

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 $F \cap Q = \emptyset$

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

Δ : finite set of transition rules with the following forms

$f(p_1, \dots, p_n) \rightarrow q$ [regular rule]

$p \rightarrow q$ [epsilon rule]

$f(p_1, \dots, p_n) \rightarrow f(q_1, \dots, q_n)$ [**monotone rule**]

Question

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

'Monotone' A-TA vs. 'Regular' A-TA

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

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

$F : +$ (binary symbol)

$a, b, c \dots$ (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 $p_1 + p_2 \rightarrow q_1 + q_2$, $p_1 + p_2 \rightarrow q$ and $a \rightarrow q$ ($a \in F$). 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. □ 4

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

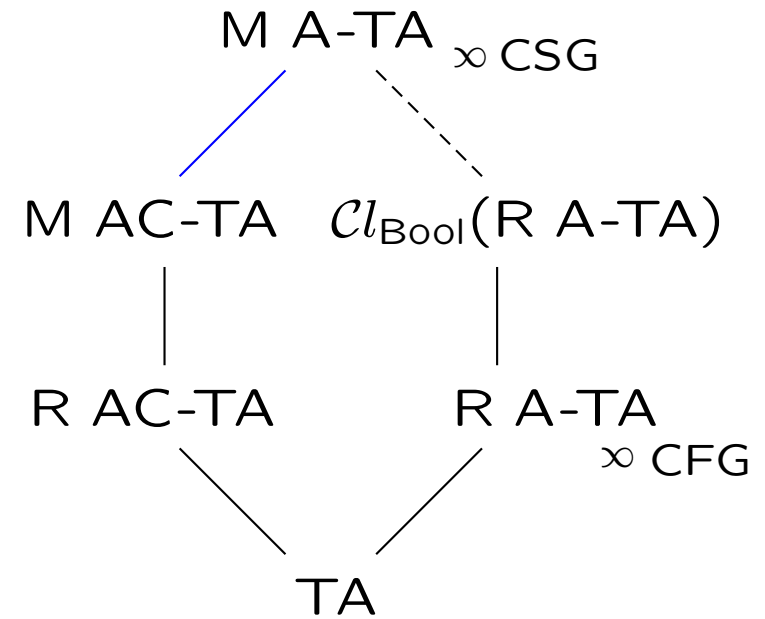
$$E_2 : E_1 - \{ f(x, y) \approx f(y, x) \mid f \in F \}$$

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

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

(Complete the proof in **Exercise**) \square 5



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 $\Delta_A = \bigcup_{f \in F_A} \Delta_f$ (instead of Δ_{AC}),

$$\begin{aligned} \Delta_f : \quad & f(\langle p_1, q_1 \rangle, \langle p_2, q_2 \rangle) \rightarrow f(\langle p, q_1 \rangle, \langle \circ, q_2 \rangle) && \text{if } \exists f(p_1, p_2) \rightarrow p \in \Delta_1, q_1, q_2 \in Q \\ & f(\langle p_1, q_1 \rangle, \langle p_2, q_2 \rangle) \rightarrow f(\langle p_3, q_1 \rangle, \langle p_4, q_2 \rangle) && \text{if } \exists f(p_1, p_2) \rightarrow f(p_3, p_4) \in \Delta_1, q_1, q_2 \in Q \\ & f(\langle \circ, q_1 \rangle, \langle p_2, q_2 \rangle) \rightarrow \langle p_2, q \rangle && \text{if } \exists f(q_1, q_2) \rightarrow q \in \Delta_2, p_2 \in P \\ & f(\langle \circ, q_1 \rangle, \langle p_2, q_2 \rangle) \rightarrow f(\langle \circ, q_3 \rangle, \langle p_2, q_4 \rangle) && \text{if } \exists f(q_1, q_2) \rightarrow f(q_3, q_4) \in \Delta_2, p_2 \in P \\ & \langle p_1, q_1 \rangle \rightarrow \langle p_2, q_1 \rangle && \text{if } \exists p_1 \rightarrow p_2 \in \Delta_1, q_1 \in Q \\ & \langle p_1, q_1 \rangle \rightarrow \langle p_1, q_2 \rangle && \text{if } \exists q_1 \rightarrow q_2 \in \Delta_2, p_1 \in P \\ & f(\langle p_1, q_1 \rangle, \langle \circ, q_2 \rangle) \rightarrow f(\langle \circ, q_1 \rangle, \langle p_1, q_2 \rangle) && \text{if } \exists p_1 \in P, q_1, q_2 \in Q \\ & f(\langle \circ, q_1 \rangle, \langle p_1, q_2 \rangle) \rightarrow f(\langle p_1, q_1 \rangle, \langle \circ, q_2 \rangle) && \text{if } \exists p_1 \in P, q_1, q_2 \in Q. \end{aligned}$$

Complementation

The class of monotone A-TA is closed under complement

Proof

We show the proof for the case that $F = \{f, g\} \cup F_0$, 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 $E = \{f(f(x, y), z) \approx f(x, f(y, z))\}$. Given $\mathcal{A} = (\mathcal{E}, Q, Q_{\text{fin}}, \Delta)$, define $\mathcal{B} = (\mathcal{E}, Q', Q'_{\text{fin}}, \Delta')$ as follows.

$$Q' : 2^Q$$

$$Q'_{\text{fin}} : \{P \subseteq Q \mid P \cap Q_{\text{fin}} = \emptyset\}$$

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

where

$$\Delta_1 : g(P_1, P_2) \rightarrow P$$

if $P_1, P_2 \subseteq Q$ and

$$P = \{p \mid g(p_1, p_2) \rightarrow p \in \Delta \text{ and } p_i \in P_i \ (1 \leq i \leq 2)\}$$

(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)

$$\Delta_2 : \begin{array}{ll} f(\alpha, \beta) \rightarrow \gamma & \text{if } \gamma \rightarrow \alpha \beta \in \mathcal{G}_P \\ f(\alpha, \beta) \rightarrow f(\gamma, \delta) & \text{if } \gamma \delta \rightarrow \alpha \beta \in \mathcal{G}_P \end{array}$$

for all $(\emptyset \neq) P \subseteq Q$ 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

$$\begin{array}{ll} p_3 \rightarrow p_1 p_2 & \text{if } f(p_1, p_2) \rightarrow p_3 \in \Delta \\ p_3, p_4 \rightarrow p_1 p_2 & \text{if } f(p_1, p_2) \rightarrow f(p_3, p_4) \in \Delta \end{array}$$

Φ : mapping onto CSG that adds production rules :

$$p \rightarrow S \quad \text{if } p \in S, S \subseteq Q$$

$$\Delta_3 : \begin{array}{ll} f(\alpha, \beta) \rightarrow \gamma & \text{if } \gamma \rightarrow \alpha \beta \in \mathcal{G}_0 \\ f(\alpha, \beta) \rightarrow f(\gamma, \delta) & \text{if } \gamma \delta \rightarrow \alpha \beta \in \mathcal{G}_0 \end{array}$$

where

\mathcal{G}_0 : CSG such that

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

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

PSPACE problems

The membership problem of CSG :

instance is grammar $\mathcal{G} = (\Sigma, T, N, q_0, \Delta)$, word w

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

This is PSPACE problem

Proof

This problem is \leq_m^P -reducible to the membership problem of LBA (linear bounded automata). The LBA membership problem is in $\text{NSPACE}(n)$ (Appendix in seminar talk 2). From Savitch’s theorem (page 17 in seminar talk 6), $\text{NPSPACE}=\text{PSPACE}$, and thus the CSG membership problem is in PSPACE. \square

Note

The membership problem of LBA is PSPACE-complete (Show that every PSPACE problem is \leq_m^P -reducible to this problem **Exercise**).

Moreover, the (reverse) transformation from the LBA membership problem to the CSG membership problem is also \leq_m^P -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. \square

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**.

($\because \mathcal{L}(\mathcal{A}_{\mathcal{E}}) = \emptyset \Leftrightarrow (\mathcal{L}(\mathcal{A}_{\mathcal{E}}))^c = \mathcal{I}_F \Leftrightarrow \mathcal{L}(\mathcal{A}_{\mathcal{E}}) \subseteq \emptyset$)

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

$$S ::= x \mid \neg S \mid S \wedge S \mid \exists x(S)$$

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}_ε and a tree t such that ψ is true iff \mathcal{A}_ε 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
$t \in \mathcal{L}(\mathcal{A}_\varepsilon)?$	✓	✓	✓	✓
$\mathcal{L}(\mathcal{A}_\varepsilon) = \emptyset?$	—	✓	✓	✓
$\mathcal{L}(\mathcal{A}_\varepsilon) = \mathcal{T}_F?$	—	?	—	✓
$\mathcal{L}(\mathcal{A}_\varepsilon) \subseteq \mathcal{L}(\mathcal{B}_\varepsilon)?$	—	—	—	✓

(✓ : 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 $\{q \in Q \mid t \rightarrow_{\mathcal{A}_{\mathcal{E}}}^* q\} \models \psi$

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

Example

Consider (regular) PTA \mathcal{A} with

$$\Delta : a \rightarrow q_a \quad b \rightarrow q_b \quad f(q_a, r) \rightarrow q_a \quad f(r, r) \rightarrow r \quad f(r, q_b) \rightarrow q_b$$

$$a \rightarrow r \quad b \rightarrow r \quad c \rightarrow r$$

$$\psi : q_a \wedge \neg q_b$$

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

Note

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

(✓ : positive, – : negative)

Tree automata with normalization (TAN)

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

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

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

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 $R_I = \{x + x \rightarrow x\}$ 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 $\mathcal{L}(\mathcal{A}_{\mathcal{R}}) = \emptyset$?

Idempotency $X + X \approx X$

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

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

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

(\because Suppose $\mathcal{A}_{\mathcal{E}}$ accepts $(L)^c$. From **Pumping Lemma**, there exist m, n ($m \neq n$) such that $s^m(0) \xrightarrow{*}_{\mathcal{A}_{\mathcal{E}}} q$ and $s^n(0) \xrightarrow{*}_{\mathcal{A}_{\mathcal{E}}} q$ ($q \in Q$). 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 $\mathcal{R}_{ACI} = (F, E_{AC}, \{s(x) + s(x) \rightarrow s(x)\})$

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

$0 \rightarrow q \quad s(q) \rightarrow q \quad q + q \rightarrow q_f \quad s(q_f) \rightarrow q \quad q + q_f \rightarrow q_f \quad (q_f: \text{final state})$

If eliminate the last two rules, the TAN accepts :

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

Exercise

1. Show that $=_{\varepsilon_1} \subseteq \rightarrow_{\Delta_2/\varepsilon_2}^*$ in page 5.
2. Show that every PSPACE problem is \leq_m^P -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 $\mathcal{C}l_{\text{Bool}}(\text{M AC-TA})$ 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. 17

Appendix : Overview of project ETA (η)

Tohoku Univ (Sendai)



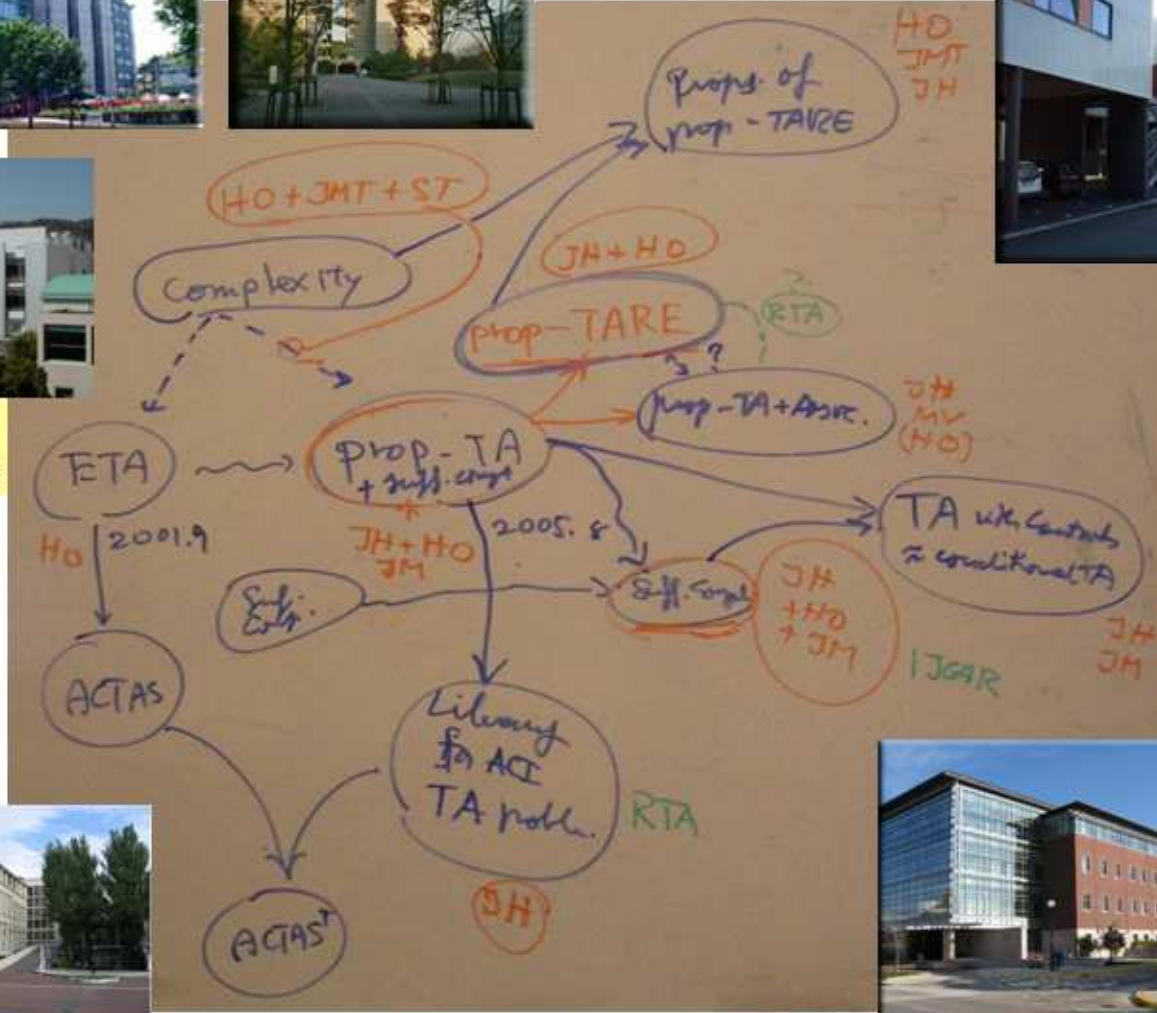
NAIST(Nara)



LIFL (Lille)



AIST (Osaka)



ENS de Cachan (Paris)



Univ of Illinois (Urbana-Champaign)

Copyright (version Jul-01-2009) © 2009 Hitoshi Ohsaki

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/>

All rights reserved.

No part of this lecture material may be reproduced in
any form or by any means, electronic, mechanical, pho-
tocopying, or otherwise, without the prior consent of the
author.