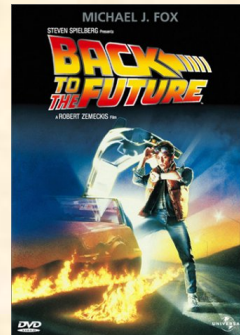


Equational Tree Automata: Past, Present and Future

**Hitoshi Ohsaki
(AIST)**

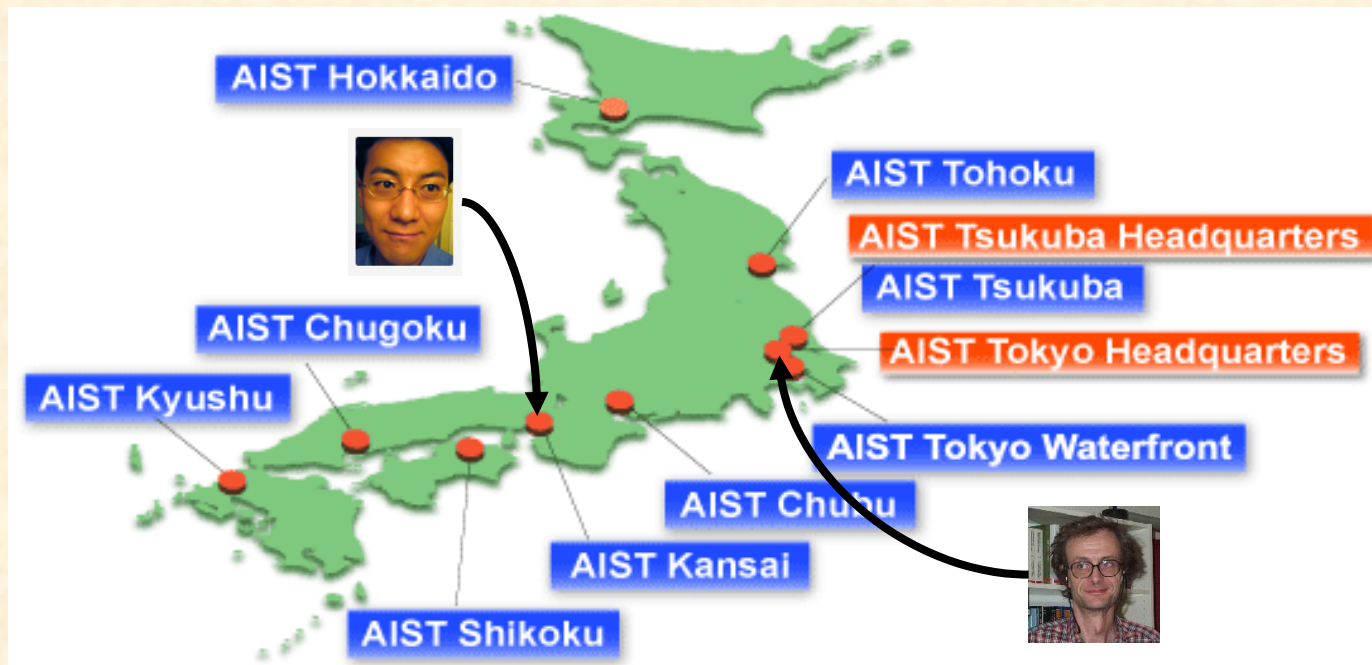


**ENS de Cachan
Nov 19, 2008**

What is **AIST**?

<http://www.aist.go.jp>

AIST: National Institute founded in 2001 by uniting 14 research organizations and 2 others. Currently we have 9 research bases in Japan, with more than 8000 researchers (including 2000 tenure).



* **NAIST, JAIST: Universities for graduate students**

TATA (1997)

Exercise 13. Let $\mathcal{F} = \{f(,), a, b\}$.

1. Let us consider the set of ground terms L_1 defined by the following two conditions:

- $f(a, b) \in L_1$,
- $t \in L_1 \Rightarrow f(a, f(t, b)) \in L_1$.

Prove that the set L_1 is recognizable.

2. Prove that the set L_1 is not recognizable (respectively $|t|_b$ is not recognizable).

Q1. $AC(L)$: regular if L is regular ?

3. Let L be a recognizable tree language over $\mathcal{F}$. Let us suppose that f is a commutative symbol. Let C be the set of equations $C = \{f(x, y) = f(y, x)\}$. Prove that $AC(L)$ is not recognizable.

Q2. $A(L)$: regular if L is regular ?

4. Let L be a recognizable tree language over $\mathcal{F}$. Let us suppose that f is a commutative and associative symbol. Let AC be the congruence closure of set L for the set of equations $AC = \{f(x, y) = f(y, x); f(x, f(y, z)) = f(f(x, y), z)\}$. Prove that in general $AC(L)$ is not recognizable.
5. Let L be a recognizable tree language over $\mathcal{F}$. Let us suppose that f is an associative symbol. Let $A(L)$ be the congruence closure of set L for the set of equations $A = \{f(x, f(y, z)) = f(f(x, y), z)\}$. Prove that in general $A(L)$ is not recognizable.



Tree
Automata
Techniques and
Applications

HUBERT COMON MAX DAUCHET RÉMI GILLERON
FLORENT JACQUEMARD DENIS LUGIEZ SOPHIE TISON
MARC TOMMASI

“AC” **does not** preserve regularity

Q1. $AC(L)$: regular if L is regular ?

A1. No.

Cf. commutative closure of CFL is **not CFL.**

Q2. $A(L)$: regular if L is regular ?

A2. No, but it 's not trivial ...

Remark

There was no general **framework for dealing with equational closure of tree languages.**

Tree automata as ground TRS

$$f(p_1, \dots, p_n) \rightarrow q$$

**Trees with accepted
derivation modulo AC**



**Accepted trees
modulo AC**



Equational rewriting

= equational reasoning + term rewriting



= equational reasoning + transition

Equational tree automata

Equational Tree Automata

$$\mathbf{A} = ((F, \mathbf{E}), Q, Q_{\text{fin}}, T)$$

F : signature

E : equations

Q : states

Q_{fin} : final states

T : transition rules

$$\longrightarrow_A : \xlongequal{\mathbf{E}} \xrightarrow{\mathbf{T}} \xlongequal{\mathbf{E}}$$

September 2001

Equational Tree Automata for Associative and Commutative Theories

Hitoshi Ohsaki

CSL' 01 (organized by LSV) in Paris



2 days before my talk

1. (Commutation Lemma)

$L(A / \equiv_E) = L(A) / \equiv_E$ if E is linear

2. (Monotone Rules)

$$\mathbf{f}(\mathbf{p}_1, \dots, \mathbf{p}_n) \rightarrow \mathbf{f}(\mathbf{q}_1, \dots, \mathbf{q}_n)$$

3. Closedness under Union & Intersection, etc.

Tree Automata Modulo (2002–2003)

- **Presburger Tree Automata**
(Seidl et al.)

Tree automata for XML database

- **Multi-tree Automata**
(Dal Zilio & Lugiez)

AC tree automata = Presburger tree automata

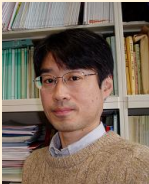
- **Two-way Alternating AC Tree Automata**
(Verma)

$L(A / \equiv_E) = L(A) / \equiv_E$ ~~if E is linear~~

Project ETA



Naoki Kobayashi



Hiroyuki Seki



Ralf Treinen

Tohoku Univ (Sendai)



NAIST(Nara)



Theory

LIFL (Lille)



Sophie Tison

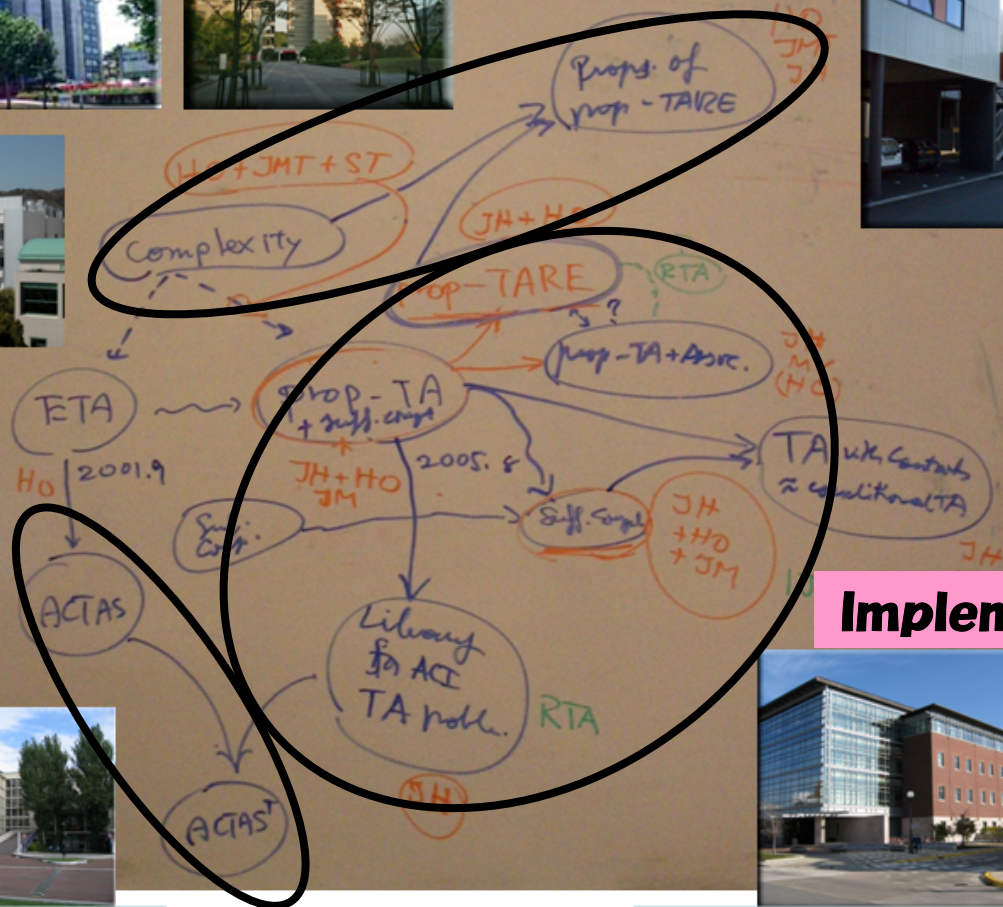


Jean-Marc Talbot



Yves Roos

AIST (Osaka)



Application



ENS de Cachan (Paris)

Implementation



Univ of Illinois (Urbana-Champaign)



Software Development

Ceta: A Library for Equational Tree Automata - Mozilla Firefox
http://formal.cs.uiuc.edu/ceta/

Ceta: A Library for Equational Tree Automata

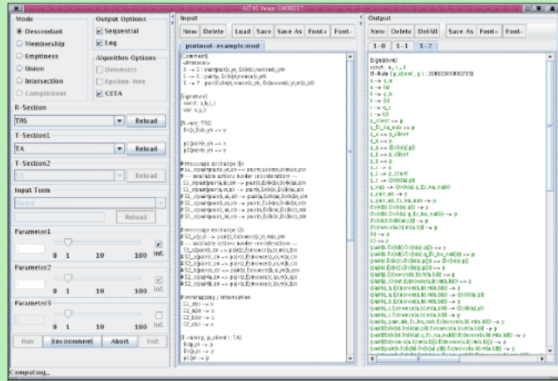
Summary

Ceta is a library for reasoning about combinations of equational tree languages. It supports emptiness testing of tree languages definable by a boolean combination of regular tree automaton over an equational theory containing operators that are associative and/or commutative with identity symbols. Ceta is based on **propositional tree automata**, and offers algorithms and datastructures for representing tree automata, combining tree automata using boolean operations, and testing emptiness.

Ceta is currently being used in the [Maude Sufficient Completeness checker](#), and ACTAS a reachability analysis tool. Using Ceta, ACTAS has been able to verify the security of the Diffie-Hellman key-exchange protocol" and Shamir's three-path protocol without approximation techniques.

Ceta is being developed at the [University of Illinois at Urbana-Champaign](#) in the United States as part of a joint project with [National Institute of Advanced Industrial Science and Technology](#) in Japan. The primary software developer is [Joe Hendrix](#) with significant support from [José Meseguer](#) of UIUC and [Hitoshi Ohsaki](#) of AIST.

The project is currently in active development, and a number of key planned features have not yet been implemented. An alpha version is now available which supports combining equational tree automata and testing emptiness of automaton that are associative, commutative, and free. We do not yet support support rules with constraints, but that is planned for a future release. Ceta is available under the GPL license.



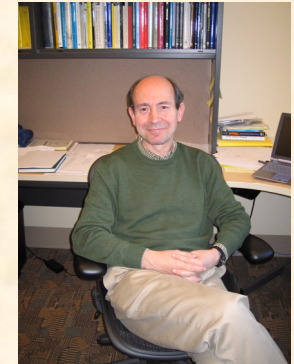
News

Feb. 4, 2007
Released Alpha 4. This release should significantly improve the performance of the emptiness test with AC symbols, and adds support for membership.

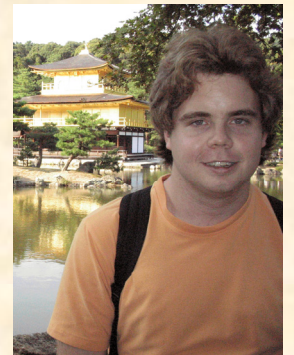
Download

The source code for **Ceta** is now available as a gzipped tar file. Building Ceta requires the followig be installed:

- A C++ compiler (gcc 3.3 or later recommended)



José Meseguer

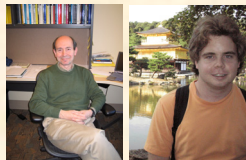


Joe Hendrix

ACTAS(reachability analysis tool) and CETA

1 + 3 Frameworks in Research

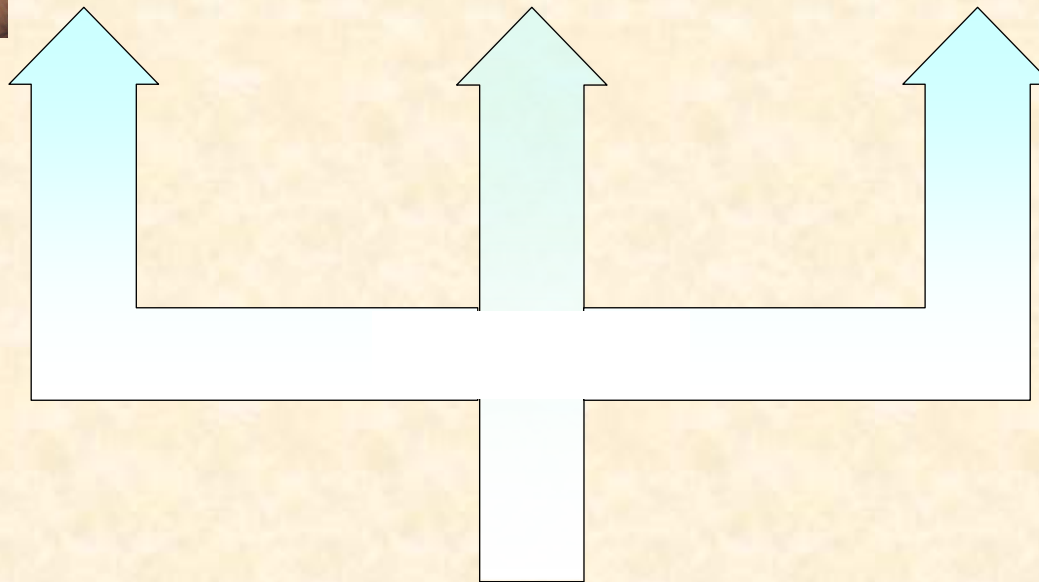
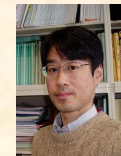
PTA
(Propositional TA)



Monotone ETA



TAN
(TA with Normalization)



(Regular) **Equational Tree Automata**

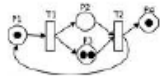


Monotone AC Tree Automata

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

$p \rightarrow q$ [epsilon]

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

	monotone AC-TA	Petri-nets 
$t \in L?$	PSPACE-complete	EXPSPACE-hard
$L = \emptyset?$	EXPSPACE-hard (?)	(decidable)
(✓ : positive – : negative)		
closed under $\cup$	✓	✓
closed under $\cap$	✓	✓
closed under $()^c$	–	–

... for Non-linear Arithmetic

decidable

undecidable

Monotone AC tree automata

Monotone exponential Diophantine arithmetic

$M ::= E(\vec{x}_n) \geq L(\vec{x}_n) \mid C \text{ (Presburger formula)} \mid M \vee M \mid M \wedge M$
 $E(\vec{x}_n) ::= a \mid x_i \mid E(\vec{x}_n) + E(\vec{x}_n) \mid E(\vec{x}_n) \times E(\vec{x}_n) \mid E(\vec{x}_n)^{E(\vec{x}_n)}$

$$x^{x^x} \geq y$$

Diophantine arithmetic

Every formula in  is definable by monotone AC tree automata



RTA'08 Best Paper Award

Open Questions

1. Universality problem of monotone AC-TA

- ▶ The emptiness is decidable, but the inclusion is **not**.
- ▶ The question is in RTA Open Question List (#101).

2. The class of monotone AC-TA is closed under projection?

- ▶ Or, every Petri-net definable language is definable by monotone AC tree automata?

3. Emptiness problem of $AC(L) \cap NF[f(x,x) \rightarrow x]$

Solved

Decidable (**Hendrix&Ohsaki** RTA '08)

Tree Automata with $f(x,x) = x$

ETA



**Tree automata modulo $f(x,x) = x$ axiom
are not closed under complement**

TAN

**Tree automata with $f(x,x) \rightarrow x$ normalization
are closed under complement**

**Moreover, this class of tree languages
is closed under all Boolean operations,
and the emptiness problem is decidable.**

Lecture note on Tree Language Theory

10 lectures from basics to advanced topics

Activities:

**Seminar talks
at UIUC, Illinois
(October 2007).**

**Lecture course
at Track(B), 4th ISR
(June 2009).**

Draft available:

<http://staff.aist.go.jp/hitoshi.ohsaki/lecturenote.html>

Introduction to tree language theory

These lectures will provide an introduction to the tree language theory in a self-contained manner. Topics of the first 5 lectures review the introductory topics: tree automata, term rewriting, and the basics of formal languages including automata, computability, complexity, and linear algebras. Advanced topics cover equational tree automata and the further extensions, from propositional tree automata (PTA) to tree automata with normalization (TAN). Characterization of each framework can be studied through the discussion of the decidability and closure properties and decidability.

Contents

1. Words
2. Grammar
3. Tree automata
4. Modulo axioms
5. P, NP & PSPACE

* Section 1 includes "Second-order logic"

Lecturenote

General information

About "Introduction to Tree Language Theory"

This lecture course will provide an introduction to the tree language theory in a self-contained manner. The course consists of 10 lectures. The first 5 lectures review the introductory topics: tree automata, term rewriting, and the basics of formal languages including automata, computability, complexity, and linear algebras. Each lecture can be useful for undergraduate or graduate students in learning or reviewing tree automata and the background. The remaining 5 lectures explain advanced topics, equational tree automata (ETA) and the further extensions, from propositional tree automata (PTA) to tree automata with normalization (TAN). The characterization of each tree automata framework can be studied through the discussion of the decidability and closure properties.

Those who are interested in technical details, omitted proofs or open questions, are recommended to refer to [our papers](#). When you find errors in the text or have suggestions for betterment, kindly ask you to contact [the author](#).

Download

Latest draft: [book-20080901.pdf](#)

List of contents

Section 1: [Words](#) (and Monadic Second-Order Logic as an advanced topic)

Section 2: [Grammar](#)

Section 3: [Trees](#)

Section 4: [Modulo axioms](#)

Section 5: [Turing Machine](#)

Section 6: [P and NP](#)

Section 7: [Parikh's theorem](#)

International School on Rewriting (ISR)

Tree Automata

Program	Monday, 22/06/2009	Tuesday, 23/06/2009	Wed, 24/06/2009	Thursday, 25/06/2009
08:30h – 09:00h	Registration	Traditional and non-Traditional Typed λ-calculus Fairouz Kamareddine (Edinburgh - Scotland)		Traditional and non-Traditional Typed λ-calculus Fairouz Kamareddine (Edinburgh - Scotland)
09:00h – 09:30h	Opening			
09:30h – 10:00h	Introduction to Term Rewriting Franz Baader (Dresden - Germany)			
10:00h – 10:30h		Coffee-Break		Coffee-Break
10:30h – 11:00h	Coffee-Break			Coffee-Break
11:00h – 11:30h	Introduction to Term Rewriting Franz Baader (Dresden - Germany)	Equational Tree Automata Hitoshi Ohsaki (Osaka - Japan)		Equational Tree Automata Hitoshi Ohsaki (Osaka - Japan)
11:30h – 12:00h				Equational Tree Automata Hitoshi Ohsaki (Osaka - Japan)
12:00h – 14:00h	Lunch	Lunch	Excursion	Lunch
14:00h – 14:30h	Introduction to Term Rewriting Franz Baader (Dresden - Germany)	Equational Tree Automata Hitoshi Ohsaki (Osaka - Japan)		Equational Tree Automata Hitoshi Ohsaki (Osaka - Japan)
14:30h – 15:00h				
15:00h – 15:30h	Coffee-Break			
15:30h – 16:00h	Introduction to Term Rewriting Franz Baader (Dresden - Germany)			
16:00h – 16:30h	Coffee-Break			
16:30h – 17:00h				
17:00h – 17:30h	Introduction to Term Rewriting Franz Baader (Dresden - Germany)			
17:30h – 18:00h				

Track(B)

Program	Monday, 22/06/2009	Tuesday, 23/06/2009	Wed, 24/06/2009	Thursday, 25/06/2009	Friday, 26/06/2009
08:30h – 09:00h	Registration				
09:00h – 09:30h	Opening	Conditional Rewriting Bernhard Gramlich (Wien - Austria)		Higher-Order Rewriting with Resources Delia Kesner (Paris - France)	Nominal Rewriting Maribel Fernandez (London - UK)
09:30h – 10:00h	Advanced Topics in Termination Aart Middeldorp (Innsbruck - Austria)				
10:00h – 10:30h		Coffee-Break		Coffee-Break	Coffee-Break
10:30h – 11:00h	Coffee-Break				
11:00h – 11:30h	Conditional Rewriting Bernhard Gramlich (Wien - Austria)	Advanced Topics in Termination Aart Middeldorp (Innsbruck - Austria)		Nominal Rewriting Maribel Fernandez (London - UK)	Higher-Order Rewriting with Resources Delia Kesner (Paris - France)
11:30h – 12:00h					
12:00h – 14:00h	Lunch	Lunch	Excursion	Lunch	Lunch
14:00h – 15:00h	Extensions of Typed λ-calculus with Applications in Logic, Programming Languages and Formalisation of Mathematics Fairouz Kamareddine (Edinburgh - Scotland)	Extensions of Typed λ-calculus with Applications in Logic, Programming Languages and Formalisation of Mathematics Fairouz Kamareddine (Edinburgh - Scotland)		Advanced Topics in Termination Aart Middeldorp (Innsbruck - Austria)	Advanced Topics in Termination Aart Middeldorp (Innsbruck - Austria)
15:00h – 16:00h	Extensions of Typed λ-calculus with Applications in Logic, Programming Languages and Formalisation of Mathematics Fairouz Kamareddine (Edinburgh - Scotland)	Extensions of Typed λ-calculus with Applications in Logic, Programming Languages and Formalisation of Mathematics Fairouz Kamareddine (Edinburgh - Scotland)		Conditional Rewriting Bernhard Gramlich (Wien - Austria)	Conditional Rewriting Bernhard Gramlich (Wien - Austria)
16:00h – 16:30h	Coffee-Break	Coffee-Break		Coffee-Break	Coffee-Break
16:30h – 17:30h	Parametric Lambda Calculus Luca Paolini (Torino - Italy)	Parametric Lambda Calculus Luca Paolini (Torino - Italy)		Parametric Lambda Calculus Luca Paolini (Torino - Italy)	Parametric Lambda Calculus Luca Paolini (Torino - Italy)

Lecturers of ISR '09

Track(A)

Franz Baader (Dresden)

Bernhard Gramlich (Wien)

Fairouz Kamareddine (Edinburgh)

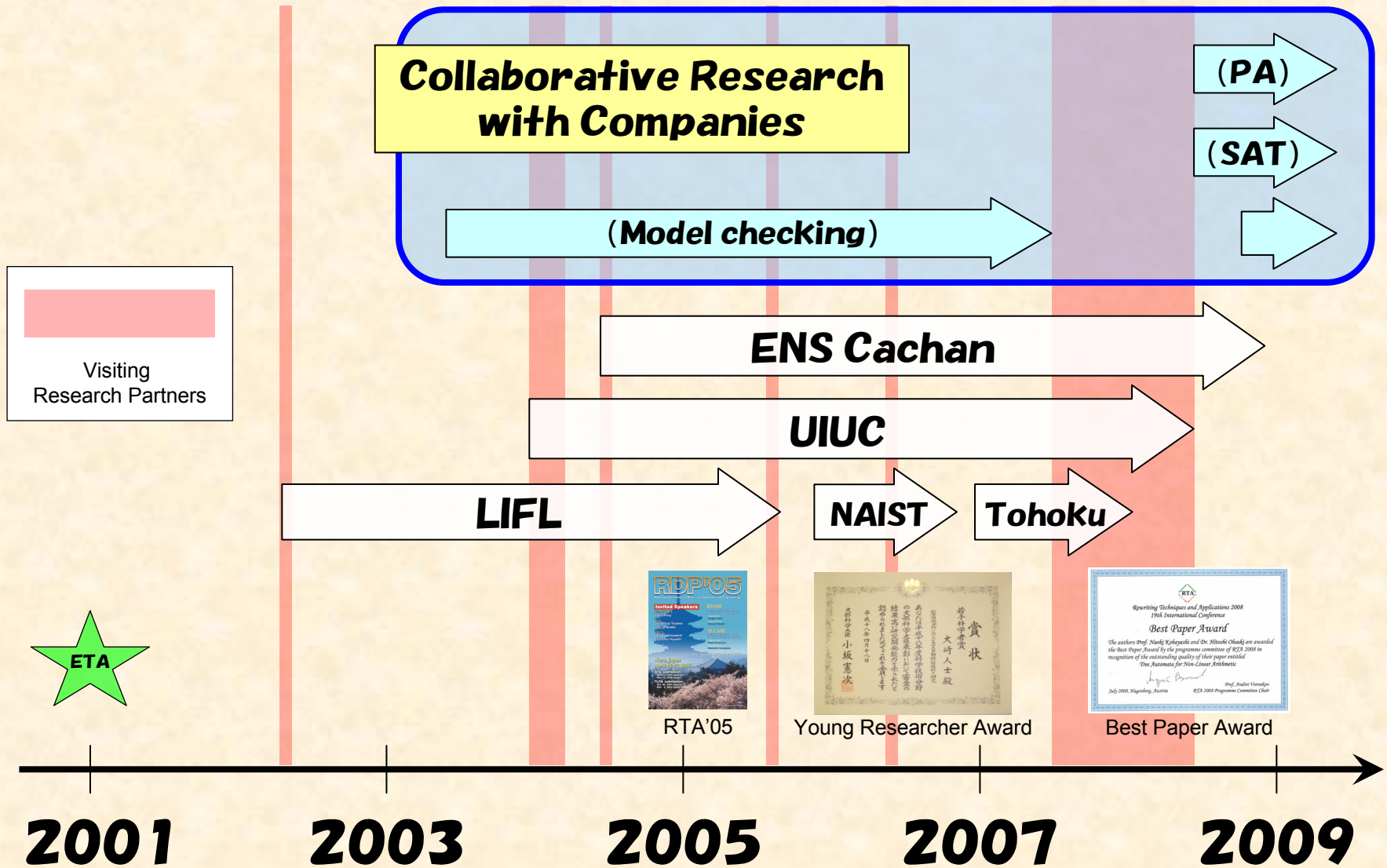
Delia Kesner (Paris)

Aart Middeldorp (Innsbruck)

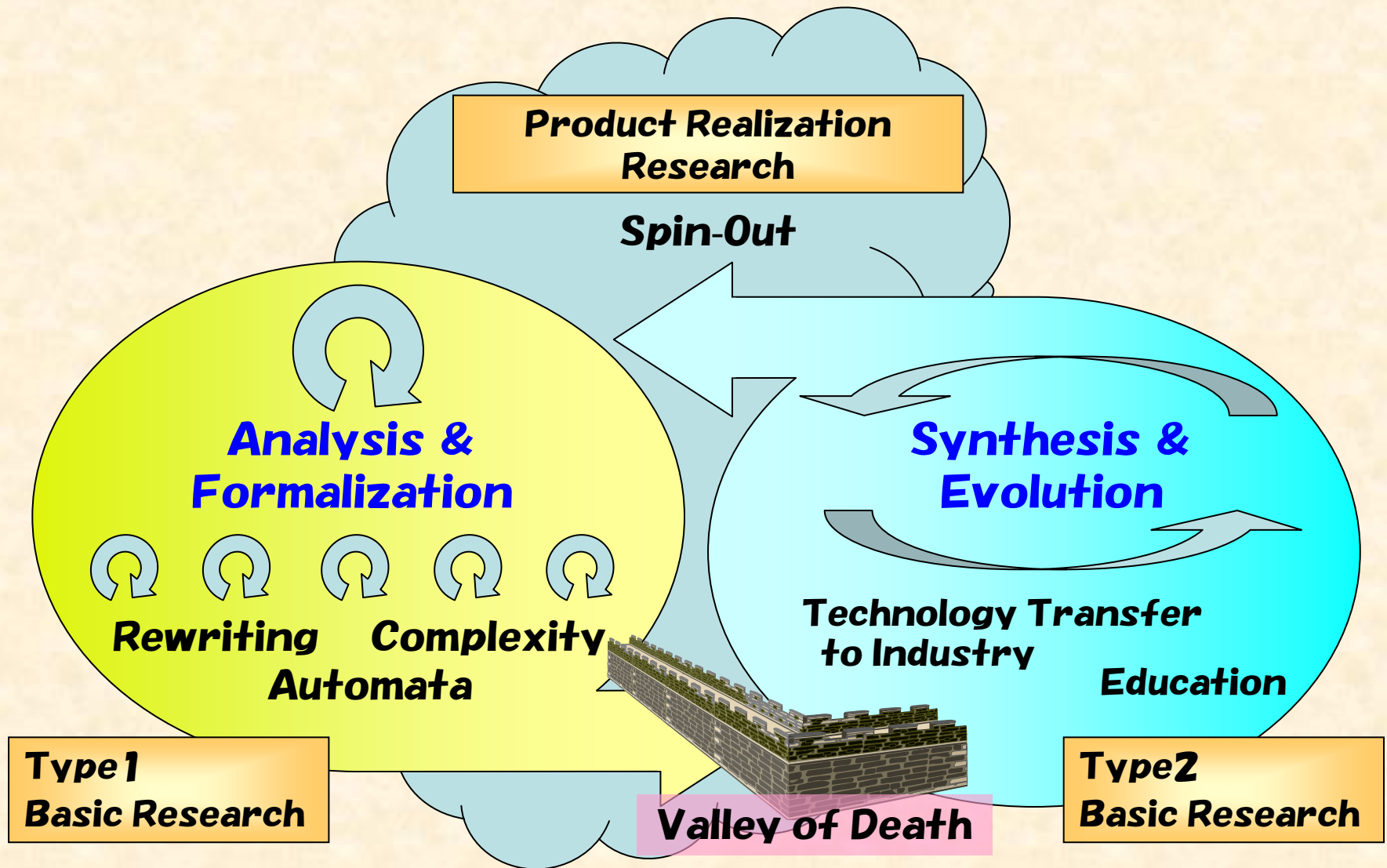
Hitoshi Ohsaki (Osaka)

Luca Paolini (Torino)

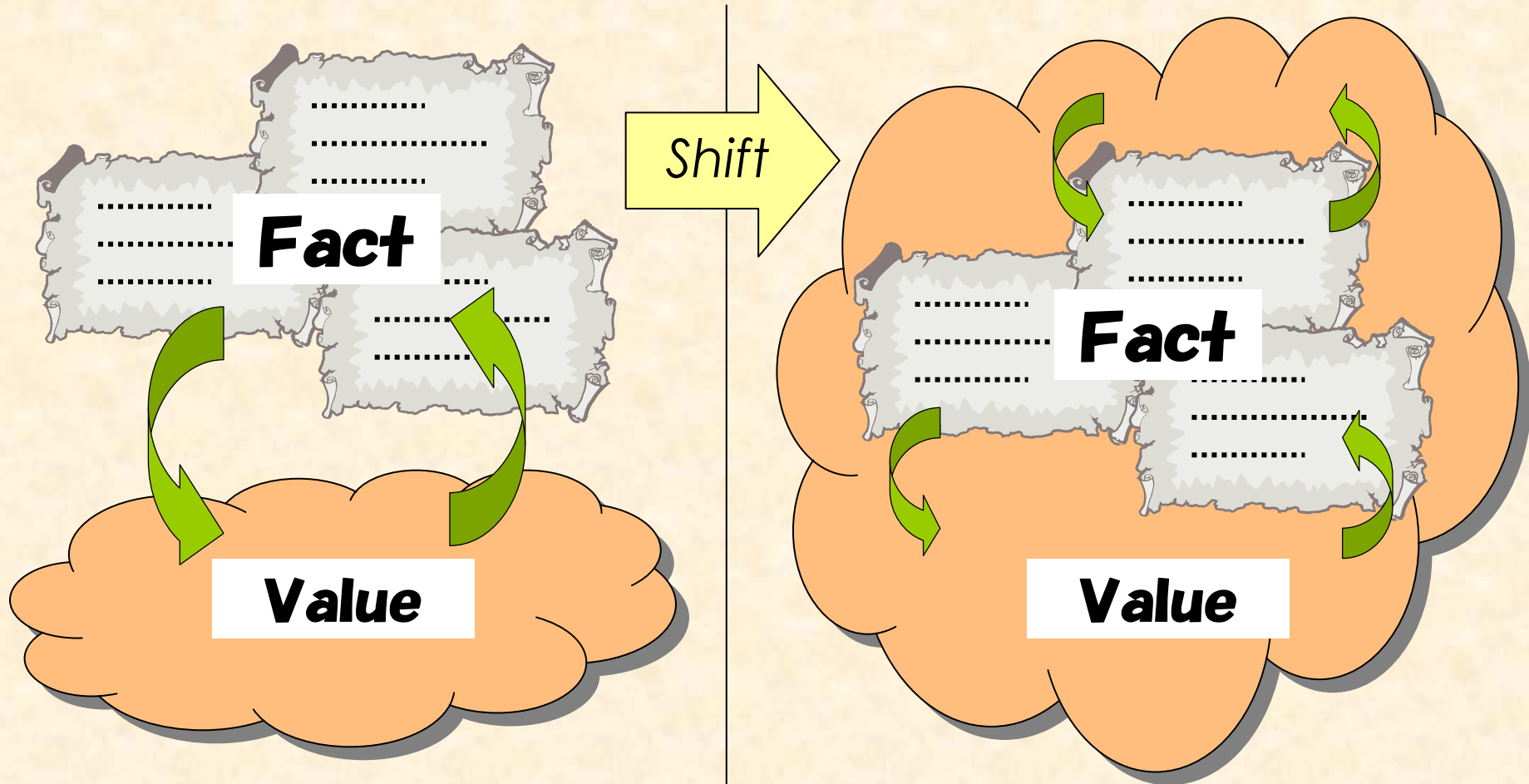
Past Present ...



Three Research Types in AIST



Collapsing Fact & Value Dichotomy



(Hilary Putnam, 2002)

AIST Journal “Synthesiology”

Synthesiology = “synthesis” + “-ology” (disciplines)



The Aim of Launching “Synthesiology”
 ... **Technology integration research** (**Type 2 Basic Research**) activities have been kept as personal know-how. They **have not been formalized as universal knowledge of what ought to be done.** ... Each paper in this journal will present scenarios selected for their social value, identify elemental knowledge and/or technologies to be integrated, and describe the procedures and processes to achieve this goal.

More about “Synthesiology”

Articles (PDF files) are available:

<http://www.aist.go.jp/synthesiology/>

Anyone can submit articles.

**Cf. On the Launch of “Synthesiology”:
Discussion with Japanese Researchers at UIUC**

Ohsaki (AIST) & Sato (Hosei Univ)

Editorial, *Synthesiology* 1(3), pp. 222–228 (2008)

Any question or comment is welcome:

ohsaki@ni.aist.go.jp

Copyright © 2008 Hitoshi Ohsaki

National Institute of Advanced Industrial Science and Technology

Office: AIST Kansai – Senri site (MSK bldg. 5th floor)

Address: Shin-Senri Nishi 1-2-14, 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 without the prior consent of the author.