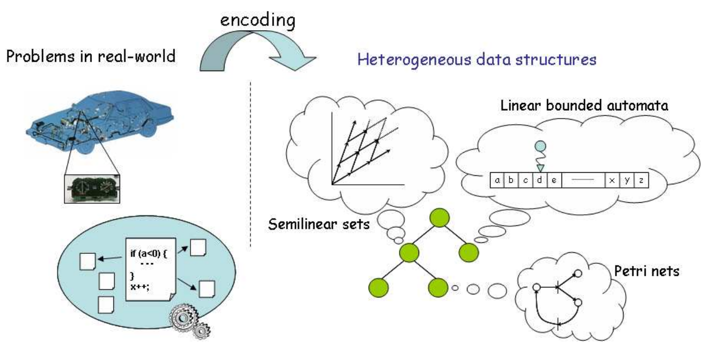
Introduction to Tree Language Theory

Hitoshi Ohsaki

National Institute of
Advanced Industrial Science and Technology (AIST)

seminar talk (8/10)

2009



VIII. Equational tree automata

- The AC-closure of a regular tree language is regular ?
- The set of the AC-closure of regular tree languages is closed under Boolean operations ?

Tree automata (TA) revisited

TA $(F, Q, Q_{\mathsf{fin}}, \Delta)$

F : signature

Q : finite set of state symbols such that $F \cap Q = \varnothing$

Q_{fin} : finite set $Q_{\mathsf{fin}}\,(\subseteq Q)$ of final states

Δ : finite set of transition rules with the following forms

$f(p_1, \ldots, p_n) \;\longrightarrow\; q$ [regular rule]

$p \;\longrightarrow\; q$ [epsilon rule]

if $f \in F_{(n)}$ & $p_1, \ldots, p_n, p, q \in Q$

regular tree automaton (RTA) if Δ does not contain an epsilon rule

Note

$\mathsf{C}(\mathsf{RTA}_F) = \mathsf{C}(\mathsf{TA}_F)$

Equational tree automata (ETA)

ETA $\mathcal{A} = (\mathcal{E}, Q, Q_{\mathsf{fin}}, \Delta)$, denoted by $\mathcal{A}_{\mathcal{E}}$ (for $\mathcal{E}$ to be explicit)

$\mathcal{E}$: equational theory (F, E)

F : signature

E : finite set of equations

Q : finite set of state symbols such that $F \cap Q = \varnothing$

Q_{fin} : finite set $Q_{\mathsf{fin}} (\subseteq Q)$ of final states

Δ : finite set of transition rules

– $\mathcal{A}_{\mathcal{E}}$ is regular if Δ does not contain an epsilon rule

Note

Proportionally equivalent notions ...

TA & ETA $\propto$ GTRS & EGTRS (G : ground) $\propto$ TRS & ETRS

Accepted trees

Given an ETA $\mathcal{A}_\mathcal{E}$: $(\mathcal{E}, Q, Q_{\text{fin}}, \Delta)$

$s \rightarrow_{\mathcal{A}_\mathcal{E}} t$ (move relation)	:	$\exists\, l \rightarrow r \in \Delta,\ C[\,] \in \mathcal{C}_{F \cup Q}$: $s =_\mathcal{E} C[l]$ & $t =_\mathcal{E} C[r]$
t is accepted by $\mathcal{A}_\mathcal{E}$	:	$\exists\, q \in Q_{\text{fin}} : t \rightarrow_{\mathcal{A}_\mathcal{E}} \cdots \rightarrow_{\mathcal{A}_\mathcal{E}} q$
tree language	:	some subset of $\mathcal{T}_F$
tree language $\mathcal{L}(\mathcal{A}_\mathcal{E})$	:	set of trees accepted by $\mathcal{A}_\mathcal{E}$
L is $\mathcal{E}$-regular tree language	:	$\exists$ regular $\mathcal{A}_\mathcal{E} : L = \mathcal{L}(\mathcal{A}_\mathcal{E})$
$\mathcal{E}$ is linear equational theory	:	$\mathcal{E}$ consists of linear equations
$\mathcal{E}$ is AC-theory	:	(See page 7, seminar talk 4)
$\mathcal{E}$ is A-theory	:	(See page 8, seminar talk 4)

Example

Consider $\mathcal{A} = (\mathcal{E}, Q, Q_{\text{fin}}, \Delta)$ where $\mathcal{E} = (F, E)$

F : $0 \quad 1 \quad \vee \quad \wedge \quad \neg$

E : $\vee(x,y) = \vee(y,x) \quad \vee(\vee(x,y),z) = \vee(x,\vee(y,z))$

$\wedge(x,y) = \wedge(y,x) \quad \wedge(\wedge(x,y),z) = \wedge(x,\wedge(y,z))$

Q : $q_0 \; q_1$

Q_{fin} : q_1

Δ : $0 \to q_0 \quad 1 \to q_1 \quad \neg(q_0) \to q_1 \quad \neg(q_1) \to q_0$

$\vee(q_0, q_0) \to q_0 \quad \vee(q_0, q_1) \to q_1 \quad \vee(q_1, q_0) \to q_1 \quad \vee(q_1, q_1) \to q_1$

$\wedge(q_0, q_0) \to q_0 \quad \wedge(q_0, q_1) \to q_0 \quad \wedge(q_1, q_0) \to q_0 \quad \wedge(q_1, q_1) \to q_1$

We take $\neg(\wedge(\vee(1,0), \wedge(1,0)))$:

$$\neg(\wedge(\vee(1,0),\wedge(1,0))) \to^*_{\mathcal{A}_\mathcal{E}} \neg(\wedge(\vee(q_1,q_0),\wedge(q_1,q_0)))$$

$$\to_{\mathcal{A}_\mathcal{E}} \neg(\wedge(q_1,\wedge(q_1,q_0)))$$

$$\to_{\mathcal{A}_\mathcal{E}} \neg(\wedge(q_1,q_0))$$

$$\to_{\mathcal{A}_\mathcal{E}} \neg(q_0)$$

$$\to_{\mathcal{A}_\mathcal{E}} q_1 \qquad \text{accepted by } \mathcal{A}_\mathcal{E}$$

Equationally equivalent trees

Let $\mathcal{A}_{\mathcal{E}}$: ETA $(\mathcal{E}, Q, Q_{\mathsf{fin}}, \Delta)$

$s =_{\mathcal{E}} t$ & s : accepted by $\mathcal{A}_{\mathcal{E}}$ $\Rightarrow$ t : accepted by $\mathcal{A}_{\mathcal{E}}$

Proof

$\exists\, q \in Q_{\mathsf{fin}}$: $s \to_{\mathcal{A}_{\mathcal{E}}} s' \to_{\mathcal{A}_{\mathcal{E}}} \cdots \to_{\mathcal{A}_{\mathcal{E}}} q$ (by assumption)

$$\exists\, l \to r \in \Delta,\ C \in \mathcal{C}_{F \cup Q} : s =_{\mathcal{E}} C[\,l\,] \ \&\ s' =_{\mathcal{E}} C[\,r\,]$$

$$\Downarrow \quad s =_{\mathcal{E}} t \quad \text{(by assumption)}$$

$$\exists\, l \to r \in \Delta,\ C \in \mathcal{C}_{F \cup Q} : t =_{\mathcal{E}} C[\,l\,] \ \&\ s' =_{\mathcal{E}} C[\,r\,]$$

$$t \to_{\mathcal{A}_{\mathcal{E}}} s' \to_{\mathcal{A}_{\mathcal{E}}} \cdots \to_{\mathcal{A}_{\mathcal{E}}} q \qquad \square$$

Remark

- Regardless of $\mathcal{E}$, this property holds, i.e. the class C(ETA$_F$) of tree languages is a sub-class of quotient sets of trees
- Regardless of $\mathcal{E}$, is C(ETA$_F$) closed under Boolean operations ?
  Cf. C(TA$_F$) is closed under Boolean operations

Closure properties of ETA (A-case)

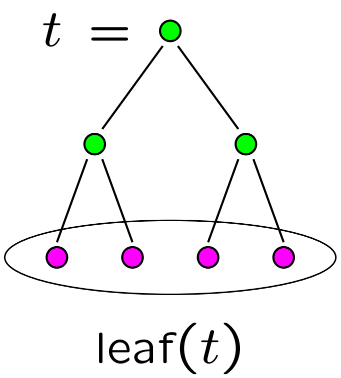
Consider $\mathcal{A}_{\mathcal{E}}$: ETA with $\mathcal{E} = (F, E)$ such that

$$F = \{\, \mathsf{f} \,\} \cup F_0 \qquad (F_0 : \text{ set of constants})$$

$$E = \{\, \mathsf{f}(\mathsf{f}(x, y), z) = \mathsf{f}(x, \mathsf{f}(y, z)) \,\}$$

then $\exists$ CFG $\mathcal{G}$ such that

$$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \{\, t \in \mathcal{T}_F \mid \mathcal{G} \text{ generates leaf}(t) \,\}$$



Corollary

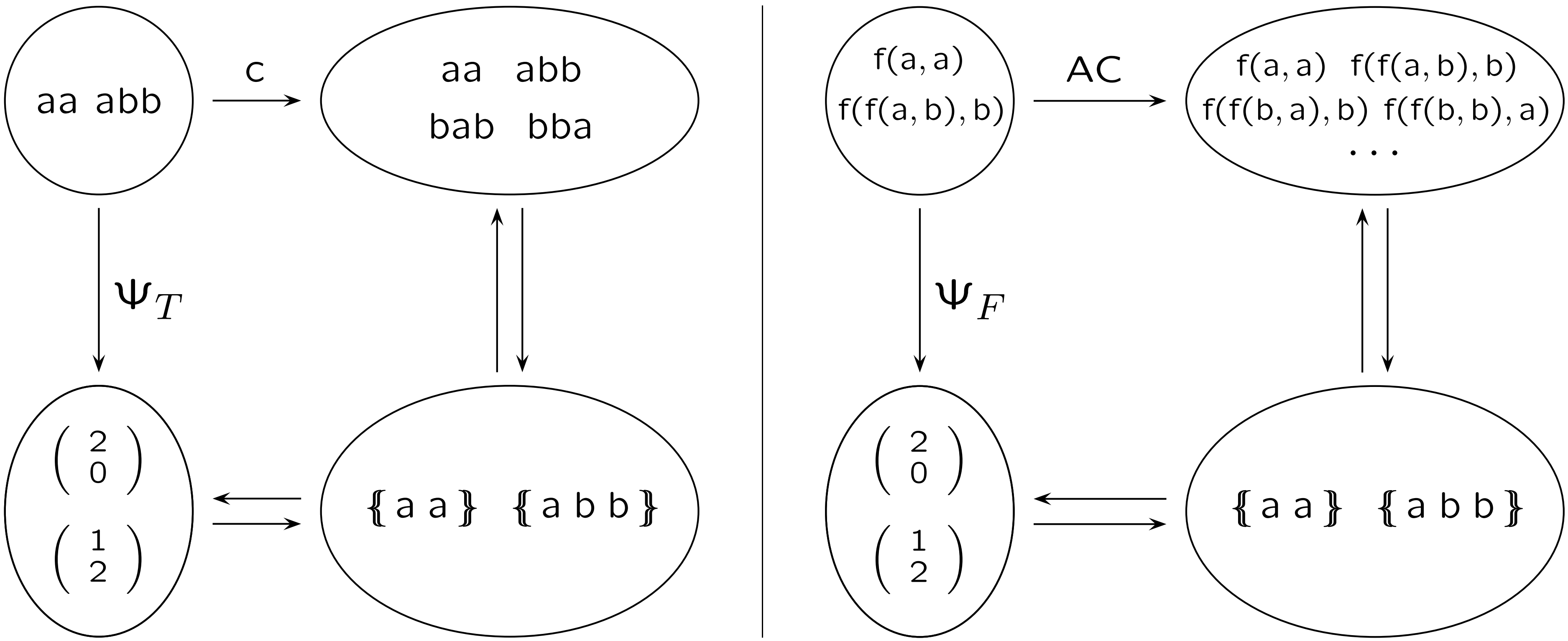
If $\mathcal{E}$ is A-theory, the class of ETA is not closed under intersection or complement

Proof

Suppose, for leading to the contradiction, that this class of ETA over the signature $\{\mathsf{f}\} \cup F_0$ is closed under intersection. Then, there exists an ETA $\mathcal{C}_{\mathcal{E}}$ that accepts $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \cap \mathcal{L}(\mathcal{B}_{\mathcal{E}})$. From the above observation, $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \cap \mathcal{L}(\mathcal{B}_{\mathcal{E}}) = \{\, t \in \mathcal{T}_F \mid \exists \text{ CFG } \mathcal{G}_{\mathcal{C}} : \mathcal{G}_{\mathcal{A}} \cap \mathcal{G}_{\mathcal{B}} \text{ generates leaf}(t) \,\}$. However, since the class of CFG is not closed under intersection, there does not always exists such $\mathcal{G}_{\mathcal{C}}$, that implies that $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \cap \mathcal{L}(\mathcal{B}_{\mathcal{E}})$ is beyond this class. Similar for the proof of the complement. $\square$

Commutative words and AC-trees

Let $T = \{ \mathsf{a}, \mathsf{b} \}$ | Let $F = \{ \mathsf{f}, \mathsf{a}, \mathsf{b} \}$ (f is AC-symbol)



commutative CFL

(Presburger arithmetic)

AC-regular TL

(Presburger arithmetic + MSO)

Closure properties of regular ETA (AC-case)

Consider $\mathcal{A}_{\mathcal{E}}$: regular ETA with $\mathcal{E} = (F, E)$ such that

$$F = \{\,\mathsf{f}\,\} \cup F_0$$

$$E = \{\,\mathsf{f}(\mathsf{f}(x, y), z) = \mathsf{f}(x, \mathsf{f}(y, z)),\ \mathsf{f}(x, y) = \mathsf{f}(y, x)\,\}$$

then $\exists$ NNVAS's $V_1, \ldots, V_n$ such that

$$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \{\, t \in \mathcal{T}_F \mid \sharp_{F_0}(\,\mathsf{leaf}(t)\,) \in [\![V_1]\!] \cup \cdots \cup [\![V_n]\!]\,\}$$

$*$ $\sharp_{F_0}$ is Parikh mapping from the alphabet F_0^* to $\mathbb{N}^{|F_0|}$ (page 2, seminar talk 7)

Corollary

If $\mathcal{E}$ is AC-theory, the class of regular ETA is closed under Boolean operations

Proof for $\cap$

If $F = \{\mathsf{f}\} \cup F_0$, this is an immediate consequence of the Boolean closedness of semi-linear sets (page 6, seminar talk 7). If $F = \{\mathsf{f}\} \cup F'$ where F' may contains non-constant symbols and $E = \{\,\mathsf{f}(\mathsf{f}(x, y), z) = \mathsf{f}(x, \mathsf{f}(y, z)),\ \mathsf{f}(x, y) = \mathsf{f}(y, x)\,\}$, let $\mathcal{A}_{\mathcal{E}}$ and $\mathcal{B}_{\mathcal{E}}$ be ETA such that $\mathcal{A} = (\mathcal{E}, P, P_{\mathsf{fin}}, \Delta_1)$ and $\mathcal{B} = (\mathcal{E}, Q, Q_{\mathsf{fin}}, \Delta_2)$ with the same $\mathcal{E}$. We suppose without loss of generality that $P \cap Q = \varnothing$. (Proof cont'd)

Proof for $\cap$ (cont'd)

Similar to the proof on page 11–12 in seminar talk 3, define $\mathcal{C}_{\mathcal{E}}$ as follows.

$$
\begin{array}{rcl}
P \times Q & : & \{ \langle p, q\rangle \mid p \in P,\ q \in Q \} \\
P_{\mathsf{fin}} \times Q_{\mathsf{fin}} & : & \{ \langle p, q\rangle \mid p \in P_{\mathsf{fin}},\ q \in Q_{\mathsf{fin}} \} \\
\Delta_{F'} & : & \left\{ f(\langle p_1, q_1\rangle, \ldots, \langle p_n, q_n\rangle) \to \langle p, q\rangle \;\middle|\; \begin{array}{l} f(p_1, \ldots, p_n) \to p \in \Delta_1 \\ f(q_1, \ldots, q_n) \to q \in \Delta_2,\ f \in F' \end{array} \right\}
\end{array}
$$

Moreover, for all $p \in P$ and $q \in Q$, define the inequations :

$$
\mathsf{ieq}_1(p, q) \;:\; \langle p, q\rangle + \sum_{\mathsf{f}(p_1,p_2) \to p \in \Delta_1,\ q_1,q_2 \in Q} x_{\langle p_1,q_1\rangle}\, x_{\langle p_2,q_2\rangle} \;\leq\; x_{\langle p,q\rangle}
$$

$$
\mathsf{ieq}_2(p, q) \;:\; \langle p, q\rangle + \sum_{\mathsf{f}(q_1,q_2) \to q \in \Delta_2,\ p_1,p_2 \in P} x_{\langle p_1,q_1\rangle}\, x_{\langle p_2,q_2\rangle} \;\leq\; x_{\langle p,q\rangle}
$$

Let $I_1 = \{ \mathsf{ieq}_1(p, q) \mid p \in P, q \in Q \}$ and $I_2 = \{ \mathsf{ieq}_2(p, q) \mid p \in P, q \in Q \}$. Each of I_1, I_2 contains language inequations with variables $V = \{x_\ell \mid \ell \in P \times Q\}$. In the commutative Kleene algebra $K[V]$, there exists a unique solution for each of the systems I_1 and I_2 of inequations. Let S_i ($i \in \{1, 2\}$) be the solution, which contains equations in the form of $x_{\langle p,q\rangle} = L$. The right-hand side L is a language finitely represented by a commutative context-free grammar. Commutative context-free languages are isomorphic to semi-linear sets (Parikh's theorem), and are effectively closed under intersection (page 6, seminar talk 7). So one can compute the set $S = \{ x_{\langle p,q\rangle} = L \mid x_{\langle p,q\rangle} = L_1 \in S_1,\ x_{\langle p,q\rangle} = L_2 \in S_2\colon L = L_1 \cap L_2 \}$. (Proof cont'd)

Proof for $\cap$ (cont'd)

To each $x_{\langle p,q\rangle} = L$ in S, one can associate a context-free grammar $\mathcal{G}_{\langle p,q\rangle}$ in Chomsky normal form such that $\mathsf{c}(\mathcal{L}(\mathcal{G}_{\langle p,q\rangle})) = L$. One should remark that $\mathcal{G}_{\langle p,q\rangle}$ does not contain a production rule in the form of $\alpha \to \varepsilon$ (as L does not contain ε). For each $p \in P$ and $q \in Q$, let $\Delta_{\langle p,q\rangle}$ be the set of production rules of $\mathcal{G}_{\langle p,q\rangle}$ and let $\alpha_{\langle p,q\rangle}$ be the starting symbol. Define

$$\begin{aligned}\Delta'_{\langle p,q\rangle} &= \{\, \alpha \to \beta\,\gamma \mid \alpha \to \beta\,\gamma \in \Delta_{\langle p,q\rangle} \,:\, \alpha \neq \alpha_{\langle p,q\rangle} \,\} \\ &\cup \{\, \langle p,q\rangle \to \beta\,\gamma \mid \alpha_{\langle p,q\rangle} \to \beta\,\gamma \in \Delta_{\langle p,q\rangle} \,\}.\end{aligned}$$

Observe that $\langle p,q\rangle \to^*_{\Delta'_{\langle p,q\rangle}} w$ and $w \in (P \cup Q)^*$ if and only if $w = \langle p,q\rangle$ or $w \in \mathcal{L}(\mathcal{G}_{\langle p,q\rangle})$.

For each $p \in P$ and $q \in Q$, define

$$T_{\langle p,q\rangle} = \{\, \mathsf{f}(\alpha,\beta) \to \gamma \mid \gamma \to \alpha\,\beta \in \Delta'_{\langle p,q\rangle} \,\}.$$

From the above observation, for every $t \in \mathcal{T}_{\{\mathsf{f}\}\cup(P\times Q)}$ with $|t| \geqslant 3$:

$$t \to^*_{T_{\langle p,q\rangle}/\mathcal{E}} \langle p,q\rangle \text{ if and only if } \pi_1(t) \to^*_{\Delta_1/\mathcal{E}} p \;\&\; \pi_2(t) \to^*_{\Delta_2/\mathcal{E}} q \qquad (*)$$

where π_i ($i \in \{1,2\}$) is the projection $\pi_i(\langle c_1, c_2\rangle) = c_i$; $\pi_i(\mathsf{f}(t_1,t_2)) = \mathsf{f}(\pi_i(t_1), \pi_i(t_2))$. Finally, let $\Delta_{\mathsf{f}} = \bigcup_{p\in P,\, q\in Q} T_{\langle p,q\rangle}$. We take $\Delta_{F'} \cup \Delta_{\mathsf{f}}$ as the set of transition rules of $\mathcal{C}_{\mathcal{E}}$. From the above property $(*)$ together with the argument ($s \to^*_{\mathcal{C}} \langle p,q\rangle$ iff $s \to^*_{\mathcal{A}_1} p$ & $s \to^*_{\mathcal{A}_2} q$) in page 12 in seminar talk 3, $\mathcal{L}(\mathcal{C}_{\mathcal{E}}) = \mathcal{L}(\mathcal{A}_{\mathcal{E}}) \cap \mathcal{L}(\mathcal{B}_{\mathcal{E}})$. $\square$

The closedness under union is obvious. For the proof idea of the complement, consult our paper [Proof of Lemma 5, Ohsaki & Seki & Takai RTA2003].

Example

$$\mathcal{A}_{1\,\mathcal{E}} : \quad \mathsf{a} \to \mathsf{p}, \quad \mathsf{b} \to \mathsf{p}, \quad \mathsf{f}(\mathsf{p},\mathsf{p}) \to \mathsf{p} \quad \Rightarrow \quad L_1 : \quad f(x) = \mathsf{a} + \mathsf{b} + x^2 \leqslant x$$

f is AC symbol

$$f'(x) = x$$

$$f(0) = \mathsf{a} + \mathsf{b}$$

$$L_1 = f'(f(0))^* \cdot f(0) = (\mathsf{a}+\mathsf{b})^* \cdot (\mathsf{a}+\mathsf{b}) = (\mathsf{a}+\mathsf{b})^+$$

$$\mathcal{A}_{2\,\mathcal{E}} : \quad \mathsf{a} \to \mathsf{q}, \quad \mathsf{b} \to \mathsf{q} \quad \Rightarrow \quad L_2 : \quad g(y) = \mathsf{a} + \mathsf{b} \leqslant y$$

$$g'(y) = 1$$

$$g(0) = \mathsf{a} + \mathsf{b}$$

$$L_2 = g'(g(0))^* \cdot g(0) = 1^* \cdot (\mathsf{a}+\mathsf{b}) = \mathsf{a}+\mathsf{b}$$

$$\mathcal{L}(\mathcal{A}_{1\,\mathcal{E}}) \cap \mathcal{L}(\mathcal{A}_{2\,\mathcal{E}}) \quad \Leftarrow \quad \mathsf{c}(L_1) \cap \mathsf{c}(L_2)$$

: effectively computable (pages 6–7, seminar 7)

$$= \mathsf{c}(\{\mathsf{a}\ \mathsf{b}\})$$

$$= \mathcal{E}(\{\mathsf{a}\ \mathsf{b}\})$$

$$= \mathcal{E}(\mathcal{L}(\mathcal{A}))$$

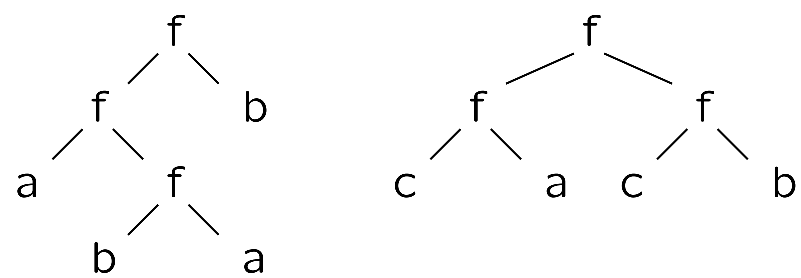
$$= \mathcal{L}(\mathcal{A}_{\mathcal{E}})$$

Tree automata that count

Let $L_=$: tree language that satisfies for all t in $L_=$,

$$|t|_{\mathsf{a}} = |t|_{\mathsf{b}}$$

the numbers of occurrences of "a" and "b" are the same



$L_=$ is accepted by AC-TA (ETA with the AC-theory of f) with the transition rules

$\mathsf{a} \to \mathsf{q_a}$, $\mathsf{b} \to \mathsf{q_b}$, $\mathsf{c} \to \mathsf{q}$, $\mathsf{f}(\mathsf{q_a}, \mathsf{q_b}) \to \mathsf{q}$, $\mathsf{f}(\mathsf{q}, \mathsf{q}) \to \mathsf{q}$ (q : final state)

Note

The class of AC-TA logically subsumes the class of arithmetic in the right table (called Presburger arithmetic). A formula δ in P is satisfiable iff there is an assignment α to free variables in δ such that the closed formula $\delta\alpha$ is true. One can show that a formula δ is satisfiable over $\mathbb{N}$ iff there is an AC-regular tree language L_δ over $F = \{\mathsf{f}\} \cup F_0$ such that f is an AC symbol and L_δ is a model of δ. The previous example $L_=$ is a model of $x_1 = x_2$.

$$\begin{array}{lcl} P & := & C \\ & | & P \vee P \\ & | & \neg(\ P\ ) \\ C & := & \exists x_i (\ P\ ) \\ & | & \sum_{i \in I} a_i x_i = b \\ & & (a_i, b \in \mathbb{Z}) \end{array}$$

Commutation lemma

Let $\mathcal{A}_{\mathcal{E}}$: ETA

$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \{\, t \mid \exists s \in \mathcal{L}(\mathcal{A})\colon s =_{\mathcal{E}} t \,\}$ ($\mathcal{L}(\mathcal{A}_{\mathcal{E}})$ is $=_{\mathcal{E}}$-closure of $\mathcal{L}(\mathcal{A})$)

if $\mathcal{E}$ is linear

Proof

The inclusion "$\supseteq$" is an immediate consequence of the result on page 5. For the reverse inclusion "$\subseteq$", we show that $\to_{\mathcal{A}} \cdot \leftrightarrow_{\mathcal{E}} \subseteq \leftrightarrow_{\mathcal{E}} \cdot \to_{\mathcal{A}}^{=}$. ($\to_{\mathcal{A}}^{=}$ is the smallest reflexive relation containing $\to_{\mathcal{A}}$). Let $\mathcal{A} = (\mathcal{E}, Q, Q_{\mathsf{fin}}, \Delta)$ and $\mathcal{E} = (F, E)$. Suppose $s \to_{\mathcal{A}} t$ and $t \leftrightarrow_{\mathcal{E}} u$. Then there exists a transition rule $l \to r \in \Delta$ such that $s = C[l]$ and $t = C[r]$, and there exists an equation $l' = r' \in E$ such that (1) $t = D[l'\sigma]$ and $u = D[r'\sigma]$, or (2) $t = D[r'\sigma]$ and $u = D[l'\sigma]$. We consider (1) only below, but the same argument can be applied to (2). Since r is a state symbol and l' does not contain any state symbol, if $l' = r'$ is applied above the position of r (at an ancestor position), σ contains a mapping $x \mapsto D'[r]$ such that x is a variable in l'. In this case, let $\sigma' = (\sigma - \{x \mapsto D'[r]\}) \cup \{x \mapsto D'[l]\}$, then $s = D[l'\sigma']$. Because $l' = r'$ is a linear equation, $D[l'\sigma'] \to_{\mathcal{E}} D[r'\sigma']$ and $D[r'\sigma'] \to_{\mathcal{A}}^{=} u$. (If x does not appear in r', $D[r'\sigma'] = u$). If $l' = r'$ is applied at a non-ancestor position of r, obviously the commutation holds. Hence, the commutation of $\to_{\mathcal{A}}$ over $=_{\mathcal{E}}$ holds. $\square$

Corollary

The class of AC-TA is closed under Boolean operations

Proof

Let $\mathcal{A}_{\mathcal{E}}$ be a (possibly non-regular) AC-TA. According to Commutation lemma, $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \{ t \mid \exists s \in \mathcal{L}(\mathcal{A}) \colon s =_{\mathcal{E}} t \}$. Since it is possible to construct DTA $\mathcal{B}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$, one can obtain a regular ETA $\mathcal{B}_{\mathcal{E}}$ such that $\mathcal{L}(\mathcal{B}_{\mathcal{E}}) = \mathcal{L}(\mathcal{A}_{\mathcal{E}})$. Because the class of regular AC-TA is closed under Boolean operations (page 7), so is the class of AC-TA. □

	AC-TA	A-TA
closed under $\cup$	✓	✓
closed under $\cap$	✓	−
closed under $(\,)^c$	✓	−

✓ : closed

− : not closed

Summary of closure properties

Discussion

Any super-/sub-class of A-TA (AC-TA) closed under Boolean operations ?

Decidability (AC-case)

The following problems are decidable for the class of AC-TA :

$t \in \mathcal{L}(\mathcal{A}_{\mathcal{E}})$?	(membership problem)
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \varnothing$?	(emptiness problem)
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{T}_F$?	(universality problem)
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \mathcal{L}(\mathcal{B}_{\mathcal{E}})$?	(inclusion problem)
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{L}(\mathcal{B}_{\mathcal{E}})$?	(equivalence problem)

Proof

Similar to the proof of the decidability of TA, it suffices to show that the emptiness problem is decidable, because thanks to closedness under Boolean operations, the membership, universality, inclusion, equivalence problems are spacial cases of the emptiness problem. Let $\mathcal{A}_{\mathcal{E}}$ be an AC-TA. ($\mathcal{E}$ is the AC-theory for some of the binary symbols.) According to the previous Commutation lemma, $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \{ t \mid \exists s \in \mathcal{L}(\mathcal{A})\colon s =_{\mathcal{E}} t \}$. This implies that $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \varnothing$ if and only if $\mathcal{L}(\mathcal{A}) = \varnothing$. Since the emptiness problem is decidable for TA, so is for AC-TA. □

The complexity of the emptiness problem of AC-TA coincides with TA (linear time). The membership problem is NP-complete. Other problems are at least $2^{2^{cn}}$ $(c > 0)$.

Decidability (A-case)

The following two problems are decidable for the class of A-TA :

$t \in \mathcal{L}(\mathcal{A}_{\mathcal{E}})$? (membership problem)

$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \emptyset$? (emptiness problem)

However, the other three problems are undecidable :

$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{T}_F$? (universality problem)

$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \mathcal{L}(\mathcal{B}_{\mathcal{E}})$? (inclusion problem)

$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{L}(\mathcal{B}_{\mathcal{E}})$? (equivalence problem)

Proof

The decidability of membership problem follows from the following observation : Given a tree t and an A-TA $\mathcal{A}_{\mathcal{E}}$ over the signature F, every reachable tree s from t satisfies $|s| \leqslant |t|$. Since the number of trees that satisfies this condition is finite, it can be determined if t is reachable to some final state.

The decidability of emptiness problem is an immediate consequence of Commutation lemma (which implies that $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \varnothing$ if and only if $\mathcal{L}(\mathcal{A}) = \varnothing$ as the associativity axiom is a linear equation).

(Proof cont'd)

Proof (cont'd)

For undecidability of the universality, inclusion, equivalence problems, we use the reduction from the same problems of context-free grammar : Given two context-free grammar $\mathcal{G}_1$ and $\mathcal{G}_2$ over the alphabet Σ, define the A-TA $\mathcal{A}_\mathcal{E}$ and $\mathcal{B}_\mathcal{E}$ associated to $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Without loss of generality, suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are in Chomsky normal form, and $\mathcal{L}(\mathcal{G}_i)$ $(i \in \{1, 2\})$ does not contain the empty word ε. Then, every production rule of $\mathcal{G}_i$ $(i \in \{1, 2\})$ is in the form of $\alpha \to \beta\gamma$ or $\alpha \to a$ $(\alpha, \beta, \gamma \in N_i$, $a \in T$, where N_i is the set of non-terminals of $\mathcal{G}_i$ and T is the set of terminals). Let F be the signature which contains f (binary symbol) and constant symbols from T, and let E be the set of associativity axiom of f. Define $\mathcal{A}_1 = (\mathcal{E}, Q_1, Q_{1\mathsf{fin}}, \Delta_1)$ and $\mathcal{A}_2 = (\mathcal{E}, Q_2, Q_{2\mathsf{fin}}, \Delta_2)$ as follows:

$$\begin{array}{rcl} Q_i & : & N_i \\ Q_{i\mathsf{fin}} & : & \{\, \mathsf{q}_i \,\} \text{ where } \mathsf{q}_i \text{ is a starting symbol of } \mathcal{G}_i \\ \Delta_i & : & \{\, \mathsf{f}(\alpha, \beta) \to \gamma \mid \gamma \to \alpha\,\beta \in \Delta_i \,\} \cup \{\, a \to \alpha \mid \alpha \to a \in \Delta_i \,\} \end{array}$$

Then the rest of the proof is straightforward, because for every word $w \in T^*$ and tree $t \in \mathcal{T}_F$ such that $\mathsf{leaf}(t) = w$, $w \in \mathcal{L}(\mathcal{G}_i)$ if and only if $t \in \mathcal{L}(\mathcal{A}_{i\mathcal{E}})$. Moreover, for every word $w \in T^* - \{\varepsilon\}$, there exists a tree $t \in \mathcal{T}_F$ such that $\mathsf{leaf}(t) = w$. (This is the proof of the reverse of the statement on page 6.) So, $\mathcal{L}(\mathcal{G}_1) \cup \{\varepsilon\} = T^*$ if and only if $\mathcal{L}(\mathcal{A}_{1\mathcal{E}}) = \mathcal{T}_F$ (universality); $\mathcal{L}(\mathcal{G}_1) \subseteq \mathcal{L}(\mathcal{G}_2)$ if and only if $\mathcal{L}(\mathcal{A}_{1\mathcal{E}}) \subseteq \mathcal{L}(\mathcal{A}_{2\mathcal{E}})$ (inclusion); $\mathcal{L}(\mathcal{G}_1) = \mathcal{L}(\mathcal{G}_2)$ if and only if $\mathcal{L}(\mathcal{A}_{1\mathcal{E}}) = \mathcal{L}(\mathcal{A}_{2\mathcal{E}})$ (equivalence). Since these problems are undecidable for context-free grammar, so are for A-TA. $\square$

Related work

- Multitree automata
  (first appeared in 2003)

  D. Lugiez : *Multitree automata that count*, TCS 333, pp. 225–263, 2005

- Alternating two-way AC-tree automata
  (first appeared in 2003)

  K.N. Verma & J. Goubault-Larrecq : *Alternating two-way AC-tree automata*, Information and Computation 205, pp. 817–869, 2007

- Presburger tree automata

  H. Seidl & T. Schwentick & A. Muscholl : *Numerical document queries*, 22nd PODS, pp. 155–166, ACM, 2003

- Propositional tree automata $*$ strictly more powerful than ETA

  J. Hendrix & H. Ohsaki & M. Viswanathan : *Propositional tree automata*, 17th RTA, LNCS 4098, pp. 50–65, 2006

References

- *Beyond Regularity: Equational Tree Automata for Associative and Commutative Theories* (Ohsaki)

  15th CSL, LNCS 2142, pp. 539–553, 2001

- *Decidability and Closure Properties of Equational Tree Languages* (Ohsaki & Takai)

  13th RTA, LNCS 2378, pp. 114–128, 2002

- *Recognizing Boolean Closed A-Tree Languages with Membership Conditional Rewriting Mechanism* (Ohsaki & Seki & Takai)

  14th RTA, LNCS 2706, pp. 483–498, 2003

- *ACTAS : A System Design for Associative and Commutative Tree Automata Theory* (Ohsaki & Takai)

  5th RULE (2004), ENTCS 124, pp. 97–111, 2005

  Tool will be available at : `http://staff.aist.go.jp/hitoshi.ohsaki/actas/`

Exercise

1. Show that the AC-TA on page 13 accepts $L_=$.
2. Show that the AC-closure of a regular tree language is not always a regular tree language. The AC-closure of L is $\{t \mid \exists t \in L\colon s =_{\mathcal{E}} t\}$ where $\mathcal{E}$ is the AC-theory. (Cf. Exercise 6,7, seminar talk 3)
3. Show that $L_=$ is accepted by A-TA, i.e. let $\mathcal{E}$ be the A-theory of f with $F = \{\mathsf{f}, \mathsf{a}, \mathsf{b}, \mathsf{c}\}$ and let $L_=$ be the set of trees over F satisfying $|t|_{\mathsf{a}} = |t|_{\mathsf{b}}$, show that $L_=$ is accepted by A-TA. (Cf. Example on page 5, seminar talk 2)
4. If $F = \{\mathsf{f}, \mathsf{a}\}$ with ar(a) = 0 and ar(f) = 2, the class C(AC-TA_F) of AC-tree automata and the class C(TA_F) of tree automata coincides.
5. Construct AC-TA $\mathcal{A}_{\mathcal{E}}$, $\mathcal{B}_{\mathcal{E}}$, $\mathcal{C}_{\mathcal{E}}$ over $F = \{\,\mathsf{f}\,\} \cup F_0$, each Parikh image of whose leaf languages is a model of (1) $\exists\, x_1, x_2\, (x_1 = 1 \vee x_2 = 2)$, (2) $\forall\, x_1, x_2, x_3\, (x_1 + x_2 = x_3)$, (3) $\forall\, x_1, x_2\, (x_1 \geqslant 2\, x_2)$.
6. Explain by showing an example why Commutation lemma (page 14) does not hold for a non-linear equational theory.

National Institute of Advanced Industrial Science and Technology (AIST) – Senri-site, AIST Kansai.

Office: Shin-Senri Nishi 1–2–14 (MSK bldg. 5th floor), Toyonaka, Osaka 560–0083, Japan

URL: **http://staff.aist.go.jp/hitoshi.ohsaki/**