Tree Automata and Applications: TD4
Alternating Tree Automata and Hedge Automata

Emile Contal
http://econtal.perso.math.cnrs.fr/teaching

October 11, 2013

1 Alternating Tree Automaton

Definition 1. A (top-down) Alternating Tree Automaton (ATA) $\mathcal{A}$ is given by $(\mathcal{Q}, \mathcal{F}, \mathcal{Q}_f, \delta)$ where $\delta : \mathcal{Q} \times \mathcal{F} \to \mathbb{B}\big(\mathcal{Q} \times \{1, \ldots, n\}\big)$ with $\delta(q, f) \in \mathbb{B}\big(\mathcal{Q} \times \{1, \ldots, Arity(f)\}\big)$.

A computation of $\mathcal{A}$ on a term $t \in T(\mathcal{F})$ in state q is a tree labelled by $\mathcal{Q}$ such that the root is labelled by q and if $t = f(t_1, \ldots, t_n)$ and $\delta(q, f) = \phi$, then there exists a subset $S \subseteq \mathcal{Q} \times \{1, \ldots, n\}$ such that :

- $S \models \phi$,
- for all $(q_i, i) \in S$, there exists a child of the root ρ which is a computation on t_i in state q_i.

An accepting computation is a computation in a final state.

Definition 2. A bottom-up ATA has transition rules of the form $f(\phi_1, \ldots, \phi_n) \to q$ where $\phi_1, \ldots, \phi_n \in \mathbb{B}(\mathcal{Q})$. A computation ρ on t in a bottom-up ATA verifies:

- the root of ρ is labelled by (q, m),
- if $t = f(t_1, \ldots, t_n)$ there exists a rule $f(\phi_1, \ldots, \phi_n) \to q$ such that if $\rho_1, \ldots, \rho_m$ is the set of the children of the root of ρ and for all i, (q_i, m_i) is the label of the root of ρ_i, then:
  - for all $j = 1, \ldots, n$, $\{q \mid \exists i. m_i = j, q = q_i\} \models \phi_j$,
  - for all $j = 1, \ldots, n$, if $m_i = j$ then ρ_i is a computation of $(\mathcal{Q}, \mathcal{F}, \{q_i\}, \delta)$ on t_j.

Example exercise. Consider the automaton on the alphabet $\big\{f(,), a, b\big\}$ with final state q_2 and transition relation:

Δ	f	a	b
q_2	$\Big[\big((q_1,1) \wedge (q_2,2)\big) \vee \big((q_1,2) \wedge (q_2,1)\big)\Big] \wedge (q_4,1)$	$\top$	$\bot$
q_1	$\big((q_2,1) \wedge (q_2,2)\big) \vee \big((q_1,2) \wedge (q_1,1)\big)$	$\bot$	$\top$
q_4	$\big((q_3,1) \wedge (q_3,2)\big) \vee \big((q_4,1) \wedge (q_4,2)\big)$	$\top$	$\top$
q_3	$\big((q_3,1) \wedge (q_2,2)\big) \vee \big((q_4,1) \wedge (q_1,2)\big) \wedge (q_5,1)\big)$	$\bot$	$\top$
q_5	$\bot$	$\top$	$\bot$

Give a run of this automaton on the term $t = f\Big(f\big(b, f(a,b)\big), b\Big)$.

Exercise 1. Let $\mathcal{A}$ be an ATA. Prove that we can compute in polynomial time an ATA $\mathcal{A}^c$ such that $\mathcal{L}(\mathcal{A}^c) = \overline{\mathcal{L}(\mathcal{A})}$.

Give an example of bottom-up ATA $\mathcal{A}$ such that every bottom-up ATA recognizing the complementary language of $\mathcal{A}$ has an exponential size.

Exercise 2. Let $\mathcal{A}$ be an ATA, show that there exists a deterministic (bottom-up) tree automaton with exponential size recognizing the language of $\mathcal{A}$.

2 Horn Logic and Alternating Automata

Let $\mathcal{C}_A$ be the class of clauses in the form:

- $q_1(x_1), \dots, q_n(x_n) \to q\big(f(x_1, \dots, x_n)\big)$, where all x_i are distinct,
- $q_1(x), q_2(x) \to q(x)$.

Let $\mathcal{C}_1$ the class of clauses in the form:

$$q_1(x_{i_1}), \dots, q_k(x_{i_k}) \to q\big(f(x_1, \dots, x_n)\big) \ ,$$

where the $x_{i_1}, \dots, x_{i_k} \in \{x_1, \dots, x_n\}$ are not necessarily distinct.

Exercise 3. Let $C \in \mathcal{C}_1$ be a finite set of clauses, show that we can compute in polynomial time a bottom-up ATA $\mathcal{A}_C$ such that the language recognizable by $\mathcal{A}_C$ in state q is the interpretation of q in the smallest model of C.

Let $\mathcal{A}$ be a bottom-up ATA, show that we can compute in polynomial time an ATA which recognizes the same language.

Let $\mathcal{A}$ be a bottom-up ATA, show that we can compute in polynomial time a set of clauses $C_\mathcal{A} \in \mathcal{C}_A$ such that $\mathcal{A}$ accepts t in state q iff t is the interpretation of q in the smallest model of C_A.

Let $\mathcal{A}$ be an ATA, show that we can compute in polynomial time a set of clauses in $C_\mathcal{A} \in \mathcal{C}_\mathcal{A}$ for $\mathcal{A}$.

3 Two Way Alternating Automata

Definition 3. A Two Way Alternating Automaton $\mathcal{A}$ is given by $\mathcal{Q}$ a finite set of states, $\mathcal{Q}_f \subseteq \mathcal{Q}$ a set of final states and δ a transition function which maps a pair (q, f) to a formula in $\mathbb{B}\big(\mathcal{Q} \times \{-1, 0, \dots, Arity(f)\}\big)$.

A computation of $\mathcal{A}$ on t is a tree ρ labelled by $\mathcal{Q} \times \mathsf{Pos}(t)$ such that:

1. $\epsilon \in \mathsf{Pos}(\rho)$ and $\rho(\epsilon) = (q, \epsilon)$,
2. if $w \in \mathsf{Pos}(\rho)$ and $\rho(w) = (q, p)$ and $\delta\big(q, t(p)\big) = \phi$, then it exists $S = \big\{(q_1, d_1), \dots, (q_n, d_n)\big\}$ such that $S \models \phi$ and for all $(q_i, d_i) \in S$,
   - $w \cdot i \in \mathsf{Pos}(\rho)$,
   - if $d_i > 0$, then $p \cdot d_i \in \mathsf{Pos}(t)$ and $\rho(w \cdot i) = (q_i, p \cdot d_i)$,
   - if $d_i = 0$, then $\rho(w \cdot i) = (q_i, p)$,
   - if $d_i = -1$, then $\rho(w \cdot i) = (q_i, p')$ with $p = p' \cdot i$.

A computation is accepting if the root is labelled by a final state.

Example exercise. Consider the automaton on the alphabet $\{f(,), a, b\}$ with final state q_2 and transition relation:

Δ	f	a	b
q_2	$\Big[\big((q_1,1) \wedge (q_2,2)\big) \vee \big((q_1,2) \wedge (q_2,1)\big)\Big] \wedge (q_4,1)$	$\top$	$\bot$
q_1	$\big((q_2,1) \wedge (q_2,2)\big) \vee \big((q_1,2) \wedge (q_1,1)\big)$	$\bot$	$\top$
q_4	$\big((q_3,1) \wedge (q_3,2)\big) \vee \big((q_4,1) \wedge (q_4,2)\big)$	$\top$	$(q_1,-1)$
q_3	$\big((q_3,1) \wedge (q_2,2)\big) \vee \big((q_4,1) \wedge (q_1,2)\big) \wedge (q_5,1)\big)$	$\bot$	$\top$
q_5	$\bot$	$(q_1,-1)$	$\bot$

Give a run of this automaton on the term $t = f\Big(f\big(b, f(a,b)\big), b\Big)$.

Exercise 4. We want to show that the canonical correspondence between a Two Way Alternating Automaton $\mathcal{A}$ and Horn clauses is not correct. To simplify, we consider that the alphabet has only unary symbols and that the transitions of $\mathcal{A}$ are in the form $\delta(q,a) = (q', \pm 1)$ or $\delta(q,a) = \top$. Let $C_\mathcal{A}$ be the smallest set of clauses such that:

- if $\delta(q,a) = (q',1)$, then $q'(x) \to q(a(x)) \in C_\mathcal{A}$,
- if $\delta(q,a) = (q',-1)$, then $q'(a(x)) \to q(x) \in C_\mathcal{A}$,
- if $\delta(q,a) = \top$, then $\to q(a) \in C_\mathcal{A}$.

Define $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \emptyset$ but the alphabet for $C_\mathcal{A}$ is not empty.

4 Hedge Automata

Definition 4. A nondeterministic finite hedge automaton (NFHA) over Σ is a tuple $\mathcal{A} = (\mathcal{Q}, \Sigma, \mathcal{Q}_f, \Delta)$ where $\mathcal{Q}$ is a finite set of states, $\mathcal{Q}_f \subseteq \mathcal{Q}$ is a set of final states, and Δ is a finite set of transition rules of the type $a(R) \to q$, where $R \subseteq \mathcal{Q}^*$ is a regular language over $\mathcal{Q}$. These languages R occurring in the transition rules are called horizontal languages.

A run of $\mathcal{A}$ on a tree $t \in T(\Sigma)$ is a tree $r \in T(\mathcal{Q})$ with the same domain as t such that for each node $p \in \mathsf{Pos}(r)$ with $a = t(p)$ and $q = r(p)$ there is a transition rule $a(R) \to q$ of $\mathcal{A}$ with $r(p1) \cdot r(pn) \in R$, where n denotes the number of successors of p. In particular, to apply a rule at a leaf, the empty word ϵ has to be in the horizontal language of the rule.

An unranked tree t is accepted by $\mathcal{A}$ if there is a run r of $\mathcal{A}$ on t whose root is labeled by a final state, i.e. with $r(\epsilon) \in \mathcal{Q}_f$. The language $\mathcal{L}(\mathcal{A})$ of $\mathcal{A}$ is the set of all unranked trees accepted by $\mathcal{A}$.

Exercise 5. Let $\Sigma = \{or, and, not, 0, 1\}$. Define a NFHA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ is the set of all trees that form correct Boolean expressions evaluating to true.

Exercise 6. Let $\Sigma = \{a, b, c\}$. Define a NFHA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ is the set of all trees such that there exists two nodes labelled b whose greatest common ancestor is labeled c.