Introduction to Tree Language Theory

Hitoshi Ohsaki

National Institute of
Advanced Industrial Science and Technology (AIST)

seminar talk (10/10)

2009

X. MTA and further extensions...

Summary of the talks

Talk-9 : Monotone tree automata

1. Monotone tree automata and accepted languages
2. First order theory and the satisfiability
3. Decidable Diophantine arithmetic
4. Open questions

Talk-10 : Monotone A-tree automata & further extensions

5. Tree language hierarchy
6. Closure properties and decidability
7. About ETA families : PTA & TAN

Monotone A-tree automata

ETA $(\mathcal{E}, Q, Q_{\text{fin}}, \Delta)$

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

F : signature

E : finite set of associativity equations

e.g. $(x + y) + z \approx x + (y + z)$

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

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

Δ : finite set of transition rules with the following forms

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

$p \to q$ [epsilon rule]

$f(p_1, \ldots, p_n) \to f(q_1, \ldots, q_n)$ [monotone rule]

Question $\mathcal{C}(\text{MA-TA}_F) \stackrel{?}{=} \mathcal{C}(\text{RA-TA}_F)$

‘Monotone’ A-TA vs. ‘Regular’ A-TA

Let $\mathcal{E} = (F, E)$

E : $(x+y)+z \approx x+(y+z)$

F : $+$ (binary symbol)

a, b, c $\cdots$ (constant symbols)

Theorem

For every A-TA $\mathcal{A}_{\mathcal{E}}$

if it is monotone, the leaf language is context-sensitive

if it is regular, the leaf language is context-free

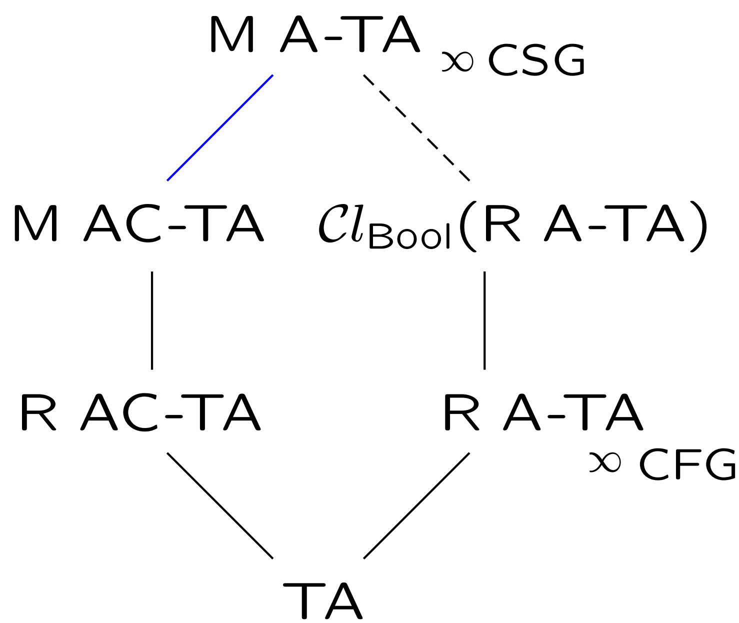
Proof

Due to the syntax, if $\mathcal{A}_{\mathcal{E}}$ is monotone, it has transition rules in the forms of $p_1 + p_2 \to q_1 + q_2$, $p_1 + p_2 \to q$ and $a \to q$ $(a \in F)$. Looking at the rules from right to left, they correspond to rules of a context-sensitive grammar (Kuroda normal form, page 16 of seminar talk 2). Likewise, if $\mathcal{A}_{\mathcal{E}}$ is regular, the rules of $\mathcal{A}_{\mathcal{E}}$ correspond to rules of a context-free grammar in Chomsky normal form. □

A vs. AC

Given a monotone AC-TA $\mathcal{A}_{\mathcal{E}_1}$, one can construct a monotone A-TA $\mathcal{B}_{\mathcal{E}_2}$ such that

$$\mathcal{L}(\mathcal{A}_{\mathcal{E}_1}) = \mathcal{L}(\mathcal{B}_{\mathcal{E}_2})$$



Proof

Let $\mathcal{A} = (\mathcal{E}_1, Q, Q_{\mathrm{fin}}, \Delta_1)$ and $\mathcal{E}_1 = (F, E_1)$. Then define $\mathcal{B} = (\mathcal{E}_2, Q, Q_{\mathrm{fin}}, \Delta_2)$ and $\mathcal{E}_2 = (F, E_2)$ where

$$E_2 \;:\; E_1 - \{\, f(x,y) \approx f(y,x) \mid f \in F \,\}$$

$$\Delta_2 \;:\; \Delta_1 \cup \{\, f(p,q) \to f(q,p) \mid p, q \in Q \,\}$$

It is straightforward to show that $\mathcal{L}(\mathcal{B}_{\mathcal{E}_2}) \subseteq \mathcal{L}(\mathcal{A}_{\mathcal{E}_1})$. So it suffices to show the reverse inclusion $\mathcal{L}(\mathcal{A}_{\mathcal{E}_1}) \subseteq \mathcal{L}(\mathcal{B}_{\mathcal{E}_2})$. The property is obtained by the following simulation relations:

1. $=_{\mathcal{E}_1} \subseteq \to^*_{\Delta_2/\mathcal{E}_2}$
2. $\to_{\Delta_1} \subseteq \to_{\Delta_2}$

(Complete the proof in Exercise) $\square$

Closure properties ($\cup, \cap$)

The class of monotone A-TA is closed under union and intersection

Proof

It is obvious that the class is closed under union. A similar proof is found in page 9 of seminar talk 3. For the proof of the closure under intersection, one can apply the tree automata construction described in page 20 of seminar talk 9: Let $\mathcal{A}_1 = (\mathcal{E}, P, P_{\text{fin}}, \Delta_1)$ and $\mathcal{A}_2 = (\mathcal{E}, Q, Q_{\text{fin}}, \Delta_2)$ with $\mathcal{E} = (F, E)$. Suppose F_{A} is a set of all A symbols in F. Then define $\Delta_{\mathsf{A}} = \bigcup_{\mathsf{f} \in F_{\mathsf{A}}} \Delta_{\mathsf{f}}$ (instead of Δ_{AC}),

$$
\begin{array}{lll}
\Delta_{\mathsf{f}}: & \mathsf{f}(\langle p_1, q_1\rangle, \langle p_2, q_2\rangle) \to \mathsf{f}(\langle p, q_1\rangle, \langle \circ, q_2\rangle) & \text{if } \exists \mathsf{f}(p_1, p_2) \to p \in \Delta_1, q_1, q_2 \in Q \\
& \mathsf{f}(\langle p_1, q_1\rangle, \langle p_2, q_2\rangle) \to \mathsf{f}(\langle p_3, q_1\rangle), \langle p_4, q_2\rangle) & \text{if } \exists \mathsf{f}(p_1, p_2) \to \mathsf{f}(p_3, p_4) \in \Delta_1, q_1, q_2 \in Q \\
& \mathsf{f}(\langle \circ, q_1\rangle, \langle p_2, q_2\rangle) \to \langle p_2, q\rangle & \text{if } \exists \mathsf{f}(q_1, q_2) \to q \in \Delta_2, p_2 \in P \\
& \mathsf{f}(\langle \circ, q_1\rangle, \langle p_2, q_2\rangle) \to \mathsf{f}(\langle \circ, q_3\rangle, \langle p_2, q_4\rangle) & \text{if } \exists \mathsf{f}(q_1, q_2) \to \mathsf{f}(q_3, q_4) \in \Delta_2, p_2 \in P \\
& \langle p_1, q_1\rangle \to \langle p_2, q_1\rangle & \text{if } \exists p_1 \to p_2 \in \Delta_1, q_1 \in Q \\
& \langle p_1, q_1\rangle \to \langle p_1, q_2\rangle & \text{if } \exists q_1 \to q_2 \in \Delta_2, p_1 \in P \\
& \mathsf{f}(\langle p_1, q_1\rangle, \langle \circ, q_2\rangle) \to \mathsf{f}(\langle \circ, q_1\rangle, \langle p_1, q_2\rangle) & \text{if } \exists p_1 \in P, q_1, q_2 \in Q \\
& \mathsf{f}(\langle \circ, q_1\rangle, \langle p_1, q_2\rangle) \to \mathsf{f}(\langle p_1, q_1\rangle, \langle \circ, q_2\rangle) & \text{if } \exists p_1 \in P, q_1, q_2 \in Q.
\end{array}
$$

□

Complementation

The class of monotone A-TA is closed under complement

Proof

We show the proof for the case that $F = \{\, \mathsf{f}, \mathsf{g} \,\} \cup F_0$, where f is A symbol and g is binary symbol. The proof can be generalized to a signature containing arbitrary many n-ary symbols ($n > 0$) and A symbols: Let $\mathcal{E} = (F, E)$ with $E = \{\, \mathsf{f}(\mathsf{f}(x, y), z) \approx \mathsf{f}(x, \mathsf{f}(y, z)) \,\}$. Given $\mathcal{A} = (\mathcal{E}, Q, Q_{\mathsf{fin}}, \Delta)$, define $\mathcal{B} = (\mathcal{E}, Q', Q'_{\mathsf{fin}}, \Delta')$ as follows.

Q' : 2^Q

Q'_{fin} : $\{\, P \subseteq Q \mid P \cap Q_{\mathsf{fin}} = \varnothing \,\}$

Δ' : $\{ a \to \{p \mid a \to p \in \Delta\} \mid a \in F_0 \,\} \cup \Delta_1 \cup \Delta_2 \cup \Delta_3$

where

Δ_1 : $\mathsf{g}(P_1, P_2) \to P$

if $P_1, P_2 \subseteq Q$ and

$P = \{\, p \mid \mathsf{g}(p_1, p_2) \to p \in \Delta$ and $p_i \in P_i$ $(1 \leqslant i \leqslant 2) \,\}$

(Proof cont'd)

H. Ohsaki, H. Seki, T. Takai: *Recognizing Boolean Closed A-Tree Languages with Membership Conditional Rewriting Mechanism*, Proc. of 14th RTA, LNCS 2706, pp.483–498, Springer, 2003

Proof (cont'd)

$$\Delta_2 \;:\; \mathsf{f}(\alpha,\beta) \to \gamma \qquad \text{if } \gamma \to \alpha\,\beta \in \mathcal{G}_P$$

$$\mathsf{f}(\alpha,\beta) \to \mathsf{f}(\gamma,\delta) \qquad \text{if } \gamma\,\delta \to \alpha\,\beta \in \mathcal{G}_P$$

for all $(\varnothing \neq)\ P \subseteq Q$ where

$\mathcal{G}_P$: CSG such that

$$\mathcal{L}(\mathcal{G}_P) = \bigcap_{p \in P} \mathcal{L}(\Phi(\mathcal{G}_q)) \;-\; \bigcup_{p \in (Q-P)} \mathcal{L}(\Phi(\mathcal{G}_q))$$

$\mathcal{G}_q$: CSG whose start symbol is q and production rules are

$$p_3 \to p_1\,p_2 \qquad \text{if } \mathsf{f}(p_1,p_2) \to p_3 \in \Delta$$

$$p_3, p_4 \to p_1\,p_2 \qquad \text{if } \mathsf{f}(p_1,p_2) \to \mathsf{f}(p_3,p_4) \in \Delta$$

Φ : mapping onto CSG that adds production rules :

$$p \to S \qquad \text{if } p \in S,\ S \subseteq Q$$

$$\Delta_3 \;:\; \mathsf{f}(\alpha,\beta) \to \gamma \qquad \text{if } \gamma \to \alpha\,\beta \in \mathcal{G}_0$$

$$\mathsf{f}(\alpha,\beta) \to \mathsf{f}(\gamma,\delta) \qquad \text{if } \gamma\,\delta \to \alpha\,\beta \in \mathcal{G}_0$$

where

$\mathcal{G}_0$: CSG such that

$$\mathcal{L}(\mathcal{G}_0) = \{\, w \in (2^Q)^* \mid |w| \geqslant 2 \,\} \;-\; \bigcup_{p \in Q} \mathcal{L}(\Phi(\mathcal{G}_q))$$

Using the above automaton, one can prove that $\mathcal{L}(\mathcal{B}_\mathcal{E}) = (\,\mathcal{L}(\mathcal{A}_\mathcal{E})\,)^{\mathsf{c}}$. $\square$

PSPACE problems

The membership problem of CSG :

instance is grammar $\mathcal{G} = (\Sigma, T, N, \mathsf{q}_0, \Delta)$, word w

answer is "yes" if $w \in \mathcal{L}(\mathcal{G})$; "no" otherwise

This is PSPACE problem

Proof

This problem is $\leqslant^P_{\mathrm{m}}$-reducible to the membership problem of LBA (linear bounded automata). The LBA membership problem is in NSPACE(n) (Appendix in seminar talk 2). From Savitch's theorem (page 17 in seminar talk 6), NPSPACE=PSPACE, and thus the CSG membership problem is in PSPACE. □

Note

The membership problem of LBA is PSPACE-complete (Show that every PSPACE problem is $\leqslant^P_{\mathrm{m}}$-reducible to this problem **Exercise**).

Moreover, the (reverse) transformation from the LBA membership problem to the CSG membership problem is also $\leqslant^P_{\mathrm{m}}$-reducible. Therefore, the membership problem of CSG is PSPACE-complete.

Membership problem of monotone A-TA

The membership problem of monotone A-TA is PSPACE-complete

Proof

PSPACE-hardness is an immediate consequence of the previous observation about the membership problem of CSG, because CSG can be simulated by monotone A-TA. On the other hand, since polynomially bounded recursion of PSPACE problem (P^{PSPACE}) is in PSPACE, so is the membership problem of monotone A-TA. □

Note 1

The membership problem of regular A-TA is in P.

Note 2

The emptiness problem of monotone A-TA is undecidable.
($\because$ The emptiness problem of CSG is so.)

The universality and inclusion problems of monotone A-TA are also undecidable.
($\because$ $\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \varnothing \Leftrightarrow (\mathcal{L}(\mathcal{A}_{\mathcal{E}}))^{c} = \mathcal{T}_F \Leftrightarrow \mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \varnothing$)

Membership problem of monotone AC-TA

The membership problem of monotone AC-TA is PSPACE-complete

Proof

Since monotone AC-TA can be simulated by monotone A-TA (page 5) and the transformation can be done in polynomial time relative to the size of the input, the problem is in PSPACE. For PSPACE-hardness, one can use the reduction from the "quantified Boolean formula" problem (QBF) :

instance is a first-order logical formula ψ in the syntax S

$$S \ ::= \ x \ | \ \neg S \ | \ S \wedge S \ | \ \exists x\,(\,S\,)$$

answer is "yes" if ψ is true ; "no" otherwise

This problem is PSPACE-complete. So it suffices to show that for an arbitrary formula ψ in S, one can construct a monotone AC-TA $\mathcal{A}_\mathcal{E}$ and a tree t such that ψ is true iff $\mathcal{A}_\mathcal{E}$ accepts t. An example of such construction can be found in, e.g., our paper [Ohsaki & Talbot & Tison & Roos].

H. Ohsaki, J.-M. Talbot, S. Tison, Y. Roos: *Monotone AC-Tree Automata*, Proc. of 12th LPAR, LNAI 3835, pp.337–351, Springer, 2005

Summary of decidability and closure properties

	M A-TA	M AC-TA	R A-TA	R AC-TA
$t \in \mathcal{L}(\mathcal{A}_{\mathcal{E}})$?	✓	✓	✓	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \emptyset$?	–	✓	✓	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{T}_F$?	–	?	–	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \mathcal{L}(\mathcal{B}_{\mathcal{E}})$?	–	–	–	✓

(✓ : positive, – : negative, ? : unknown)

	M A-TA	M AC-TA	R A-TA	R AC-TA
closed under $\cup$	✓	✓	✓	✓
closed under $\cap$	✓	✓	–	✓
closed under $()^c$	✓	–	–	✓

Propositional tree automata (PTA)

PTA $\mathcal{A} = (\mathcal{E}, Q, \psi, \Delta)$

ψ : propositional formula over Q

t : accepted by $\mathcal{A}_\mathcal{E}$ if $\{ q \in Q \mid t \to_{\mathcal{A}^*_\mathcal{E}} q \} \models \psi$

where $P \models q$ if $q \in P$

$P \models \psi_1 \wedge \psi_2$ if $P \models \psi_1$ & $P \models \psi_2$

$P \models \neg\psi$ if $P \not\models \psi$

Note

The class of PTA is closed under Boolean operations. (Exercise)

The membership problem of PTA is decidable if $\{ q \in Q \mid t \to_{\mathcal{A}^*_\mathcal{E}} q \}$ is computable. (In particular, if $\mathcal{E}$ is AC theory, the membership problem is in Δ_2^P.)

J. Hendrix, H. Ohsaki, M. Viswanathan: *Propositional Tree Automata*, Proc. of 17th RTA, LNCS 4098, pp.50–65, Springer, 2006

Example

Consider (regular) PTA $\mathcal{A}$ with

$$\Delta \;:\; \mathsf{a} \to q_{\mathsf{a}} \quad \mathsf{b} \to q_{\mathsf{b}} \quad \mathsf{f}(q_{\mathsf{a}}, r) \to q_{\mathsf{a}} \quad \mathsf{f}(r, r) \to r \quad \mathsf{f}(r, q_{\mathsf{b}}) \to q_{\mathsf{b}}$$

$$\mathsf{a} \to r \quad \mathsf{b} \to r \quad \mathsf{c} \to r$$

$$\psi \;:\; q_{\mathsf{a}} \wedge \neg q_{\mathsf{b}}$$

$\mathcal{A}$ accepts trees whose leftmost leaf is 'a' and rightmost is not 'b'

Note

	M A-PTA	M AC-PTA	R A-PTA	R AC-PTA
$t \in \mathcal{L}(\mathcal{A}_{\mathcal{E}})$?	✓	✓	✓	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \emptyset$?	−	−	−	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \mathcal{T}_F$?	−	−	−	✓
$\mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \mathcal{L}(\mathcal{B}_{\mathcal{E}})$?	−	−	−	✓

(✓ : positive, − : negative)

Tree automata with normalization (TAN)

TAN $\mathcal{A} = (\mathcal{R}, Q, Q_{\mathsf{fin}}, \Delta)$

$\mathcal{R}$: equational rewrite system $(\mathcal{E}, R)$

t : accepted by $\mathcal{A}_{\mathcal{R}}$ if $\exists u \colon t \to^{!}_{\mathcal{R}} u$ & $u \to^{*}_{\mathcal{A}_{\mathcal{E}}} q$ $(q \in Q_{\mathsf{fin}})$

Note

The class of TAN is closed under Boolean operations if $\mathcal{R}$ is confluent & terminating.

The membership problem of TAN is decidable if $\mathcal{R}$ is terminating & the membership problem of $\mathcal{A}_{\mathcal{E}}$ is decidable. (relative decidability)

Question

Let $R_I = \{ x + x \to x \}$ and $E_{AC} = \{ (x+y)+z \approx x+(y+z) \quad x+y \approx y+x \}$
Is it decidable for TAN $\mathcal{A}_{\mathcal{R}}$ with R_I and E_{AC}, whether $\mathcal{L}(\mathcal{A}_{\mathcal{R}}) = \varnothing$?

H. Ohsaki, H. Seki: *Languages Modulo Normalization*, Proc. of 6th FroCoS, LNCS 4720, pp.221–236, Springer, 2007

Idempotency $X + X \approx X$

Let $E_{ACI} = E_{AC} \cup \{ s(x) + s(x) \approx s(x) \}$

Consider tree language L : $s^i(0) + \cdots + s^i(0) \quad (i \in \mathbb{N})$

- L is accepted by ETA with $\mathcal{E} = (F, E_{ACI})$
- $(L)^{\mathsf{c}}$ is not accepted by ETA with $\mathcal{E} = (F, E_{ACI})$

  (∵ Suppose $\mathcal{A}_{\mathcal{E}}$ accepts $(L)^{\mathsf{c}}$. From Pumping Lemma, there exist m, n $(m \neq n)$ such that $s^m(0) \to^*_{\mathcal{A}_{\mathcal{E}}} q$ and $s^n(0) \to^*_{\mathcal{A}_{\mathcal{E}}} q$ $(q \in Q)$. Since $\mathcal{A}_{\mathcal{E}}$ accepts $s^m(0) + s^n(0)$, $\mathcal{A}_{\mathcal{E}}$ accepts also $s^m(0) + s^m(0)$, leading to the contradiction.)

Let $\mathcal{R}_{ACI} = (F, E_{AC}, \{ s(x) + s(x) \to s(x) \})$

- $(L)^{\mathsf{c}}$ is accepted by TAN with $\mathcal{R}_{ACI}$:

  $0 \to q \quad s(q) \to q \quad q + q \to q_{\mathsf{f}} \quad s(q_{\mathsf{f}}) \to q \quad q + q_{\mathsf{f}} \to q_{\mathsf{f}}$ (q_{f}: final state)

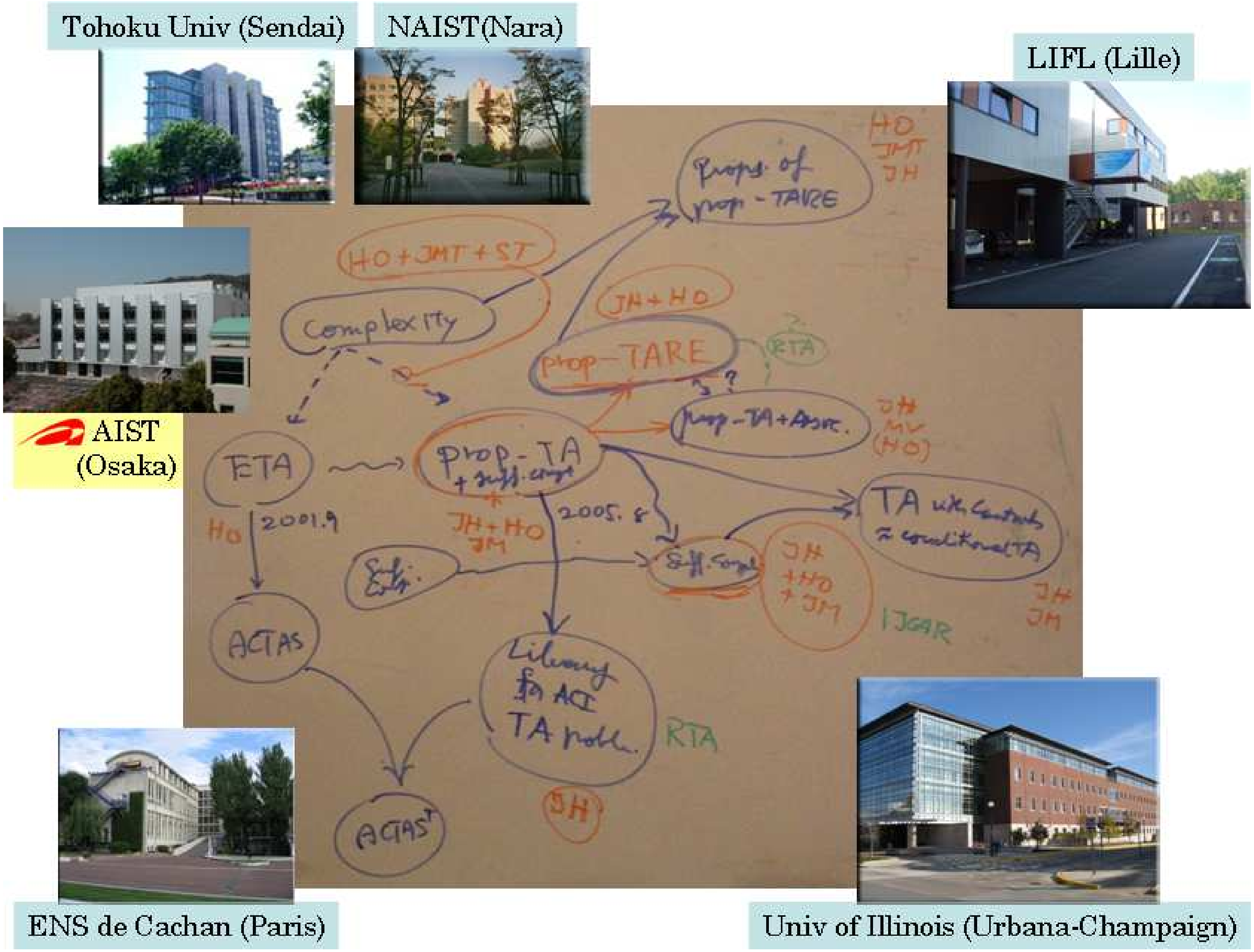
  If eliminate the last two rules, the TAN accepts :

  $$s^{m_1}(0) + \cdots + s^{m_k}(0) \qquad (m_i \neq m_j \text{ for some } i, j)$$

Exercise

1. Show that $=_{\mathcal{E}_1} \subseteq \to^*_{\Delta_2/\mathcal{E}_2}$ in page 5.
2. Show that every PSPACE problem is $\leqslant^P_{\mathrm{m}}$-reducible to the membership problem of LBA (page 9). (Hint : For a problem with input of the size n and with worktape of the length bounded by a polynomial $P(n)$, consider how to simulate the computation of the problem by LBA.)
3. Show that the class of monotone A-TA subsumes the class of the Boolean closure $\mathcal{C}l_{\mathrm{Bool}}$(M AC-TA) of monotone AC-TA.
4. Show that the class of PTA is closed under Boolean operations (both in regular and monotone cases).
5. Show that (1) the emptiness problem of (regular) AC-PTA is decidable, and (2) the emptiness problem of monotone AC-PTA is undecidable.
6. Show that the class of TAN with ETRS $\mathcal{R}$ is closed under Boolean operations if $\mathcal{R}$ is terminating and confluent.
7. Show that $\mathcal{R} = (F, E_{AC}, R_I)$ in page 15 is terminating and confluent.

Appendix : Overview of project ETA (η)



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/**