Bernd Finkbeiner Date: July 12, 2011

Automata, Games and Verification: Lecture 14

Reference: *An Automata-Theoretic Approach to Branching-Time Model Checking* by Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper

23 Alternating Tree Automata

Definition 1 *An alternating tree automaton over binary* Σ*-trees is a tuple* $\mathcal{A} = (S, s_0, \delta, \varphi)$*:*

- S*: finite set of states*
- $s_0 \in S$
- $\delta : S \times \Sigma \to \mathbb{B}^+(\{0,1\} \times S)$ *is the transition function.*
- φ*: acceptance condition (Büchi, parity, ...)*

More general: set of directions $\mathcal{D} = \{0, \ldots, k-1\}$, $T \subseteq \mathcal{D}^*$, degree $d : \mathcal{D}^* \to \{1, \ldots, k\}$

Definition 2 *An alternating tree automaton over* Σ*-trees is a tuple* $\mathcal{A} = (S, s_0, \delta, \varphi)$*:*

- S*: finite set of states*
- $s_0 \in S$
- $\delta : S \times \Sigma \times \{1, \ldots k\} \to \mathbb{B}^+(\{0, 1, \ldots k-1\} \times S)$ *is the transition function.*
- φ*: acceptance condition (Büchi, parity, ...)*

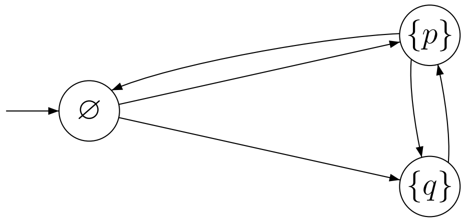
Definition 3 *A run of a tree automaton* $\mathcal{A}$ *on a* Σ*-tree* v *is a* $\mathcal{D}^* \times S$*-tree* (T, r)*, s.t.*

1. $r(\epsilon) = (\epsilon, s_0)$
2. *Let* $y \in T$ *with* $r(y) = (x, q)$ *and* $\delta(q, v(x), d(x)) = \theta$*. Then there is a (possibly empty) set* $Q = \{(c_0, q'_0), (c_1, q'_1), \ldots, (c_n, q'_n)\} \subseteq \{0, \ldots, d(x)-1\} \times S$*, such that the following hold:*
   - $Q \models \theta$
   - *for all* $0 \leq i \leq n$*, we have* $y \cdot i \in T$ *and* $r(y \cdot i) = (x \cdot c_i, q'_i)$*.*

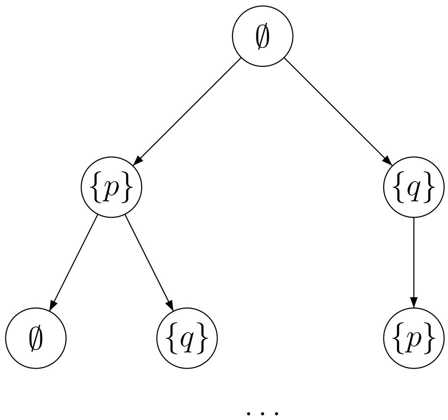
Definition 4 *A run is accepting if every branch is accepting (by* φ*). A* Σ*-tree is accepted if there exists an accepting run.*

Tree automata on Kripke structures

Example: For a pointed Kripke structure $(\mathcal{M}, s_0)$:



we build a computation tree t



Let k be the max number of successors in $\mathcal{M} = (S, R, L)$. Define a mapping: $f : \{0, \ldots, k-1\}^* \to S$:

- $f(\epsilon) = s_o$
- Assume there is, for each $s \in S$, a fixed order on the successors $(s, s'_1), (s, s'_2), \ldots \in E$ $f(w \cdot i) = s'_i$ where s'_i is the ith successor of $s = f(w)$.

Definition 5 *For a pointed Kripke structure $(\mathcal{M}, s_0)$ over AP with $\mathcal{M} = (S, R, L)$, the computation tree of $(\mathcal{M}, s_0)$ is a 2^{AP}-tree (T, t) with $t(v) = L(f(v))$ and $d(v) = d(f(v))$ for all $v \in T$.*

Theorem 1 *The computation tree of a pointed Kripke structure $(\mathcal{M}, q_0), \mathcal{M} = (S_\mathcal{M}, R, L)$ is accepted by an alternating tree automaton $\mathcal{A} = (S_\mathcal{A}, s_0, \delta, \varphi)$ iff Player Accept has a winning strategy from (s_0, q_0) in the following game:*

- $V_0 = S_\mathcal{A} \times S_\mathcal{M}$

- $V_0 = S_\mathcal{A} \times 2^{\{0,\ldots,k-1\}\times S_\mathcal{A}} \times S_\mathcal{M}$
- $E = \{((s,q),(s,\eta,q)) \mid \eta \models \delta(s, L(q), d(q))\}$
  $\cup\{((s,\eta,q),(s',q')) \mid (i,q') \in \eta, s' \textit{ is the ith successor of } s\}$
- *winning condition: φ applied to the first component*

CTL

Translation from CTL formula φ to alternating Büchi tree automaton $\mathcal{A}_\varphi$:

- S =closure(φ):= set of all subformulas and their negations
- for $p \in AP$:
  - $\delta(p,\sigma,k) = \mathit{true}$ if $p \in \sigma$
  - $\delta(p,\sigma,k) = \mathit{false}$ if $p \notin \sigma$
  - $\delta(\neg p,\sigma,k) = \mathit{false}$ if $p \in \sigma$
  - $\delta(\neg p,\sigma,k) = \mathit{true}$ if $p \notin \sigma$
- $\delta(\varphi \wedge \psi, \sigma, k) = \delta(\varphi,\sigma,k) \wedge \delta(\psi,\sigma,k)$
- $\delta(\varphi \vee \psi, \sigma, k) = \delta(\varphi,\sigma,k) \vee \delta(\psi,\sigma,k)$
- $\delta(\mathrm{AX}\varphi,\sigma,k) = \bigwedge_{c=0}^{k-1}(c,\varphi)$
- $\delta(\mathrm{EX}\varphi,\sigma,k) = \bigvee_{c=0}^{k-1}(c,\varphi)$
- $\delta(\mathrm{A}\varphi\ \mathcal{U}\ \psi,\sigma,k) = \delta(\psi,\sigma,k) \vee (\delta(\varphi,\sigma,k) \wedge \bigwedge_{c=0}^{k-1}(c, \mathrm{A}\varphi\ \mathcal{U}\ \psi)$
- $\delta(\mathrm{E}\varphi\ \mathcal{U}\ \psi,\sigma,k) = \delta(\psi,\sigma,k) \vee (\delta(\varphi,\sigma,k) \wedge \bigvee_{c=0}^{k-1}(c, \mathrm{E}\varphi\ \mathcal{U}\ \psi)$
- $\delta(\neg\varphi,\sigma,k) = \overline{\delta(\varphi,\sigma,k)}$

Theorem 2 *For every CTL formula φ and a set of directions $\mathcal{D}$ there is an alternating Büchi tree automaton $\mathcal{A}_\varphi$ such that $\mathcal{L}(\mathcal{A}_{\mathcal{D},\varphi})$ is exactly the set of $\mathcal{D}$-branching trees that satisfy φ.*

Alternation-free μ-calculus

Guarded formulas: A μ-calculus formula is *guarded* if it is in normal form and for every quantified atomic proposition p, all occurrences are in the scope of a modality that is in the scope of the quantifier.

Example: $\mu y.(p \vee \Diamond y)$ is guarded, $\Diamond \mu y.(p \vee y)$ is not guarded. ✚

Theorem 3 *For every μ calculus formula in normal form there is an equivalent guarded formula.*

Proof:

- Function *new* : μ-calculus formulas $\times\{\mu,\nu\} \times AP \to$ μ-calculus formulas:
  - $new(p, \mu, p) = false$
  - $new(p, \nu, p) = true$
  - $new(\varphi \wedge \psi, \lambda, p) = new(\varphi, \lambda, p) \wedge new(\psi, \lambda, p)$
  - $new(\varphi \vee \psi, \lambda, p) = new(\varphi, \lambda, p) \vee new(\psi, \lambda, p)$
  - For all other formulas φ: $new(\varphi, \lambda, p) = \varphi$
- Note that the definition of the *new* function ensures that $\lambda y.\varphi(y)$ is semantically equivalent to $\lambda y.new(\varphi, \mu, y)(y)$ for all μ-calculus formulas $\varphi(y)$.
- Translation: Starting from the innermost quantified subformulas,
  replace $\lambda y.\varphi(y)$ by $new(\varphi, \lambda, y)(\lambda y.new(\varphi, \lambda, y)(y))$ (1)
- Note that in $new(\varphi, \mu, y)(y)$, all occurrences of y are in the scope of a modality, hence in (1), all occurrences of variables (e.g. z) that are in the scope of a fixpoint operator are also in the scope of a modality.

■

Closure $cl(\varphi)$ of a μ-calculus formula φ:

- $\varphi \in cl(\varphi)$
- if $\psi \vee \eta \in cl(\varphi)$ then $\psi, \eta \in cl(\varphi)$
- if $\psi \wedge \eta \in cl(\varphi)$ then $\psi, \eta \in cl(\varphi)$
- if $\Diamond\psi \in cl(\varphi)$ then $\psi \in cl(\varphi)$
- if $\Box\psi \in cl(\varphi)$ then $\psi \in cl(\varphi)$
- if $\mu y.\psi(y) \in cl(\varphi)$ then $\psi(\mu y.\psi(y)) \in cl(\varphi)$
- if $\nu y.\psi(y) \in cl(\varphi)$ then $\psi(\nu y.\psi(y)) \in cl(\varphi)$

Alternation-free μ-calculus: no ν between $\mu y.$ and y; no μ between $\nu y.$ and y.
Translation from a guarded alternation-free μ-calculus formula φ to an alternating Büchi tree automaton $\mathcal{A}_\varphi$:

- $\delta(p, \sigma, k) = true$ if $p \in \sigma$
- $\delta(p, \sigma, k) = false$ if $p \notin \sigma$
- $\delta(\neg p, \sigma, k) = false$ if $p \in \sigma$
- $\delta(\neg p, \sigma, k) = true$ if $p \notin \sigma$
- $\delta(\varphi \wedge \psi, \sigma, k) = \delta(\varphi, \sigma, k) \wedge \delta(\psi, \sigma, k)$
- $\delta(\varphi \vee \psi, \sigma, k) = \delta(\varphi, \sigma, k) \vee \delta(\psi, \sigma, k)$

- $\delta(\Box\varphi, \sigma, k) = \bigwedge_{c=0}^{k-1}(c, \varphi)$
- $\delta(\Diamond\varphi, \sigma, k) = \bigvee_{c=0}^{k-1}(c, \varphi)$
- $\delta(\mu y.\psi(y), \sigma, k) = \delta(\psi(\mu y.\psi(y)), \sigma, k)$
- $\delta(\nu y.\psi(y), \sigma, k) = \delta(\psi(\nu y.\psi(y)), \sigma, k)$

Note that since φ is guarded, the definition is not circular.

Let $\approx$ be an equivalence relation on μ-calculus formulas such that $\varphi \approx \psi$ if $\varphi \in cl(\psi)$ and $\psi \in cl(\varphi)$.

$F = \{$set of formulas that are equivalent to some formula $\nu y.\psi(y) \in cl(\varphi)\}$

Theorem 4 *For every alternation-free μ-calculus formula φ and a set of directions $\mathcal{D}$ there is an alternating Büchi tree automaton $\mathcal{A}_\varphi$ such that $\mathcal{L}(\mathcal{A}_{\mathcal{D},\varphi})$ is exactly the set of $\mathcal{D}$-branching trees satisfying φ.*