Example 4: "Even p"

- This is NOT a PTL formula!
- Meaning: p holds in very even state.
  (Note: the states of a sequence are numbered 0,1,2,3,…)
- Satisfying models:
  - $(p)^\omega$
  - $(p{\sim}p)^\omega$
  - $p{\sim}pp{\sim}p(pp)^\omega$
- Unsatisfying models:
  - $p{\sim}ppp{\sim}p(pp)^\omega$

"Even p" as a Büchi Automaton



F = {q0,q1,q2}

Main Features of GOAL

- Drawing and Running Büchi Automata
- PTL Formulae to BA Translation
- Boolean Operations on BA
  - Union
  - Intersection
  - Complement
- Tests on BA
  - Emptiness
  - Containment (language containment)
  - Equivalence (language equivalence)
- Repositories of pre-drawn BA.

Test Running a BA

- To get an intuitive understanding of what language is being defined by the BA.
- Input format
  - Input string: $ppp{\sim}pp({\sim}pp)^\omega$
    Real format: (p)(p)(p)(~p)(p){(~p)(p)}
  - Input string: $({\sim}pq)\ (\ ({\sim}pq)\ ({\sim}p{\sim}q)\ ({\sim}p{\sim}q)\ )^\omega$
    Real format: (~pq) **{**(~pq) (~p~q) (~p~q)**}**

Demo Script

- Draw a BA, intended for <>[]p.
- Check if it is correct, by comparing with a machine-translated one.
- Try to specify “Even p” in PTL.
- See why it fails.
- Perhaps more …

The Future of GOAL

GOAL is constantly being improved; possible future extensions include:

- Integration with LTL model checkers
  - For example, export automata as Promela code for SPIN
- QPTL, PSL, S1S, etc. to Büchi automata (and vice versa)
- Minimization of Büchi automata
- Transformation to and from other variants of ω-automata
- Even better editing environment
  - Faster local updates in large graph layouts