Automata theory and its applications

Lecture 17 -18: Automata-theoretical approach to model checking

Zhilin Wu

State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences

April 17, 2013

Outline

1. Linear temporal logic (LTL)
2. LTL model checking: Automata theoretical approach
3. Computation tree logic (CTL)
4. (Weak) alternating tree automata
5. CTL model checking: Automata theoretical approach

Temporal logics: The general background

A brief history

- Introduced by a philosopher Arthur Prior in 1950's (known as tense logic).
- Introduced to computer science (Linear temporal logic) by Amir Pnueli in 1977.

Temporal logics: The general background

A brief history

- Introduced by a philosopher Arthur Prior in 1950's (known as tense logic).
- Introduced to computer science (Linear temporal logic) by Amir Pnueli in 1977.

Classifications

Linear time versus branching time

- Linear time: Each moment has a unique future.
- Branching time: Each moment may have several possible futures.

Time point versus intervals

- Refer to the time by time points: Linear temporal logic, Computation tree logic, Modal μ-calculus,
- Refer to the time by time intervals: Interval temporal logics.

Temporal logics: The general background

A brief history

- Introduced by a philosopher Arthur Prior in 1950's (known as tense logic).
- Introduced to computer science (Linear temporal logic) by Amir Pnueli in 1977.

Classifications

Linear time versus branching time

- Linear time: Each moment has a unique future.
- Branching time: Each moment may have several possible futures.

Time point versus intervals

- Refer to the time by time points: Linear temporal logic, Computation tree logic, Modal μ-calculus,
- Refer to the time by time intervals: Interval temporal logics.

Extensions

Timed, probabilistic, …

Linear temporal logic (LTL)

Syntax of LTL:

$$\varphi := p(p \in AP) \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi_1 \mid X\varphi_1 \mid \varphi_1 U \varphi_2.$$

Semantics of LTL:

Let $w \in (2^{AP})^\omega$ and φ be a LTL formula. Then

- $(w,i) \vDash p$ iff $p \in w_0$,
- $(w,i) \vDash \varphi_1 \vee \varphi_2$ iff $(w,i) \vDash \varphi_1$ or $(w,i) \vDash \varphi_2$,
- $(w,i) \vDash \neg\varphi_1$ iff not $(w,i) \vDash \varphi_1$,
- $(w,i) \vDash X\varphi_1$ iff $(w,i+1) \vDash \varphi_1$,
- $(w,i) \vDash \varphi_1 U \varphi_2$ iff $\exists j$ s.t. $j \geq i$, $(w,j) \vDash \varphi_2$ and $\forall k : i \leq k < j$, $(w,k) \vDash \varphi_1$.

$w \vDash \varphi$ iff $(w,0) \vDash \varphi$.

$L(\varphi)$: $\{w \in (2^{AP})^\omega \mid w \vDash \varphi\}$.

Derived temporal operators:

$\top := p \vee \neg p$, $F\varphi := \top\, U\varphi$, $G\varphi := \neg F \neg\varphi$, $\varphi_1 R \varphi_2 := \neg(\neg\varphi_1 U \neg\varphi_2)$, ...

Remark: X: neXt, U: Until, F: Future, G: Global, R: Release.

Expressiveness of LTL

Examples: Xp, pUq, $G(p \to Fq)$, FGp, $GFp \to GFq$.

Proposition. The property "event p occurs at all even time points" is not expressible in LTL.

How about the formula $p \wedge G(p \to Xq) \wedge G(q \to Xp)$*?*

Expressiveness of LTL

Examples: Xp, pUq, $G(p \to Fq)$, FGp, $GFp \to GFq$.

Proposition. The property "event p occurs at all even time points" is not expressible in LTL.

> *How about the formula* $p \wedge G(p \to Xq) \wedge G(q \to Xp)$*?*

Lemma. Let $AP = \{p\}$. Then for every LTL formula φ of size n over AP and every $m, m' \geq n$, $\{p\}^m(\varnothing\{p\})^\omega \vDash \varphi$ iff $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash \varphi$.

Proof (Proposition).

For contradiction, suppose that "event p occurs at even time points" can be defined by a LTL formula φ.
Let $n = |\varphi|$.
From the lemma, $\{p\}^n(\varnothing\{p\})^\omega \vDash \varphi$ iff $\{p\}^{n+1}(\varnothing\{p\})^\omega \vDash \varphi$.
On the other hand, either not $\{p\}^n(\varnothing\{p\})^\omega \vDash \varphi$ or not $\{p\}^{n+1}(\varnothing\{p\})^\omega \vDash \varphi$.
We get a contradiction. □

Theorem. $LTL \equiv FO[AP, +1, <]$.

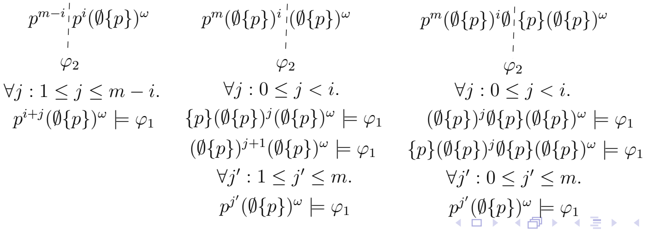
Expressiveness of LTL

Proof of the lemma.

Induction on the structure of φ.

- $\varphi = p$ and $m, m' \geq n = 1$: $\{p\}^m(\varnothing\{p\})^\omega \vDash p$ iff $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash p$,
- $\varphi = \varphi_1 \vee \varphi_2$ or $\varphi = \neg\varphi_1$: easy,
- $\varphi = X\varphi_1$: $\{p\}^m(\varnothing\{p\})^\omega \vDash X\varphi_1$ iff $\{p\}^{m-1}(\varnothing\{p\})^\omega \vDash \varphi_1$ iff $\{p\}^{m'-1}(\varnothing\{p\})^\omega \vDash \varphi_1$ iff $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash X\varphi_1$,
- $\varphi = \varphi_1 U \varphi_2$: By symmetry, it is sufficient to show $\{p\}^m(\varnothing\{p\})^\omega \vDash \varphi_1 U \varphi_2$ $\Rightarrow$ $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash \varphi_1 U \varphi_2$. There are three situations.

□



$p^{m-i}\,p^i(\emptyset\{p\})^\omega$

φ_2

$\forall j : 1 \leq j \leq m - i.$

$p^{i+j}(\emptyset\{p\})^\omega \models \varphi_1$

$p^m(\emptyset\{p\})^i\,(\emptyset\{p\})^\omega$

φ_2

$\forall j : 0 \leq j < i.$

$\{p\}(\emptyset\{p\})^j(\emptyset\{p\})^\omega \models \varphi_1$

$(\emptyset\{p\})^{j+1}(\emptyset\{p\})^\omega \models \varphi_1$

$\forall j' : 1 \leq j' \leq m.$

$p^{j'}(\emptyset\{p\})^\omega \models \varphi_1$

$p^m(\emptyset\{p\})^i\emptyset\,\{p\}(\emptyset\{p\})^\omega$

φ_2

$\forall j : 0 \leq j < i.$

$(\emptyset\{p\})^j\emptyset\{p\}(\emptyset\{p\})^\omega \models \varphi_1$

$\{p\}(\emptyset\{p\})^j\emptyset\{p\}(\emptyset\{p\})^\omega \models \varphi_1$

$\forall j' : 0 \leq j' \leq m.$

$p^{j'}(\emptyset\{p\})^\omega \models \varphi_1$

Expressiveness of LTL

Proof of the lemma.

Induction on the structure of φ.

- $\varphi = p$ and $m, m' \geq n = 1$: $\{p\}^m(\varnothing\{p\})^\omega \vDash p$ iff $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash p$,
- $\varphi = \varphi_1 \vee \varphi_2$ or $\varphi = \neg\varphi_1$: easy,
- $\varphi = X\varphi_1$: $\{p\}^m(\varnothing\{p\})^\omega \vDash X\varphi_1$ iff $\{p\}^{m-1}(\varnothing\{p\})^\omega \vDash \varphi_1$ iff $\{p\}^{m'-1}(\varnothing\{p\})^\omega \vDash \varphi_1$ iff $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash X\varphi_1$,
- $\varphi = \varphi_1 U \varphi_2$: By symmetry, it is sufficient to show $\{p\}^m(\varnothing\{p\})^\omega \vDash \varphi_1 U \varphi_2$ $\Rightarrow$ $\{p\}^{m'}(\varnothing\{p\})^\omega \vDash \varphi_1 U \varphi_2$. There are three situations.

□

To exemplify the proof, consider the second situation:

$$(\varnothing\{p\})^\omega \vDash \varphi_1 \text{ and } \forall j' : 1 \leq j' \leq m. \{p\}^{j'}(\varnothing\{p\})^\omega \vDash \varphi_1.$$

Then

$$\{p\}^m(\varnothing\{p\})^\omega \vDash \varphi_1 \Rightarrow \forall n \leq j' \leq m'. \{p\}^{j'}(\varnothing\{p\})^\omega \vDash \varphi_1 \ \textit{(By IH)} \Rightarrow$$
$$\forall 1 \leq j' \leq m'. \{p\}^{j'}(\varnothing\{p\})^\omega \vDash \varphi_1 \Rightarrow \{p\}^{m'}(\varnothing\{p\})^\omega \vDash \varphi_1 U \varphi_2.$$

The arguments for the other two situations are similar.

Outline

1. Linear temporal logic (LTL)
2. LTL model checking: Automata theoretical approach
3. Computation tree logic (CTL)
4. (Weak) alternating tree automata
5. CTL model checking: Automata theoretical approach

Kripke structure

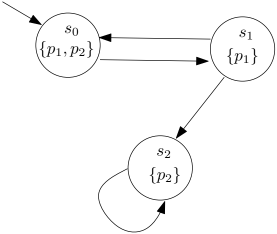
A Kripke structure $\mathcal{S}$ is a tuple $(S, AP, \rightarrow, I, L)$, where

- S: the set of states,
- AP: the set of atomic propositions,
- $\rightarrow \subseteq S \times S$: the transition relation s.t. $\forall s \exists s'. s \rightarrow s'$,
- $I \subseteq S$: The set of initial states,
- $L : S \rightarrow 2^{AP}$: The labelling function.

A *path* π in $\mathcal{S}$: An infinite sequence of states $s_0 s_1 \dots$ s.t. $\forall i. s_i \rightarrow s_{i+1}$.

A path $s_0 s_1 \dots$ is *initial* if $s_0 \in I$.

$L(\mathcal{S}) = \{L(\pi) \mid \pi$ is an initial path in $\mathcal{S}\}$, where $L(\pi) = L(s_0)L(s_1)\dots$ if $\pi = s_0 s_1 \dots$.



LTL model checking

Let $\mathcal{S} = (S, AP, \rightarrow, I, L)$ be a Kripke structure and φ be an LTL formula. Then $\mathcal{S} \models \varphi$ iff for every initial path π in $\mathcal{S}$, $L(\pi) \models \varphi$.

Model checking (MC) problem:

> *Given a Kripke structure $\mathcal{S}$ and an LTL formula φ, decide whether $\mathcal{S} \models \varphi$.*

Automata-theoretical approach to MC problem

The idea:

> $\mathcal{S} = (S, AP, \rightarrow, I, L)$ *can be viewed as a Büchi automaton* $\mathcal{A}_{\mathcal{S}} = (S, 2^{AP}, \delta, I, S)$, *where* $(s, P, s') \in \delta$ *iff* $s \rightarrow s'$ *and* $P = L(s)$.

The algorithm:

1. Construct an equivalent Büchi automaton $\mathcal{A}_{\neg\varphi}$ from $\neg\varphi$.
2. Construct $\mathcal{A}'$ as a product of $\mathcal{A}_{\mathcal{S}}$ and $\mathcal{A}_{\neg\varphi}$ accepting $L(\mathcal{A}_{\mathcal{S}}) \cap L(\mathcal{A}_{\neg\varphi})$.
3. Decide whether $L(\mathcal{A}')$ is empty.

Question: How to construct $\mathcal{A}_{\neg\varphi}$ from $\neg\varphi$?

Generalised Büchi automata (GBA)

A GBA $\mathcal{A}$ is a tuple $(Q, 2^{AP}, \delta, I, \mathcal{F})$, where

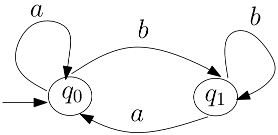
- Q: the set of states,
- δ: the set of states,
- I: the set of initial states,
- $\mathcal{F} \subseteq 2^Q$: the acceptance component.

The runs of a GBA over ω-words are defined similarly to those of BA.

A run $r = q_0 q_1 \dots$ of a GBA $\mathcal{A}$ is accepting if $\forall F \in \mathcal{F}$, $\text{Inf}(r) \cap F \neq \emptyset$.

Example:

$$\mathcal{F} = \{\{q_0\}, \{q_1\}\}$$



GBA ≡ BA

Proposition. Given a GBA $\mathcal{A}$, an equivalent BA $\mathcal{A}'$ can be constructed in quadratic time.

Proof.

Let $\mathcal{A} = (Q, 2^{AP}, \delta, I, \mathcal{F})$ be a GBA.
Suppose $\mathcal{F} = \{F_1, \ldots, F_k\}$, we construct a BA $\mathcal{A}' = (Q', 2^{AP}, \delta', I', F')$ as follows.

- $Q' = Q \times \{0, \ldots, k\}$,
- $I' = I \times \{0\}$,
- $F' = Q \times \{k\}$,
- δ' is defined by the following rules,
    - for every $(q, P, q') \in \delta$ and every $i : 1 \leq i \leq k$ s.t. $q' \in F_i$, $((q, i-1), P, (q', i)) \in \delta'$,
    - for every $(q, P, q') \in \delta$, $((q, k), P, (q', 0)) \in \delta'$.

☐

Closure of LTL formulas

For an LTL formula φ, let $\mathrm{sub}(\varphi)$ denote the set of subformulas of φ.

Given an LTL formula φ, the *closure* of φ, denoted by $\mathrm{cl}(\varphi)$, is

$$\mathrm{sub}(\varphi) \cup \{\neg\psi \mid \psi \in \mathrm{sub}(\varphi)\} \text{ (where } \neg\neg\psi \text{ and } \psi \text{ are identified).}$$

Example:

Suppose $\varphi = G(p \to Fq) = \neg\,(true\ U \neg(\neg p \vee Fq))$. Then

$$\mathrm{cl}(\varphi) = \left\{ \begin{array}{c} p, \neg p, q, \neg q, true, \neg true, \\ Fq, \neg Fq, \\ \neg p \vee Fq, \neg(\neg p \vee Fq), \\ true\ U \neg(\neg p \vee Fq), \varphi \end{array} \right\},$$

where $true = p \vee \neg p, Fq = true\ U q$.

Elementary sets of formulas

Let φ be an LTL formula and $B \subseteq \mathrm{cl}(\varphi)$.

Then B is said to be *elementary* if B satisfies the following conditions,

- Consistency wrt. Boolean operators: For every $\psi_1 \vee \psi_2, \psi \in \mathrm{cl}(\varphi)$,
  - $\psi_1 \vee \psi_2 \in B$ iff $\psi_1 \in B$ or $\psi_2 \in B$,
  - if $\psi \in B$, then $\neg\psi \notin B$,
- Local consistency wrt. Until operators: For every $\psi_1 U \psi_2 \in \mathrm{cl}(\varphi)$,
  - if $\psi_2 \in B$, then $\psi_1 U \psi_2 \in B$,
  - if $\psi_1 U \psi_2 \in B$ and $\psi_2 \notin B$, then $\psi_1 \in B$,
- Maximality: For every $\psi \in \mathrm{cl}(\varphi)$, if $\psi \notin B$, then $\neg\psi \in B$.

Example:

Let $\varphi = G(p \to Fq) = \neg(\mathit{true}\ U \neg(\neg p \vee Fq))$.

Suppose $B = \{\neg p, q, \mathit{true}, Fq, \neg p \vee Fq, \mathit{true}\ U \neg(\neg p \vee Fq)\}$.

Then B is elementary.

- Boolean cosistency: $\neg p \in B \Rightarrow \mathit{true}, \neg p \vee Fq \in B, \dots,$
- Local consistency wrt. Until: $q \in B \Rightarrow Fq \in B$,
  $\mathit{true}\ U \neg(\neg p \vee Fq) \in B$, $\neg(\neg p \vee Fq) \notin B \Rightarrow \mathit{true} \in B$,
- Maximality: $\varphi \notin B \Rightarrow \mathit{true}\ U \neg(\neg p \vee Fq) \in B, \dots.$

From LTL to GBA

Theorem. Given an LTL formula φ, an equivalent GBA $\mathcal{A} = (Q, 2^{AP}, \delta, I, \mathcal{F})$ s.t. $|Q| = 2^{O(|\varphi|)}$ and $|\mathcal{F}| = O(|\varphi|)$ can be constructed.

Proof.

Let φ be an LTL formula.
Define a GBA $\mathcal{A} = (Q, 2^{AP}, \delta, I, \mathcal{F})$ as follows.

- Q is the set of elementary set of formulas $B \subseteq \mathrm{cl}(\varphi)$,
- $I = \{B \mid \varphi \in B\}$,
- δ is the set of tuples (B, P, B') s.t.
  - $P = \{p \in AP \mid p \in B\}$,
  - for every $\psi, X\psi \in \mathrm{cl}(\varphi)$, $X\psi \in B$ iff $\psi \in B'$,
  - for every $\psi_1 U \psi_2 \in \mathrm{cl}(\varphi)$,

$$\psi_1 U \psi_2 \in B \Leftrightarrow (\psi_2 \in B \text{ or } (\psi_1 \in B, \psi_1 U \psi_2 \in B')).$$

- $\mathcal{F} = \{F_{\psi_1 U \psi_2} \mid \psi_1 U \psi_2 \in \mathrm{cl}(\varphi)\}$, where

$$F_{\psi_1 U \psi_2} = \{B \in Q \mid \psi_1 U \psi_2 \in B \Rightarrow \psi_2 \in B\}.$$

Claim. For every $w \in (2^{AP})^\omega$, $w \vDash \varphi$ iff $w \in L(\mathcal{A})$. ☐

From LTL to GBA

Claim. For every $w \in (2^{AP})^\omega$, $w \models \varphi$ iff $w \in L(\mathcal{A})$.

Proof.

"Only if" direction: Suppose $w \models \varphi$.
For every $i \in \mathbb{N}$, let $B_i = \{\psi \in \mathrm{cl}(\varphi) \mid (w, i) \models \psi\}$.
Then $B_0 B_1 \dots$ is a run of $\mathcal{A}$ over w.
$B_0 B_1 \dots$ is also an accepting run:

For every $\psi_1 U \psi_2 \in \mathrm{cl}(\varphi)$,

- if $\exists i. \forall j : j \geq i.\ \psi_1 U \psi_2 \notin B_j$, then
$$\forall j : j \geq i.\ B_j \in F_{\psi_1 U \psi_2} \Rightarrow \mathrm{Inf}(B_0 B_1 \dots) \cap F_{\psi_1 U \psi_2} \neq \varnothing,$$
- if $\exists$ infinitely many i s.t. $\psi_1 U \psi_2 \in B_i$, in other words, $(w, i) \models \psi_1 U \psi_2$, then
  $\exists$ *infinitely many* i' *s.t.* $(w, i') \models \psi_2$,
  thus, $\psi_2, \psi_1 U \psi_2 \in B_{i'}$, *so,* $B_{i'} \in F_{\psi_1 U \psi_2}$
  $\Rightarrow$
  $\mathrm{Inf}(B_0 B_1 \dots) \cap F_{\psi_1 U \psi_2} \neq \varnothing$.

□

From LTL to GBA

Claim. For every $w \in (2^{AP})^\omega$, $w \models \varphi$ iff $w \in L(\mathcal{A})$.

Proof.

"If" direction: Suppose $w \in L(\mathcal{A})$.
Then there is an accepting run $B_0 B_1 \ldots$ of $\mathcal{A}$ over w.
It is sufficient to show that for every $\psi \in \mathrm{cl}(\varphi)$, the following holds,

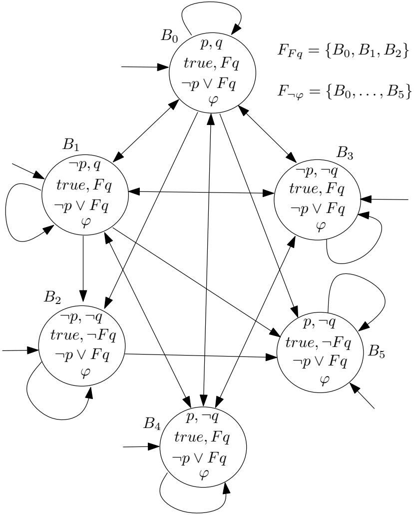
$$\text{for every } i \in \mathbb{N} \text{ s.t. } \psi \in B_i,\ (w,i) \models \psi.$$

Induction on the structure of formulas.

- $\psi = p$: Then $p \in B_i$, so $p \in w_i$ (from the construction of $\mathcal{A}$), $(w,i) \models \psi$,
- $\psi = \psi_1 \vee \psi_2$ or $\psi = \neg\psi_1$: Easy.
- $\psi = X\psi_1$: Then $\psi_1 \in B_{i+1}$, so $(w,i+1) \models \psi_1$ (by induction hypothesis), $(w,i) \models X\psi_1$.
- $\psi = \psi_1 U \psi_2$: Then either $\psi_2 \in B_i$ or ($\psi_1 \in B_i$ and $\psi_1 U \psi_2 \in B_{i+1}$).
  From $\mathrm{Inf}(B_0 B_1 \ldots) \cap F_{\psi_1 U \psi_2} \neq \emptyset$, we know
  $$\exists j : j \geq i.\ \psi_2 \in B_j \text{ and } \forall k : i \leq k < j.\ \psi_1 \in B_k.$$
  By induction hypothesis, $(w,j) \models \psi_2$ and $\forall k : i \leq k < j.\ (w,k) \models \psi_1$.
  We deduce that $(w,i) \models \psi_1 U \psi_2$.

From LTL to GBA: An example

Let $\varphi = G(p \to Fq) = \neg(true\ U \neg(\neg p \vee Fq))$.



$F_{Fq} = \{B_0, B_1, B_2\}$

$F_{\neg\varphi} = \{B_0, \dots, B_5\}$

Outline

1. Linear temporal logic (LTL)
2. LTL model checking: Automata theoretical approach
3. Computation tree logic (CTL)
4. (Weak) alternating tree automata
5. CTL model checking: Automata theoretical approach

CTL

Syntax:

$\varphi := p (p \in AP) \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi_1 \mid EX\varphi_1 \mid AX\varphi_1 \mid E\varphi_1 U \varphi_2 \mid A\varphi_1 U \varphi_2$

Semantics:

Given a Kripke structure $\mathcal{S} = (S, AP, \rightarrow, I, L)$ and a CTL formula φ,

- $(\mathcal{S}, s) \vDash p$ iff $p \in L(s)$,
- $(\mathcal{S}, s) \vDash \varphi_1 \vee \varphi_2$ iff $(\mathcal{S}, s) \vDash \varphi_1$ or $(\mathcal{S}, s) \vDash \varphi_2$,
- $(\mathcal{S}, s) \vDash \neg\varphi_1$ iff not $(\mathcal{S}, s) \vDash \varphi_1$,
- $(\mathcal{S}, s) \vDash EX\varphi_1$ iff there exists s' s.t. $s \rightarrow s'$ and $(\mathcal{S}, s') \vDash \varphi_1$,
- $(\mathcal{S}, s) \vDash AX\varphi_1$ iff for all s' s.t. $s \rightarrow s'$, it holds $(\mathcal{S}, s') \vDash \varphi_1$,
- $(\mathcal{S}, s) \vDash E\varphi_1 U \varphi_2$ iff there exists a path π of $\mathcal{S}$ starting from s s.t. $\pi \vDash \varphi_1 U \varphi_2$,
- $(\mathcal{S}, s) \vDash A\varphi_1 U \varphi_2$ iff for every path π of $\mathcal{S}$ starting from s, $\pi \vDash \varphi_1 U \varphi_2$,

where $\pi \vDash \varphi_1 U \varphi_2$ iff $\exists i \geq 0$, $(\mathcal{S}, \pi(i)) \vDash \varphi_2$ and $\forall j : 0 \leq j < i$, $(\mathcal{S}, \pi(j)) \vDash \varphi_1$.

$\mathcal{S} \vDash \varphi$ iff for every $s_0 \in I$, $(\mathcal{S}, s_0) \vDash \varphi$.

Example: AFq, $AG(p \rightarrow AFq)$.

Positive normal form (PNF) of CTL

Recall: R (Release) operator, $\varphi_1 R \varphi_2 = \neg(\neg\varphi_1 U \neg\varphi_2)$.
Let $w \in (2^{AP})^\omega$ and $\varphi_1 R \varphi_2$ be a LTL formula, then $(w, i) \vDash \varphi_1 R \varphi_2$ iff

- either for every $j : i \leq j$, $(w, j) \vDash \varphi_2$,
- or there exists $j : i \leq j$ s.t. $(w, j) \vDash \varphi_1$ and for every $k : i \leq k \leq j$, $(w, k) \vDash \varphi_2$.

Fact. $\neg(\varphi_1 U \varphi_2) \equiv (\neg\varphi_1) R (\neg\varphi_2)$ and $\neg(\varphi_1 R \varphi_2) \equiv (\neg\varphi_1) U (\neg\varphi_2)$.

Positive normal form for CTL:

$$\varphi := \begin{array}{c} true \mid false \mid p \mid \neg p \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid EX\varphi_1 \mid AX\varphi_1 \mid \\ E\varphi_1 U \varphi_2 \mid A\varphi_1 U \varphi_2 \mid E\varphi_1 R \varphi_2 \mid A\varphi_1 R \varphi_2 \end{array}$$

Proposition. Every CTL formula can be transformed into an equivalent formula in positive normal form.

Proof.

The idea: Push $\neg$ to the front of atomic positions.
For instance, $\neg(E\varphi_1 U \varphi_2) \equiv A(\neg\varphi_1) R (\neg\varphi_2)$, $\neg(E\varphi_1 R \varphi_2) \equiv A(\neg\varphi_1) U (\neg\varphi_2)$. ☐

Outline

1. Linear temporal logic (LTL)
2. LTL model checking: Automata theoretical approach
3. Computation tree logic (CTL)
4. **(Weak) alternating tree automata**
5. CTL model checking: Automata theoretical approach

Alternating automata over binary trees

A notation:

Let X be a finite set. Then $\mathcal{B}^+(X)$ is the positive Boolean combinations of elements of X, formally,

$$\varphi := true \mid false \mid x (x \in X) \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$$

An *alternating Büchi automaton* over infinite binary trees (ABTA) $\mathcal{A}$ is a tuple $(Q, 2^{AP}, \delta, q_0, F)$, where

- Q, q_0, F are similar to those of nondeterministic Büchi automata,
- $\delta \subseteq Q \times 2^{AP} \rightarrow \mathcal{B}^+(\{0,1\} \times Q)$.

Alternating automata over binary trees

A notation:

Let X be a finite set. Then $\mathcal{B}^+(X)$ is the positive Boolean combinations of elements of X, formally,

$$\varphi := true \mid false \mid x (x \in X) \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2$$

An *alternating Büchi automaton* over infinite binary trees (ABTA) $\mathcal{A}$ is a tuple $(Q, 2^{AP}, \delta, q_0, F)$, where

- Q, q_0, F are similar to those of nondeterministic Büchi automata,
- $\delta \subseteq Q \times 2^{AP} \to \mathcal{B}^+(\{0,1\} \times Q)$.

A *run* of a ABTA $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ over a binary tree $t = (D, L)$ is an infinite tree $r_{\mathcal{A},t} = (D_r, L_r)$, where $D_r \subseteq \mathbb{N}^*$ is a tree domain and $L_r : D_r \to D \times Q$ satisfying the following conditions.

$\forall y \in D_r$ s.t. $L_r(y) = (x, q)$ and $\delta(q, L(x)) = \theta$.

Then there is $S = \{(b_0, q_0), \ldots, (b_n, q_n)\} \subseteq \{0,1\} \times Q$ s.t.

$S \vDash \theta$, *and* $\forall i : 0 \leq i \leq n$, $yi \in D_r$ *and* $L_r(yi) = (xb_i, q_i)$.

In particular, if $\delta(q, L(x)) = true$, then S can be empty.

A run $r_{\mathcal{A},t}$ is *accepting* if for every infinite path π in $r_{\mathcal{A},t}$, $\text{Inf}(L_r(\pi)) \cap F \neq \emptyset$.

ABTA over binary trees: Example

$AG(p_1 \rightarrow AFp_2)$

$\mathcal{A} = (Q, 2^{\{p_1,p_2\}}, \delta, q_0, F)$

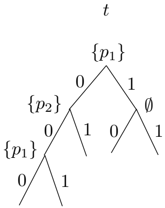
$Q = \{q_0, q_1\} \qquad F = \{q_0\}$

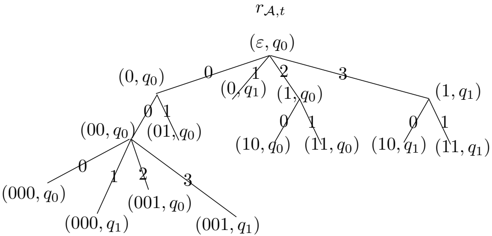
$$\begin{array}{r} \delta(q_0,\emptyset) \\ \delta(q_0,\{p_2\}) \\ \delta(q_0,\{p_1,p_2\}) \end{array} = (0,q_0) \wedge (1,q_0)$$

$$\delta(q_0,\{p_1\}) = (0,q_0) \wedge (0,q_1) \wedge (1,q_0) \wedge (1,q_1)$$

$$\begin{array}{r} \delta(q_1,\emptyset) \\ \delta(q_1,\{p_1\}) \end{array} = (0,q_1) \wedge (1,q_1)$$

$$\delta(q_1,\{p_2\}) = \delta(q_1,\{p_1,p_2\}) = true$$





Finitely-branching trees

Recall: A tree domain $D \subseteq \mathbb{N}^*$ s.t.

- $\forall xi \in \mathbb{N}^*$, if $xi \in D$, then $x \in D$ as well,
- $\forall xi \in \mathbb{N}^*$, if $xi \in D$, then $xj \in D$ for every $j : 0 \leq j < i$.

A tree domain D is *finitely branching* if

$$\forall x \in D, \exists n \in \mathbb{N} \ s.t. \ \forall m \geq n, \ xm \notin D.$$

A *finitely-branching tree* t over 2^{AP} is a pair (D, L) s.t.

$$D \text{ is a finitely branching tree domain and } L : D \to 2^{AP}.$$

Alternating automata over finitely-branching trees

Transition conditions over Q (TC^Q):

- $true, false \in TC^Q$,
- $\forall p \in AP$, $p, \neg p \in TC^Q$,
- for every $q_1, q_2 \in Q$, $q_1 \vee q_2, q_1 \wedge q_2 \in TC^Q$,
- for every $q \in Q$, $q, \Diamond q, \Box q \in TC^Q$.

Alternating automata over finitely-branching trees

An *alternating Büchi automaton* over finitely-branching trees (ABTA) $\mathcal{A}$ is a tuple $(Q, 2^{AP}, \delta, q_0, F)$ where $\delta : Q \rightarrow TC^Q$.

A run of an ABTA $\mathcal{A}$ over a (finitely-branching) tree $t = (D, L)$ is a winning strategy for Player 0 in the Büchi game $\mathcal{G} = (V_0, V_1, E, F \cup \{q_\top\})$, where

- $V_0 \subseteq D \times Q \cup \{q_\top\}$ s.t. $q_\top \in V_0$, and $(x, q) \in V_0$ iff
  - $\delta(q) = false$, or
  - $\delta(q) = p$ and $p \notin L(x)$, or
  - $\delta(q) = \neg p$ and $p \in L(x)$, or
  - $\delta(q) = q'$, or
  - $\delta(q) = q_1 \vee q_2$, or
  - $\delta(q) = \Diamond q'$.
- $V_1 \subseteq D \times Q \cup \{q_\bot\}$ s.t. $q_\bot \in V_1$, and $(x, q) \in V_1$ iff
  - $\delta(q) = true$, or
  - $\delta(q) = p$ and $p \in L(x)$, or
  - $\delta(q) = \neg p$ and $p \notin L(x)$, or
  - $\delta(q) = q_1 \wedge q_2$, or
  - $\delta(q) = \Box q'$.

Alternating automata over finitely-branching trees

An *alternating Büchi automaton* over finitely-branching trees (ABTA) $\mathcal{A}$ is a tuple $(Q, 2^{AP}, \delta, q_0, F)$ where $\delta : Q \to TC^Q$.

A run of an ABTA $\mathcal{A}$ over a (finitely-branching) tree $t = (D, L)$ is a winning strategy for Player 0 in the Büchi game $\mathcal{G} = (V_0, V_1, E, F \cup \{q_\top\})$, where

- E is defined as follows: $(q_\perp, q_\perp), (q_\top, q_\top) \in E$, and for every $(x, q) \in V_0 \cup V_1$,
  - if $\delta(q) = false$, or $\delta(q) = p$ and $p \notin L(x)$, or $\delta(q) = \neg p$ and $p \in L(x)$, then $((x, q), q_\perp) \in E$,
  - if $\delta(q) = true$, or $\delta(q) = p$ and $p \in L(x)$, or $\delta(q) = \neg p$ and $p \notin L(x)$, then $((x, q), q_\top) \in E$,
  - if $\delta(q) = q'$, then $((x, q), (x, q')) \in E$,
  - if $\delta(q) = q_1 \vee q_2$ (or $q_1 \wedge q_2$), then $((x, q), (x, q_1)), ((x, q), (x, q_2)) \in E$,
  - if $\delta(q) = \Diamond q'$ (or $\Box q'$), then for every children xi of x, $((x, q), (xi, q')) \in E$.

Remark: (V_0, V_1, E) defined above may not be a bipartite graph.

Acceptance:

> $\mathcal{A}$ *accepts* t *iff Player 0 has a winning strategy in* $\mathcal{G}$ *starting from* (ε, q_0).

ABTA over finitely-branching trees: Example

$AG(p_1 \rightarrow AFp_2)$

$\mathcal{A} = (Q, 2^{\{p_1,p_2\}}, \delta, q_0, F)$

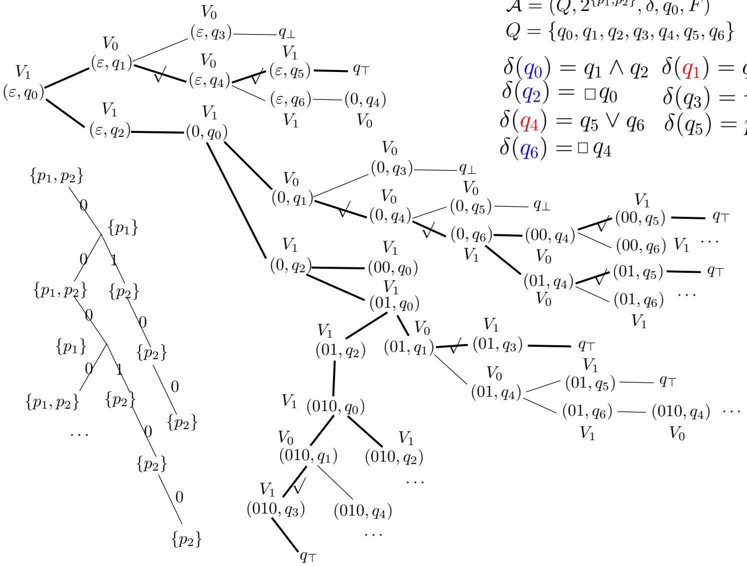
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$ $F = \{q_0\}$

$\delta(q_0) = q_1 \wedge q_2$ $\delta(q_1) = q_3 \vee q_4$

$\delta(q_2) = \Box q_0$ $\delta(q_3) = \neg p_1$

$\delta(q_4) = q_5 \vee q_6$ $\delta(q_5) = p_2$

$\delta(q_6) = \Box q_4$



Unwinding of Kripke structures

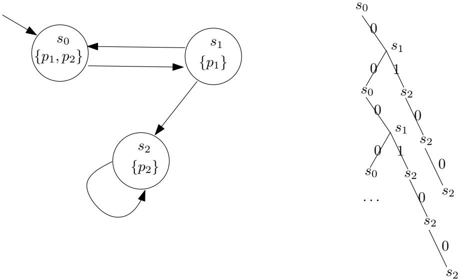
Let $\mathcal{S} = (S, AP, \to, \{s_0\}, L)$ be a Kripke structure.
$\forall s \in S$, let $suc(s)$ denote the set of successors of s.
Moreover, we assume that the states in $suc(s)$ are ordered.

$\mathcal{S}$ can be seen as an infinite tree $T_{\mathcal{S}} = (D_{\mathcal{S}}, L_{\mathcal{S}})$ as follows.

- $L_{\mathcal{S}}(\varepsilon) = s_0$,
- for every $y \in D_{\mathcal{S}}$, if $L_{\mathcal{S}}(y) = s$ and $suc(s) = \{s'_0, \dots, s'_k\}$, then for every $i : 0 \leq i \leq k$, $yi \in D_{\mathcal{S}}$ and $L_{\mathcal{S}}(yi) = s'_i$.

We can also view $T_{\mathcal{S}}$ as a tree over the alphabet 2^{AP}:
Replace $L_{\mathcal{S}}(y) = s$ with $L_{\mathcal{S}}(y) = L(s)$.

Example:



ABTA interpreted over Kripke structures

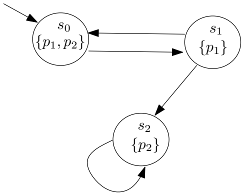
Suppose $\mathcal{A} = (Q, 2^{AP}, \delta, q_0, F)$ be an ABTA over finitely-branching trees and $\mathcal{S} = (S, AP, \rightarrow, s_0, L)$ be a Kripke structure.

A run of $\mathcal{A}$ over $\mathcal{S}$ is a run of $\mathcal{A}$ over $T_{\mathcal{S}}$.

As a matter of fact, a run of $\mathcal{A}$ over $\mathcal{S}$ can be defined by the winning strategies of Player 0 in the Büchi game $\mathcal{G}' = (V_0', V_1', E', (S \times F) \cup \{q_\top\})$, where

- $V_0' \subseteq S \times Q \cup \{q_\top\}$ and $V_1' \subseteq S \times Q \cup \{q_\bot\}$ are defined similar to V_0 and V_1 in $\mathcal{G}$,
- E is defined as follows: $(q_\bot, q_\bot), (q_\top, q_\top) \in E$, and for every $(s, q) \in V_0' \cup V_1'$,
    - if $\delta(q) = false$, or $\delta(q) = p$ and $p \notin L(s)$, or $\delta(q) = \neg p$ and $p \in L(s)$, then $((s,q), q_\bot) \in E$,
    - if $\delta(q) = true$, or $\delta(q) = p$ and $p \in L(s)$, or $\delta(q) = \neg p$ and $p \notin L(s)$, then $((s,q), q_\top) \in E$,
    - if $\delta(q) = q'$, then $((s,q),(s,q')) \in E$,
    - if $\delta(q) = q_1 \vee q_2$ (or $q_1 \wedge q_2$), then $((s,q),(s,q_1)), ((s,q),(s,q_2)) \in E$,
    - if $\delta(q) = \Diamond q'$ (or $\Box q'$), then for every successor s' of s, $((s,q),(s',q')) \in E$.

ABTA over Kripke structures: Example



$AG(p_1 \rightarrow AFp_2)$

$\mathcal{A} = (Q, 2^{\{p_1,p_2\}}, \delta, q_0, F)$

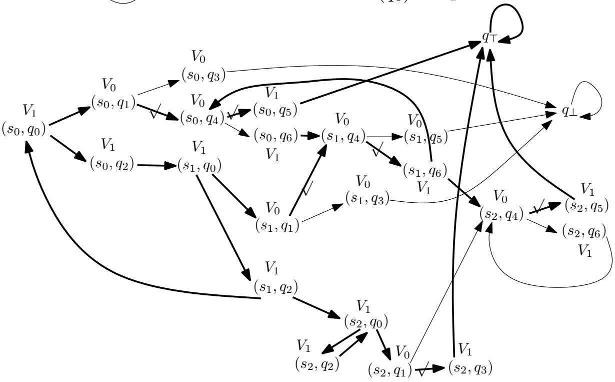
$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$ $F = \{q_0, q_2\}$

$\delta(q_0) = q_1 \wedge q_2$ $\delta(q_1) = q_3 \vee q_4$

$\delta(q_2) = \Box q_0$ $\delta(q_3) = \neg p_1$

$\delta(q_4) = q_5 \vee q_6$ $\delta(q_5) = p_2$

$\delta(q_6) = \Box q_4$



Weak alternating Büchi tree automata (WABTA)

A WABTA $\mathcal{A}$ (over Kripke structures) is a ABTA $(Q, 2^{AP}, \delta, q_0, F)$ s.t.

- Q is partitioned into n pairwise-disjoint subsets $Q_1, \ldots, Q_n$,
- there is partial order $\leq$ among $Q_1, \ldots, Q_n$ s.t. $\forall q \in Q_i, q' \in Q_j$, if q' occurs in $\delta(q)$, then $Q_j \leq Q_i$,
- for every Q_i, either $Q_i \subseteq F$ or $Q_i \cap F = \varnothing$.

Observation.

Every infinite path in a run finally get trapped in some Q_i.
The infinite path satisfies the acceptance condition iff $Q_i \subseteq F$.

Example:

The ABTA $\mathcal{A}$ for $AG(p_1 \rightarrow AFp_2)$ is in fact a WABTA:

- $\delta(q_0) = q_1 \wedge q_2$, $\delta(q_1) = q_3 \vee q_4$, $\delta(q_2) = \Box(q_0)$, $\delta(q_3) = \neg p_1$,
- $\delta(q_4) = q_5 \vee q_6$, $\delta(q_5) = p_2$, $\delta(q_6) = \Box q_4$,
- $F = \{q_0, q_2\}$.

The partition and the partial order:

$$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq \begin{array}{c} Q_3 = \{q_3\} \\ Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\} \end{array} .$$

Outline

1. Linear temporal logic (LTL)
2. LTL model checking: Automata theoretical approach
3. Computation tree logic (CTL)
4. (Weak) alternating tree automata
5. CTL model checking: Automata theoretical approach

CTL model checking: Automata-theoretic approach

W.l.o.g. in CTL model checking problem for $\mathcal{S} = (S, AP, \rightarrow, I, L)$ and φ, we assume that I is a singleton.

Automata-theoretical approach to CTL model checking:

Let $\mathcal{S} = (S, AP, \rightarrow, s_0, L)$ be a Kripke structure and φ be a CTL formula.

1. construct a WABTA $\mathcal{A}_\varphi = (Q, 2^{AP}, \delta, q_0, F)$ from φ in linear time,
2. construct the Büchi game $\mathcal{G}' = (V_0', V_1', E', (S \times F) \cup \{q_\top\})$ in time $O(\|\mathcal{A}_\varphi\| \times \|\mathcal{S}\|)$,
3. decide whether Player 0 has a winning strategy in $\mathcal{G}'$ starting from (s_0, q_0) in time $O(\|\mathcal{G}'\|)$.

Remark: In the third step above, the fact that $\mathcal{A}_\varphi$ is a WABTA is used.

Therefore, by using automata-theoretic approach, we get the following result.

Theorem. Given a Kripke structure $\mathcal{S}$ and a CTL formula φ, the problem whether $\mathcal{S} \models \varphi$ can be decided in time $O(\|\mathcal{S}\| \times |\varphi|)$.

From CTL to WABTA

Proposition. Given a CTL formula φ, a WABTA $\mathcal{A}_\varphi$ can be constructed in linear time s.t. $L(\mathcal{A}_\varphi)$ is the set of Kripke structures satisfying φ.

Proof.

$\mathcal{A}_\varphi = (sub(\varphi), 2^{AP}, \delta, q_0, F)$, where

- $q_0 = \varphi$, $F = \{\psi_1 R \psi_2 \mid \psi_1 R \psi_2 \in \mathrm{cl}(\varphi)\}$,
- $\{\varphi_1\} \leq \{\varphi_2\}$ iff $\varphi_1 \in sub(\varphi_2)$,
- and δ is defined as follows:
  - $\delta(true) = true$, $\delta(false) = false$,
  - $\delta(p) = p$, $\delta(\neg p) = \neg p$,
  - $\delta(\varphi_1 \vee \varphi_2) = \varphi_1 \vee \varphi_2$, $\delta(\varphi_1 \wedge \varphi_2) = \varphi_1 \wedge \varphi_2$,
  - $\delta(EX\varphi_1) = \Diamond\varphi_1$, $\delta(AX\varphi_1) = \Box\varphi_1$,
  - $\delta(E\varphi_1 U \varphi_2) = \varphi_2 \vee (\varphi_1 \wedge \Diamond E\varphi_1 U \varphi_2)$, $\delta(A\varphi_1 U \varphi_2) = \varphi_2 \vee (\varphi_1 \wedge \Box A\varphi_1 U \varphi_2)$,
  - $\delta(E\varphi_1 R \varphi_2) = \varphi_2 \wedge (\varphi_1 \vee \Diamond E\varphi_1 R \varphi_2)$, $\delta(A\varphi_1 R \varphi_2) = \varphi_2 \wedge (\varphi_1 \vee \Box A\varphi_1 R \varphi_2)$.

Remark: $\delta(E\varphi_1 U \varphi_2) = \varphi_2 \vee (\varphi_1 \wedge \Diamond E\varphi_1 U \varphi_2)$ are abbrev. of transitions
$\delta(E\varphi_1 U \varphi_2) = \varphi_2 \vee q$, $\delta(q) = \varphi_1 \wedge q'$, $\delta(q') = \Diamond E\varphi_1 U \varphi_2$,
where q, q' are new introduced states in the same partition as $E\varphi_1 U \varphi_2$. □

The special structure of Büchi game $\mathcal{G}'$

Let $\mathcal{S} = (S, AP, \to, s_0, L)$ be a Kripke structure and $\mathcal{A}_\varphi = (sub(\varphi), 2^{AP}, \delta, q_0, F)$ be a WABTA.

The special structure of $\mathcal{A}_\varphi$ induces a special structure of the game $\mathcal{G}' = (V_0', V_1', E', (S \times F) \cup \{q_\top\})$:

- $V_0' \cup V_1'$ can be partitioned into $(S \times \{\psi\})_{\psi \in sub(\varphi)}, \{q_\bot\}, \{q_\top\}$,
- $S \times \{\psi_1\} \leq S \times \{\psi_2\}$ iff $\{\psi_1\} \leq \{\psi_2\}$, $\forall \psi \in sub(\varphi), q_\top, q_\bot \leq S \times \{\psi\}$,
- E' is non-increasing wrt. $\leq$.

Weak Büchi game:

A Büchi game (V_0, V_1, E, F) is weak if $V_0 \cup V_1$ can be partitioned into subsets $V_1', \ldots, V_n'$ s.t.

- $\forall q \in V_i', q' \in V_j'.\ (q, q') \in E$ implies $V_j' \leq V_i'$.
- $\forall i.$ either $V_i' \subseteq F$ or $V_i' \cap F = \varnothing$.

Theorem. Weak Büchi game can be solved in linear time.

Solving weak Büchi game in linear time

Theorem. Weak Büchi game can be solved in linear time.

Proof.

Let $\mathcal{G} = (V_0, V_1, E, F)$ be a weak Büchi game with partitions $V_1', \ldots, V_n'$.
W.l.o.g. we assume that

- for every $v \in V_0 \cup V_1$, $vE \neq \emptyset$,
- for every i, j, if $V_i' \geq V_j'$, then $i \leq j$.

□

Solving weak Büchi game in linear time

Theorem. Weak Büchi game can be solved in linear time.

Proof.

Let $\mathcal{G} = (V_0, V_1, E, F)$ be a weak Büchi game with partitions $V'_1, \ldots, V'_n$.

The algorithm.

Compute $I : V_0 \cup V_1 \to \{true, false\}$ as follows.

> *Initially, set* $I(v) = \bot$ *(undefined) for every* $v \in V_0 \cup V_1$.
> *For* i *from* n *to* 1, *do the following computation.*
>
> 1. *For every* $v \in V'_i$ *s.t.* $I(v) = \bot$, *set* $I(v) = true$ *iff* $V'_i \subseteq F$.
> 2. *Repeat the following procedure until* $I(v)$*'s no more updated:*
>   *For every* $v \in V_0 \cup V_1$,
>   - $v \in V_0$:
>   - *if* $\exists$ *a successor of* v, *say* v', *s.t.* $I(v') = true$, *then set* $I(v) = true$,
>   - *if every successor* v' *of* v *satisfy* $I(v') = false$, *then set* $I(v) = false$.
>   - $v \in V_1$:
>   - *if* $\exists$ *a successor* v' *of* v *satisfy* $I(v') = false$, *then set* $I(v) = false$,
>   - *if every successor* v' *of* v *satisfy* $I(v') = true$, *then set* $I(v) = true$.

Claim. Player 0 has a winning strategy in $\mathcal{G}$ starting from q_0 iff $I(q_0) = true$.

Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

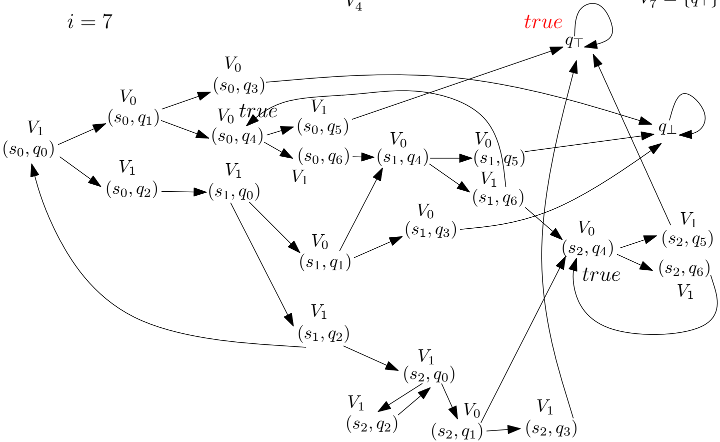
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

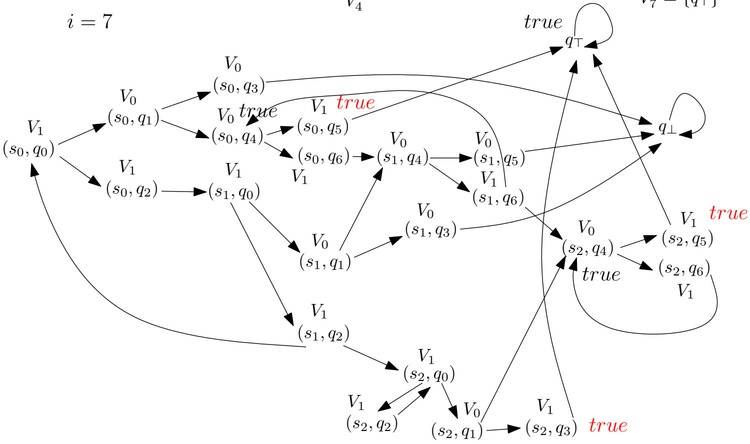
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

V'_1 $\qquad$ V'_2 $\qquad$ $Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

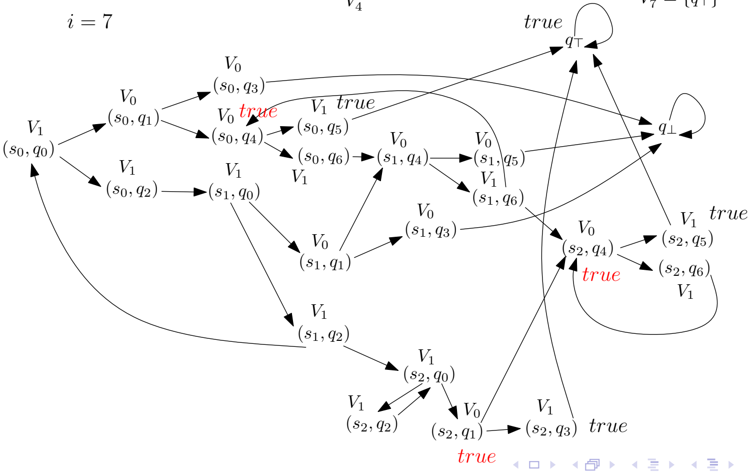
V'_3 $\qquad$ V'_5

V'_4

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

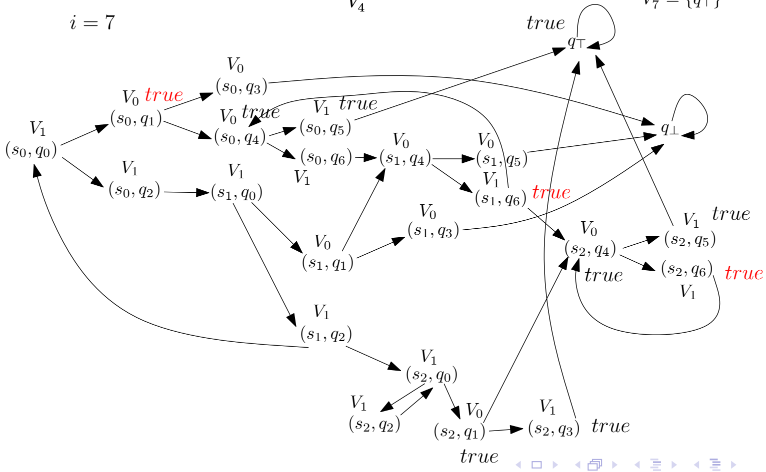
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\perp\}$

$V'_7 = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

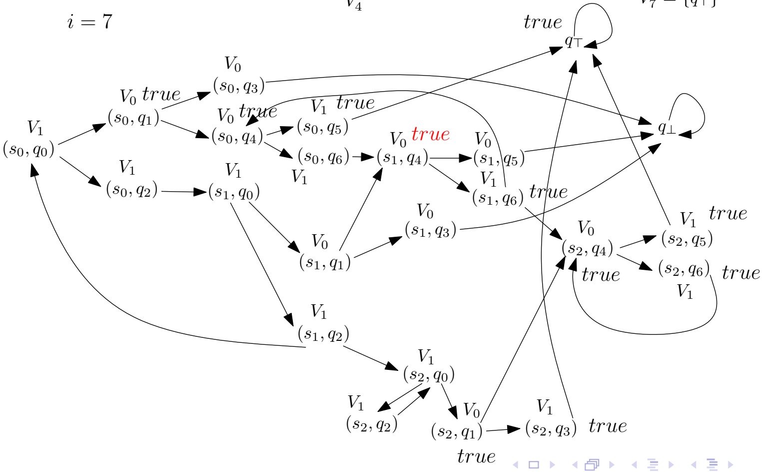
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V_1' V_2' V_3' V_4' V_5'

$V_6' = \{q_\perp\}$

$V_7' = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

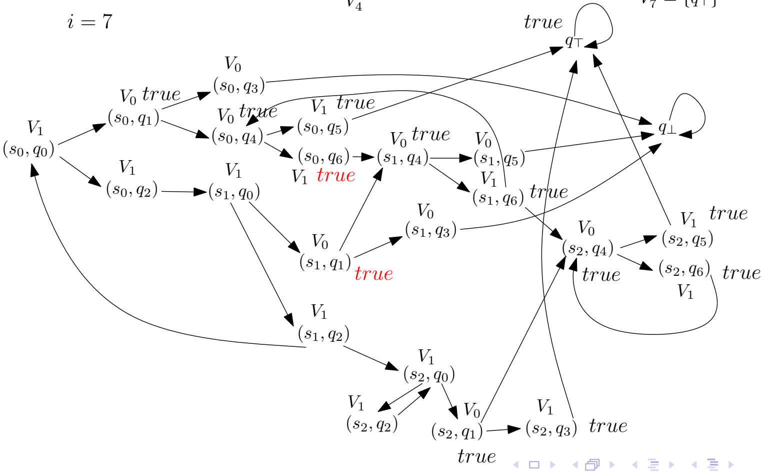
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\perp\}$

$V'_7 = \{q_\top\}$

$i = 7$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

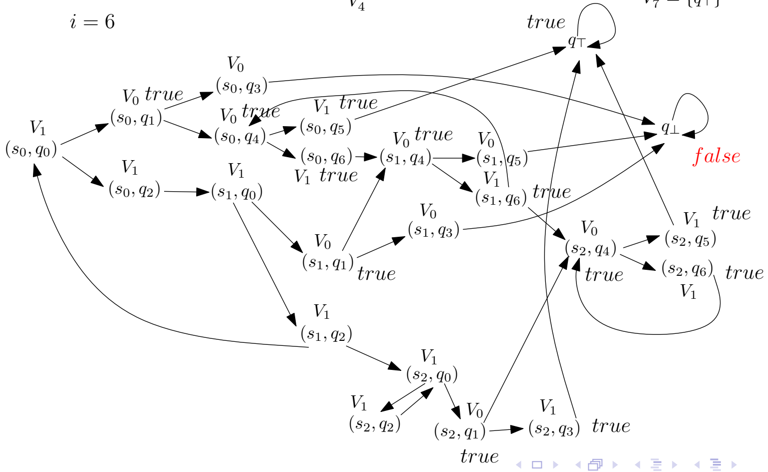
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 6$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq Q_3 = \{q_3\}$

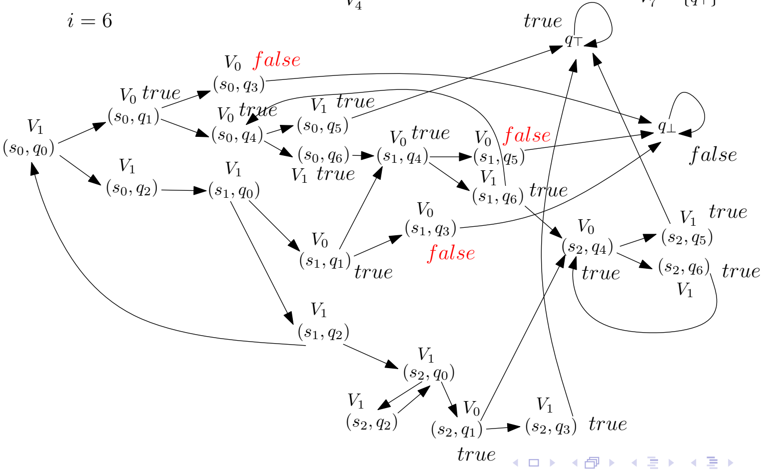
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 6$



Solving weak Büchi game: Example

$F = \{q_0, q_2\}$

$Q_1 = \{q_0, q_2\} \geq Q_2 = \{q_1\} \geq$ $Q_3 = \{q_3\}$

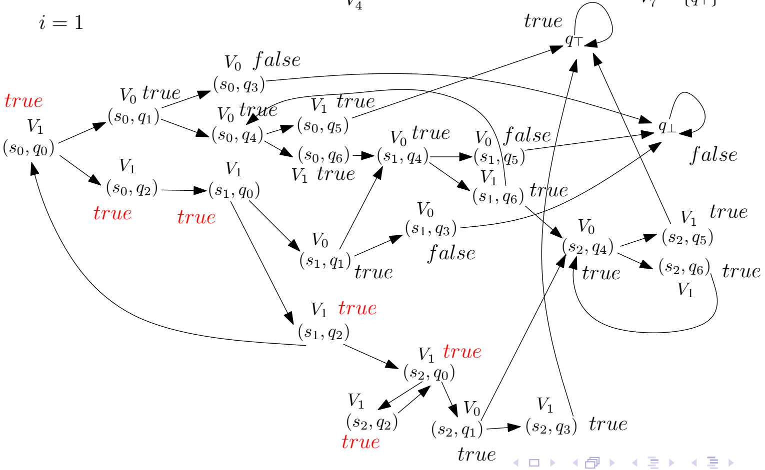
$Q_4 = \{q_4, q_6\} \geq Q_5 = \{q_5\}$

V'_1 V'_2 V'_3 V'_4 V'_5

$V'_6 = \{q_\bot\}$

$V'_7 = \{q_\top\}$

$i = 1$



References

The main references for these two lectures.

LTL model checking:

- Christel Baier, Joost-Pieter Katoen, Principles of Model Checking, The MIT Press, 2008.

CTL model checking:

- Orna Kupferman, Moshe Vardi, Pierre Wolper, An automata-theoretic approach to branching-time model checking, Journal of ACM, Vol. 47, No. 2, 312-360, 2000.
- Daniel Kirsten, Alternating tree automata and parity games, Chapter 9, Automata, logics, and infinite games, LNCS 2500, 2002.
- Javier Esparza, Orna Kupferman, Moshe Y. Vardi, Automata-theoretic verification, `www.cs.rice.edu/~vardi/papers/hba11.pdf`

What remains . . .

Applications to XML document processing