Recursion Theory for Process Model

Yuxi Fu

BASICS, Shanghai Jiao Tong University

Symposium in honour of Pierre-Louis Curien
Thanks to Pierre-Louis.
Best wishes to Pierre-Louis.
Computation and Interaction

**Thesis on Computation.** All computation models share a common submodel that is physically implementable.

**Thesis on Interaction.** All interaction models share a common submodel.
We look for a theory of models that allows one to carry out model independent investigations for all models of interest.
I. Motivation
Point I. In Computer Science a lot of models have been proposed. There is not yet a model theory.

- Computation Models
- Concurrency Models

Well, they are all about interactions.

Computation Theory and Process Theory have been two separated developments. An integrated treatment ought to be beneficial to both theories.
Why Model Theory?

Point II. Some of the foundational assumptions of Computer Science are actually postulates in Model Theory.

- In Computability Theory, Church-Turing Thesis.
- In Complexity Theory, Extended Church-Turing Thesis.
- In Programming Theory, existence of universal program.

There is no way to formalize these foundational assumptions without a theory of models.
Why Model Theory?

Point III. Most basic concepts in Computer Science are model independent.

- expressiveness
- implementation
- correctness

Are there any basic concepts in Computer Science that are not model independent?
Why Model Theory?

Point IV. Some of the fundamental problems in Computer Science are best understood when cast in the light of Model Theory.

- ‘NP $\neq$ P?’
- Compare the above problem to ‘BPP = P?’
II. Two Relations
Equality and Submodel Relation

Model Theory begins with two most fundamental relations:

- the equality relationship ‘=’ within a model, and
- the expressiveness relationship ‘⊆’ between models.

Both = and ⊆ must be defined in a model independent manner.
Four Principles

I. Principle of Object. There are two kinds of objects.

II. Principle of Action. There are two aspects of actions.

III. Principle of Observation. There are two universal operators.

IV. Principle of Consistency. There are two unequal objects.
Application of the Principles

In computation theory

- bisimulation is implicit in equivalence proofs
- divergent computation \( \neq \) terminating computation

In process theory

- Milner and Park’s bisimulation
- van Glabbeek and Weijland’s branching bisimulation
- Milner and Sangiorgi’s barbed bisimulation
Intensional Requirement

A symmetric relation $\simeq$ is a **bisimulation** if it validates the following bisimulation property:

- If $Q \simeq P \xrightarrow{\tau} P'$ then one of the following is valid:
  1. $\exists Q'. Q \implies Q' \simeq P' \land Q' \simeq P$.
  2. $\exists Q', Q''. Q \implies Q'' \simeq P \land Q'' \xrightarrow{\tau} Q' \simeq P'$.

A symmetric relation $\simeq$ is **codivergent** if the following codivergence property holds whenever $P \simeq Q$:

- If $P \xrightarrow{\tau} P_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} P_{i+1} \ldots$ is an infinite internal action sequence then $\exists Q'. \exists i \geq 1. Q \xrightarrow{\tau} Q' \simeq P_i$.
A symmetric relation $\simeq$ is **equipollent** if $P \simeq Q$ implies that
- $P$ can interact if and only if $Q$ can interact.

A symmetric relation $\simeq$ is **extensional** if the following are valid:
- If $M \simeq N$ and $P \simeq Q$ then $(M \upharpoonright P) \simeq (N \upharpoonright Q)$.
- If $P \simeq Q$ then $(a)P \simeq (a)Q$ for every name $a$. 
Absolute Equality, Subbisimilarity

The absolute equality $=_{M}$ is the largest relation on $M$-processes that validates the following statements:

1. It is reflexive.
2. It is equipollent, extensional, codivergent and bisimilar.
Absolute Equality, Subbisimilarity

The **absolute equality** $=_{\mathcal{M}}$ is the largest relation on $\mathcal{M}$-processes that validates the following statements:

1. It is reflexive.
2. It is equipollent, extensional, codivergent and bisimilar.

A relation $\mathcal{R}$ from the set of $\mathcal{M}_0$-processes to the set of $\mathcal{M}_1$-processes is a **subbisimilarity**, notation $\mathcal{R} : \mathcal{M}_0 \rightarrow \mathcal{M}_1$, if it validates the following statements:

1. It is total and sound.
2. It is equipollent, extensional, codivergent and bisimilar.

We write $\mathcal{M}_0 \sqsubseteq \mathcal{M}_1$ if there is some subbisimilarity from $\mathcal{M}_0$ to $\mathcal{M}_1$. 

---

Yuxi Fu

Recursion Theory for Process Model
Remark

$P =_{M} Q$ means that $P, Q$ are equal objects/processes of model $M$.

$M \sqsubseteq N$ means that $N$ is at least as expressive as $M$.

Using $=$ and $\sqsubseteq$, one may formulate postulates for the class $M$ of models. For example

$$\forall L, N \in M. \exists M \in M. L \sqsubseteq M \sqsubseteq N.$$
III. Axiom of Completeness
To start with we need to formalize Church-Turing Thesis.
Initial Model $\mathbb{C}$

Grammar of $\mathbb{C}$:

$$P := 0 \mid \Omega \mid F^b_a(f(x)) \mid \overline{a}(i) \mid P \mid P,$$

where $f$ is a computable function and $i$ is a natural number.

Semantics of $\mathbb{C}$:

- $F^b_a(f(x)) \xrightarrow{a(i)} \overline{b}(j)$ if $f(i) = j$;
- $F^b_a(f(x)) \xrightarrow{a(i)} \Omega$ if $f(i) \uparrow$;
- $\overline{a}(j) \xrightarrow{\overline{a}(j)} 0$;
- $\Omega \xrightarrow{\tau} \Omega$. 
Formalizing Church-Turing Thesis

Axiom of Completeness. \( \forall M \in \mathcal{M}. \, C \subseteq M. \)

A model \( \mathcal{M} \) is said to be complete if \( C \subseteq M \).
Some Complete Models

Theorem

Both $\mathbb{VPC}$ and $\pi$ are complete.
Some Incomplete Models

Theorem

*Neither CCS nor the higher order process calculus is complete.*
IV. Computation Theory
Before looking at computation theory in this framework, let’s take a look at nondeterministic structures of computation. It is an example of how $=$ and $\sqsubseteq$ help.
Nondeterminism

A one-step deterministic computation $A \rightarrow B$ is an internal action $A \xrightarrow{\tau} B$ such that $A = B$.

A one-step nondeterministic computation $A \xrightarrow{\iota} B$ is an internal action $A \xrightarrow{\tau} B$ such that $A \neq B$. 
Theorem

There is a complete axiomatic system for the finite computations.
Infinite Structure
Infinite Structure

It can be defined for example in CCS.

\[ \text{Centipeda} = (inc)(dec)(o)(e)(Cp \mid Cnt \mid o.O \mid e.E), \]

where

\[
\begin{align*}
Cp &= \tau.\gamma_0 + \tau.(\tau.\gamma_1 + \tau.(\overline{o} \mid !o.inc.\overline{e} \mid !e.inc.\overline{o})), \\
Cnt &= inc.(d)(A(d) \mid d), \\
A(x) &= dec.\overline{x} + inc.(d)(A(d) \mid d.A(x)), \\
E &= \mu X.(\tau.X + \tau + \text{dec}.O), \\
O &= \tau + \tau.\Omega + \text{dec}.E.
\end{align*}
\]
Nondeterminism is Model Independent

Theorem

\( \forall \mathcal{M} \in \mathcal{M}. \forall P \in \mathcal{M}. \neg (P \downarrow) \Rightarrow \exists Q \in \mathcal{C}. \neg (Q \downarrow) \land Q = P. \)
V. Process Theory
Expressiveness of process models can be studied from the point of view of completeness and the relation \( \sqsubseteq \).
Largest Subbisimilarity?

Theorem

There are an infinite number of subbisimilarities into $\mathbb{VPC}^!$.

Theorem

The largest subbisimilarity from a $\pi$-variant to itself exists and coincides with the absolute equality.
Old Result in New Theory
Theorem

$\forall \pi L \subsetneq \pi \subsetneq \forall \pi R \subsetneq \forall \pi M \subsetneq \forall \pi S \subsetneq VPC_{\text{def}}$.

Theorem

polyadic $\pi \not\sqsubseteq$ monadic $\pi$. 

Yuxi Fu  
Recursion Theory for Process Model 35 / 49
VI. Programming Theory
Programming Theory is based on the existence of interpreter/universal process.
Suppose \( L, M \) are complete and \( L \subseteq M \).

We intend to formalize the idea that \( M \) is capable of interpreting all the \( L \)-processes within \( M \).
Interpreter

We write $L \in M$ if there is an interpreter of $L$ in $M$. 
Let $\text{VPC}^!$ be the value-passing calculus with replication, and \( \text{VPC}^{\text{def}} \) the value-passing calculus with parametric definition.

**Theorem**

$\text{VPC}^! \in \text{VPC}^{\text{def}}$. 
An index of a \( \mathsf{VPC}^! \)-process can be defined as follows:

\[
\begin{align*}
\langle 0 \rangle_v & \overset{\text{def}}{=} 0, \\
\langle a(x).T \rangle_v & \overset{\text{def}}{=} 7 \langle \varsigma(a), \varsigma(x), \langle T \rangle_v \rangle + 1, \\
\langle \overline{a}(t).T \rangle_v & \overset{\text{def}}{=} 7 \langle \varsigma(a), \langle t \rangle_\varsigma, \langle T \rangle_v \rangle + 2, \\
\langle T | T' \rangle_v & \overset{\text{def}}{=} 7 \langle \langle T \rangle_v, \langle T' \rangle_v \rangle + 3, \\
\langle (c)T \rangle_v & \overset{\text{def}}{=} 7 \langle \varsigma(c), \langle T \rangle_v \rangle + 4, \\
\langle \text{if } \varphi \text{ then } T \rangle_v & \overset{\text{def}}{=} 7 \langle \langle \varphi \rangle_\varsigma, \langle T \rangle_v \rangle + 5, \\
\langle !a(x).T \rangle_v & \overset{\text{def}}{=} 7 \langle \varsigma(a), \varsigma(x), \langle T \rangle_v \rangle + 6, \\
\langle !\overline{a}(t).T \rangle_v & \overset{\text{def}}{=} 7 \langle \varsigma(a), \langle t \rangle_\varsigma, \langle T \rangle_v \rangle + 7.
\end{align*}
\]
The simulator $S_v(z)$ is defined by the following

\textbf{case} \ z \ \textbf{of}
\begin{align*}
  r_7(z) = 0 & \Rightarrow 0; \\
  r_7(z) = 1 & \Rightarrow Nth(d_7(z)_0,j).a_j(x).S_v([x/d_7(z)_1]d_7(z)_2); \\
  r_7(z) = 2 & \Rightarrow Nth(d_7(z)_0,j).\overline{a_j}(val(d_7(z)_1)).S_v(d_7(z)_2); \\
  r_7(z) = 3 & \Rightarrow S_v(d_7(z)_0) \mid S_v(d_7(z)_1); \\
  r_7(z) = 4 & \Rightarrow Nth(d_7(z)_0,j).(a_j)S_v(d_7(z)_1); \\
  r_7(z) = 5 & \Rightarrow \text{if \ } val(d_7(z)_0) \ \text{then} \ S_v(d_7(z)_1); \\
  r_7(z) = 6 & \Rightarrow Nth(d_7(z)_0,j).!a_j(x).S_v([x/d_7(z)_1]d_7(z)_2); \\
  r_7(z) = 7 & \Rightarrow Nth(d_7(z)_0,j).!\overline{a_j}(val(d_7(z)_1)).S_v(d_7(z)_2).
\end{align*}

\textbf{end case}

Parametric definition plays an essential role in the simulator.
Universal Process

A model $M$ admits a universal process if $M \in M$.

Theorem

- $\pi \in \pi$.
- $\text{VPC}^{\text{def}} \in \text{VPC}^{\text{def}}$.

It is highly unlikely that $\text{VPC}^! \in \text{VPC}^!$. 
We say that $M$ is a **programming model** if $M \in M$.

In the world of programming models, $L \subseteq M$ iff $L \in M$. 
Axiom of Programming. $\forall M. M \in M.$
Application

1. Modeling security protocol
2. Process-Passing as Value-Passing
VII. Recursion Theory
Recursion Theory for Process

The existence of a universal process allows one to develop a Recursion Theory for an interaction model.

Enumeration Theorem, S-m-n Theorem, Recursion Theorem; and then the rest of it.

Reductions between processes, process degrees, ...
@ http://basics.sjtu.edu.cn/~yuxi/

- F. The Universal Process.
- F. Nondeterministic Structure of Computation.
- F. Theory of Interaction.
- F. and Zhu. The Name-Passing Calculus.
- Cai and F. The $\lambda$-Calculus in the $\pi$-Calculus. MSCS, 2011.

Thanks!