IA159 Formal Verification Methods

LTL→BA via Alternating 1-Weak BA

Jan Strejček

Faculty of Informatics
Masaryk University

Focus and sources

Focus

- linear temporal logic (LTL) and Büchi automata (BA)
- alternating 1-weak Büchi automata (A1W)
- translation LTL→A1W
- translation A1W→BA

Source

- M. Y. Vardi: *An Automata-Theoretic Approach to Linear Temporal Logic*, LNCS 1043, Springer, 1995.

Syntax of LTL

Linear Temporal Logic (LTL) is defined by

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

where $\top$ stands for true and a ranges over a countable set AP of atomic propositions.

Abbreviations: $\bot \equiv \neg\top$ $\quad$ $\mathsf{F}\varphi \equiv \top \, \mathsf{U} \, \varphi$ $\quad$ $\mathsf{G}\varphi \equiv \neg\mathsf{F}\neg\varphi$

Terminology and intuitive meaning

$\mathsf{X}a$	next	$\bullet \; a \; \bullet \; \bullet \; \bullet \; \ldots$
$a \, \mathsf{U} \, b$	until	$a \; a \; \ldots \; a \; b \; \bullet \; \bullet \; \bullet \; \ldots$
$\mathsf{F}a$	eventually	$\bullet \; \bullet \; \ldots \; \bullet \; a \; \bullet \; \bullet \; \bullet \; \ldots$
$\mathsf{G}a$	always	$a \; a \; a \; a \; \ldots$

Semantics of LTL

Let $\Sigma = 2^{AP'}$, where $AP' \subseteq AP$ is a finite subset. We interpret LTL on infinite words $w = w(0)w(1)\ldots \in \Sigma^\omega$. By w_i we denote the suffix of w of the form $w(i)w(i+1)w(i+2)\ldots$.
The validity of an LTL formula φ for $w \in \Sigma^\omega$, written $w \models \varphi$, is defined as

$$
\begin{array}{lll}
w \models \top & & \\
w \models a & \text{iff} & a \in w(0) \\
w \models \neg\varphi & \text{iff} & w \not\models \varphi \\
w \models \varphi_1 \wedge \varphi_2 & \text{iff} & w \models \varphi_1 \wedge w \models \varphi_2 \\
w \models \mathrm{X}\varphi & \text{iff} & w_1 \models \varphi \\
w \models \varphi_1 \,\mathrm{U}\, \varphi_2 & \text{iff} & \exists i \in \mathbb{N}_0 : w_i \models \varphi_2 \wedge \forall\, 0 \leq j < i : w_j \models \varphi_1
\end{array}
$$

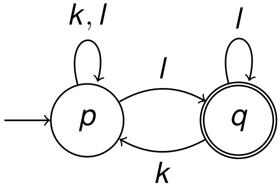
Given an alphabet Σ, an LTL formula φ defines the language

$$L^\Sigma(\varphi) = \{w \in \Sigma^\omega \mid w \models \varphi\}.$$

Büchi automata (BA)

A Büchi automaton (BA) is a tuple $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ defined precisely as a finite automaton. There are just two differences:

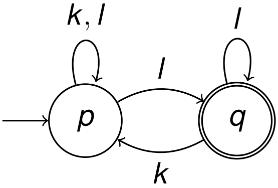
- a Büchi automaton is interpreted over infinite words
- a run is accepting if it visits some accepting state infinitely often



Büchi automata (BA)

A Büchi automaton (BA) is a tuple $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ defined precisely as a finite automaton. There are just two differences:

- a Büchi automaton is interpreted over infinite words
- a run is accepting if it visits some accepting state infinitely often



Accepts all infinite words over $\Sigma = \{k, l\}$ with infinitely many l.

LTL$\rightarrow$BA translations in general

- applications in automata-based LTL model checking, vacuity checking (checks trivial validity of a specification formula), ...
- many LTL$\rightarrow$BA translations
  - LTL $\rightarrow$ generalized Büchi automata (GBA) $\rightarrow$ BA (Spin)
  - LTL $\rightarrow$ transition-based GBA (TGBA) $\rightarrow$ BA (Spot)
  - LTL $\rightarrow$ alternating 1-weak Büchi automata (A1W) $\rightarrow$ BA
  - LTL $\rightarrow$ A1W $\rightarrow$ TGBA $\rightarrow$ BA (LTL2BA, LTL3BA)
  - ...
- translations via alternating 1-weak automata offer
  - size-reducing optimizations of alternating 1-weak BA
  - smaller resulting BA (in some cases)

Alternating Büchi automata

Positive boolean formulae

Positive boolean formulae over set Q ($\mathcal{B}^+(Q)$) are defined as

$$\varphi \ ::= \ \top \mid \bot \mid q \mid \varphi_1 \land \varphi_2 \mid \varphi_1 \lor \varphi_2$$

where $\top$ stands for true, $\bot$ stands for false, and q ranges over Q.

$S \subseteq Q$ is a model of φ $\iff$ the valuation assigning true just to elements of S satisfies φ

S is a minimal model of φ (written $S \models \varphi$) $\iff$ S is a model of φ and no proper subset of S is a model of φ

Examples of positive boolean formulae

formulae of $\mathcal{B}^+(\{p, q, r\})$	(minimal) models
$\bot$	no model
$\top$	$\emptyset, \{p\}, \{q\}, \{r\}, \{p, q\}, \ldots$
$p \wedge q$	$\{p, q\}, \{p, q, r\}$
$p \vee (q \wedge r)$	$\{p\}, \{p, q\}, \{p, r\}, \{q, r\}, \{p, q, r\}$
$p \wedge (q \vee r)$	$\{p, q\}, \{p, r\}, \{p, q, r\}$

Examples of positive boolean formulae

formulae of $\mathcal{B}^+(\{p, q, r\})$	(minimal) models
$\bot$	no model
$\top$	$\emptyset$, $\{p\}$, $\{q\}$, $\{r\}$, $\{p, q\}$, ...
$p \wedge q$	$\{p, q\}$, $\{p, q, r\}$
$p \vee (q \wedge r)$	$\{p\}$, $\{p, q\}$, $\{p, r\}$, $\{q, r\}$, $\{p, q, r\}$
$p \wedge (q \vee r)$	$\{p, q\}$, $\{p, r\}$, $\{p, q, r\}$

minimal models = clauses in disjunctive normal form

$$\varphi \equiv \bigvee_{S \models \varphi} \left(\bigwedge_{p \in S} p \right)$$

Alternating Büchi automata

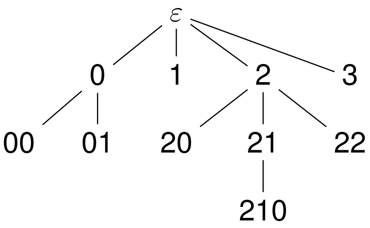
An alternating Büchi automaton is a tuple $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, where

- Σ is a finite alphabet,
- Q is a finite set of states,
- $\delta : Q \times \Sigma \rightarrow \mathcal{B}^+(Q)$ is a transition function,
- $q_0 \in Q$ is an initial state,
- $F \subseteq Q$ is a set of accepting states.

Trees

A tree is a set $T \subseteq \mathbb{N}_0^*$ such that if $xc \in T$, where $x \in \mathbb{N}_0^*$ and $c \in \mathbb{N}_0$, then also

- $x \in T$ and
- $xc' \in T$ for all $0 \leq c' < c$.

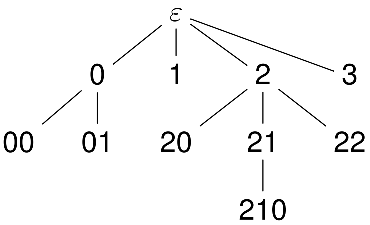


$$T = \{ \varepsilon, 0, 1, 2, 3, 00, 01, 20, 21, 22, 210 \}$$

Trees

A tree is a set $T \subseteq \mathbb{N}_0^*$ such that if $xc \in T$, where $x \in \mathbb{N}_0^*$ and $c \in \mathbb{N}_0$, then also

- $x \in T$ and
- $xc' \in T$ for all $0 \leq c' < c$.



$$T = \{\ \varepsilon, 0, 1, 2, 3, 00, 01, 20, 21, 22, 210\ \}$$

A Q-labeled tree is a pair (T, r) of a tree T and a labeling function $r : T \to Q$.

Alternating Büchi automata: a run

A run of an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ on word $w = w(0)w(1)\ldots \in \Sigma^\omega$ is a Q-labeled tree (T, r) such that

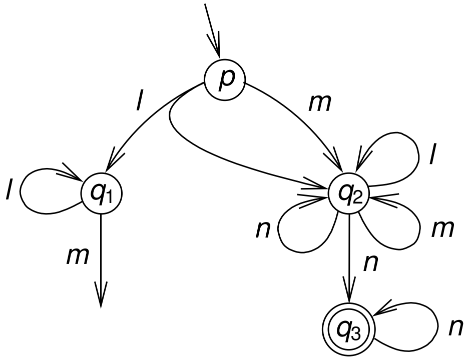
- $r(\varepsilon) = q_0$ and
- for each $x \in T$: $\{r(xc) \mid c \in \mathbb{N}_0, xc \in T\} \models \delta(r(x), w(|x|))$.

A run (T, r) is accepting iff for each infinite path π in T it holds that $Inf(\pi) \cap F \neq \emptyset$, where $Inf(\pi)$ is the set of all labels (i.e. states) appearing on π infinitely often.

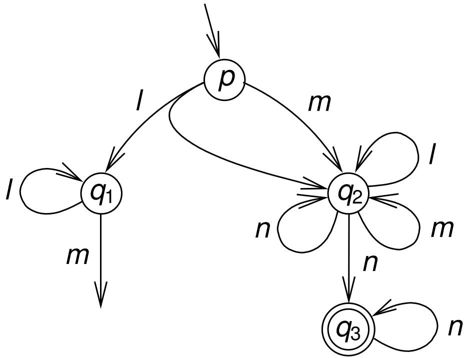
An automaton $\mathcal{A}$ accepts a word w iff there is an accepting run of $\mathcal{A}$ on w. We set

$$L(\mathcal{A}) = \{w \in \Sigma^\omega \mid \mathcal{A} \text{ accepts } w\}.$$

Example of an alternating Büchi automaton



Example of an alternating Büchi automaton



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

Alternating 1-weak Büchi automata (A1W)

Intuitively, an alternating BA is 1-weak (or linear or very weak, written A1W or VWAA) if it contains no cycles except selfloops.

Formally, let $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ be an alternating BA. For each $p \in Q$ we define the set of all successors of p as

$$Succ(p) = \{q \mid \exists l \in \Sigma, S \subseteq Q : S \cup \{q\} \models \delta(p, l)\}.$$

Automaton $\mathcal{A}$ is 1-weak (or linear or very weak) if there exists a partial order $\leq$ on Q such that for all $p, q \in Q$ it holds:

$$q \in Succ(p) \implies q \leq p$$

Notes

- standard Büchi automata are alternating Büchi automata where each $\delta(p, l)$ is $\perp$ or a disjunction of states
- A1W automata have the same expressive power as LTL

LTL→BA via alternating 1-weak BA

LTL→A1W

LTL→A1W

Input: an LTL formula φ and an alphabet $\Sigma = 2^{AP'}$
for some finite $AP' \subseteq AP$
Output: A1W automaton $\mathcal{A} = (\Sigma, Q, \delta, q_\varphi, F)$ accepting $L^\Sigma(\varphi)$

LTL→A1W

Input: an LTL formula φ and an alphabet $\Sigma = 2^{AP'}$ for some finite $AP' \subseteq AP$

Output: A1W automaton $\mathcal{A} = (\Sigma, Q, \delta, q_\varphi, F)$ accepting $L^\Sigma(\varphi)$

- $Q = \{q_\psi, q_{\neg\psi} \mid \psi \text{ is a subformula of } \varphi\}$

LTL→A1W

Input: an LTL formula φ and an alphabet $\Sigma = 2^{AP'}$
for some finite $AP' \subseteq AP$

Output: A1W automaton $\mathcal{A} = (\Sigma, Q, \delta, q_\varphi, F)$ accepting $L^\Sigma(\varphi)$

- $Q = \{q_\psi, q_{\neg\psi} \mid \psi \text{ is a subformula of } \varphi\}$
- δ is defined as follows (where $\overline{\alpha} \in \mathcal{B}^+(Q)$ satisfies $\overline{\alpha} \equiv \neg\alpha$)

$$
\begin{array}{lcl}
\delta(q_\top, I) & = & \top \\
\delta(q_a, I) & = & \top \text{ if } a \in I,\ \bot \text{ otherwise} \\
\delta(q_{\neg\psi}, I) & = & \overline{\delta(q_\psi, I)} \\
\delta(q_{\psi\wedge\rho}, I) & = & \delta(q_\psi, I) \wedge \delta(q_\rho, I) \\
\delta(q_{\mathsf{X}\psi}, I) & = & q_\psi \\
\delta(q_{\psi\mathsf{U}\rho}, I) & = & \delta(q_\rho, I) \vee (\delta(q_\psi, I) \wedge q_{\psi\mathsf{U}\rho})
\end{array}
\qquad
\begin{array}{lcl}
\overline{\top} & = & \bot \\
\overline{\bot} & = & \top \\
\overline{q_{\neg\psi}} & = & q_\psi \\
\overline{q_\psi} & = & q_{\neg\psi} \\
\overline{\beta \wedge \gamma} & = & \overline{\beta} \vee \overline{\gamma} \\
\overline{\beta \vee \gamma} & = & \overline{\beta} \wedge \overline{\gamma}
\end{array}
$$

LTL→A1W

Input: an LTL formula φ and an alphabet $\Sigma = 2^{AP'}$ for some finite $AP' \subseteq AP$

Output: A1W automaton $\mathcal{A} = (\Sigma, Q, \delta, q_\varphi, F)$ accepting $L^\Sigma(\varphi)$

- $Q = \{q_\psi, q_{\neg\psi} \mid \psi \text{ is a subformula of } \varphi\}$
- δ is defined as follows (where $\overline{\alpha} \in \mathcal{B}^+(Q)$ satisfies $\overline{\alpha} \equiv \neg\alpha$)

$$
\begin{array}{lcl}
\delta(q_\top, I) &=& \top \\
\delta(q_a, I) &=& \top \text{ if } a \in I,\ \bot \text{ otherwise} \\
\delta(q_{\neg\psi}, I) &=& \overline{\delta(q_\psi, I)} \\
\delta(q_{\psi\wedge\rho}, I) &=& \delta(q_\psi, I) \wedge \delta(q_\rho, I) \\
\delta(q_{\mathsf{X}\psi}, I) &=& q_\psi \\
\delta(q_{\psi\mathsf{U}\rho}, I) &=& \delta(q_\rho, I) \vee (\delta(q_\psi, I) \wedge q_{\psi\mathsf{U}\rho})
\end{array}
\qquad
\begin{array}{lcl}
\overline{\top} &=& \bot \\
\overline{\bot} &=& \top \\
\overline{q_{\neg\psi}} &=& q_\psi \\
\overline{q_\psi} &=& q_{\neg\psi} \\
\overline{\beta \wedge \gamma} &=& \overline{\beta} \vee \overline{\gamma} \\
\overline{\beta \vee \gamma} &=& \overline{\beta} \wedge \overline{\gamma}
\end{array}
$$

- $F = \{q_{\neg(\psi\mathsf{U}\rho)} \mid \psi\,\mathsf{U}\,\rho \text{ is a subformula of } \varphi\}$

LTL→A1W

Note that every infinite path of a run of $\mathcal{A}$ has a suffix labeled with a state of the form $q_{\psi \mathsf{U} \rho}$ or $q_{\neg(\psi \mathsf{U} \rho)}$ (other states have no loops and can appear at most once on a path). F is defined to prevent the first case: $\psi \mathsf{U} \rho$ is satisfied only if ρ eventually holds.

Theorem

Given an LTL formula φ and an alphabet Σ, one can construct an A1W automaton $\mathcal{A}$ accepting $L^{\Sigma}(\varphi)$ and such that the number of states of $\mathcal{A}$ is linear in the length of φ.

LTL$\rightarrow$BA via alternating 1-weak BA

A1W$\rightarrow$BA

A1W→BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q_0', F')$ accepting $L(\mathcal{A})$

A1W→BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q_0', F')$ accepting $L(\mathcal{A})$

Intuitively, $\mathcal{A}'$ tracks states on each level of the computation tree of $\mathcal{A}$. Moreover, $\mathcal{A}'$ has to divide the set of states into two sets: states labeling paths with recent occurrence of an accepting state, and states labeling the other paths.

A1W→BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q'_0, F')$ accepting $L(\mathcal{A})$

- $Q' = 2^Q \times 2^Q$

A1W$\to$BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q_0', F')$ accepting $L(\mathcal{A})$

- $Q' = 2^Q \times 2^Q$
- $q_0' = (\{q_0\}, \emptyset)$

A1W→BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q_0', F')$ accepting $L(\mathcal{A})$

- $Q' = 2^Q \times 2^Q$
- $q_0' = (\{q_0\}, \emptyset)$
- $\delta'((U, V), l)$ is defined as:
  - if $U \neq \emptyset$ then

$$\begin{aligned}\delta'((U,V),l) = \{(U',V') \mid {} & \exists X, Y \subseteq Q \text{ such that} \\ & X \models \textstyle\bigwedge_{q\in U} \delta(q,l) \text{ and} \\ & Y \models \textstyle\bigwedge_{q\in V} \delta(q,l) \text{ and} \\ & U' = X \smallsetminus F \text{ and } V' = Y \cup (X \cap F)\}\end{aligned}$$

  - if $U = \emptyset$ then

$$\begin{aligned}\delta'((\emptyset,V),l) = \{(U',V') \mid {} & \exists Y \subseteq Q \text{ such that} \\ & Y \models \textstyle\bigwedge_{q\in V} \delta(q,l) \text{ and} \\ & U' = Y \smallsetminus F \text{ and } V' = Y \cap F)\}\end{aligned}$$

A1W→BA

Input: an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$
Output: a BA $\mathcal{A}' = (\Sigma, Q', \delta', q_0', F')$ accepting $L(\mathcal{A})$

- $Q' = 2^Q \times 2^Q$
- $q_0' = (\{q_0\}, \emptyset)$
- $\delta'((U, V), l)$ is defined as:
  - if $U \neq \emptyset$ then

$$\begin{aligned}\delta'((U, V), l) = \{(U', V') \mid\ & \exists X, Y \subseteq Q \text{ such that} \\ & X \models \textstyle\bigwedge_{q \in U} \delta(q, l) \text{ and} \\ & Y \models \textstyle\bigwedge_{q \in V} \delta(q, l) \text{ and} \\ & U' = X \setminus F \text{ and } V' = Y \cup (X \cap F)\}\end{aligned}$$

  - if $U = \emptyset$ then

$$\begin{aligned}\delta'((\emptyset, V), l) = \{(U', V') \mid\ & \exists Y \subseteq Q \text{ such that} \\ & Y \models \textstyle\bigwedge_{q \in V} \delta(q, l) \text{ and} \\ & U' = Y \setminus F \text{ and } V' = Y \cap F)\}\end{aligned}$$

- $F' = \{\emptyset\} \times 2^Q$

LTL→A1W

Theorem

Given an alternating BA $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$, one can construct a BA $\mathcal{A}'$ accepting $L(\mathcal{A})$ and such that the number of states of $\mathcal{A}'$ is $2^{\mathcal{O}(|Q|)}$.

Corollary

Given an LTL formula φ and an alphabet Σ, one can construct a BA $\mathcal{A}'$ accepting $L^{\Sigma}(\varphi)$ and such that the number of states of $\mathcal{A}'$ is $2^{\mathcal{O}(|\varphi|)}$.

Coming next week

Partial order reduction

- When can a state/transition be safely removed from a Kripke structure?
- What is a stuttering principle?
- Can we effectively compute the reduction?