The Power of Abstraction
in a Theory, a Project, and an Inspiring Mind

Symposium on the Categorical Unity of the Sciences
Kyoto, Japan. March 22, 2019

Ichiro Hasuo
National Institute of Informatics & SOKENDAI
Research Director, JST ERATO MMSD Project
Why Categories?

- Why category theory?

- My answer: for **generalization** & **abstraction**

  - A plain math. language

  - Equipped only with a (typed) **compositional structure**

  - A **nicely extendable** base language: accommodates various additional features (monoidal, topos, locally presentable/accessible, effectus, ...)

  - Make **theory builders** aware of what exactly (s)he is using, and thus extract the essence of a theory

- Then, why **abstraction**??

“Unity” as Yoshi talked about (?)

“**Scalability** is the end, **compositionality** is the means, **category theory** is the means to the means.” (Jules Hedges)
The Power of Abstraction

in a theory

in a project

... and in an inspiring mind
Outline

* Abstraction in a Theory I
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02] [Hasuo & Hoshino, LICS’11] [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]

* Abstraction in a Theory II
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* Abstraction in a Project
  Introducing ERATO MMSD Project:
  from categorical foundations to automated driving

* Abstraction in an Inspiring Mind
Abstraction in a Theory: “Categorical Transfer”

Abstract Technique

\[ T[\_] \]

Identify “mathematical essence”

Choose parameter \( e_1 \)

Existing Technique

\[ T_1 = T[e_1] \]

Choose parameter \( e_2 \)

Novel Technique

\[ T[e_2] \]
Outline

* Abstraction in a Theory I
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02]  [Hasuo & Hoshino, LICS’11]  [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16]  [Hasuo & Hoshino, APAL’17]

* Abstraction in a Theory II
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15]  [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* Abstraction in a Project
  Introducing **ERATO MMSD** Project:
  from categorical foundations to automated driving

* Abstraction in an Inspiring Mind
Abstraction in a Theory I
Categorical GoI

\[ f : X \times \mathbb{N} \rightarrow T(X \times \mathbb{N}) \]

[Abramsky, Haghverdi & Scott, MSCS’02]
[Hasuo & Hoshino, LICS’11] [Hoshino, Muroya & Hasuo, CSL-LICS’14]
[Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]
Abstraction in a Theory:
“Categorical Transfer”

Abstract Technique

\[ T[\_] \]

Identify “mathematical essence”

Choose parameter \( e_1 \)

Choose parameter \( e_2 \)

Existing Technique

\[ T_1 = T[e_1] \]

Novel Technique

\[ T[e_2] \]

Geometry of Interaction [Girard]

Gol with nondeterministic, probabilistic, quantum, and other effects

… \( \rightarrow \) Memoryful Geometry of Interaction [Hoshino, Muroya, Hasuo, …]
Outline

“GoI Animation”

Categorical GoI

[ Abramsky, Haghverdi & Scott, MSCS’02]

GoI w/ $T$-branching

[ IH & Hoshino, LICS’11]

Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS’14 & POPL’16]

Coalgebra meets higher-order computation in Geometry of Interaction

[ Girard, LC’88]
Geometry of Interaction (GoI)

J.-Y. Girard, at Logic Colloquium ’88

Provides “denotational” semantics (w/ operational flavor) for linear \( \lambda \)-term \( M \)

As a compilation technique

[McKee, POPL’95] [Pinto, TLCA’01] [Ghica et al., POPL’07, POPL’11, ICFP’11, ...]

Two presentations:

* (Operator-) Algebraic [Girard]

* Token machines/ interaction abstract machines
  [Danos & Regnier, TCS’99] [McKee, POPL’95]
The GoI Animation

\[
[M] = (\mathbb{N} \rightarrow \mathbb{N}, \text{a partial function})
\]

= “piping”

... (countably many)
The GoI Animation

- Function application $[MN]$
- by “parallel composition + hiding”
$[MN]$ = "parallel composition + hiding" (cf. AJM games)
\[ [M N] \]

\[
\begin{align*}
M & = \lambda x. x + 1 \\
N & = 2 \\
M & = \lambda x. 1 \\
N & = 2 \\
M & = \lambda f. f1 \\
N & = \lambda x. (x + 1)
\end{align*}
\]
Outline

Coalgebra meets higher-order computation in Geometry of Interaction [Girard, LC’88]

“GoI Animation”

Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS’02]
Categorical GoI

Axiomatics of GoI in the categorical language

Our main reference:


Especially its technical report version (Oxford CL), since it’s a bit more detailed

See also:

arxiv.org/abs/1605.05079
The Categorical GoI Workflow

Traced monoidal category $C$ + other constructs $\Rightarrow$ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

Applicative str. + combinators

Model of untyped calculus

PER, $\omega$-set, assembly, ...

"Programming in untyped $\lambda$"

Model of typed calculus
GoI situation

**Defn.** (GoI situation [AHS02])

A *GoI situation* is a triple \((C, F, U)\) where

- \(C = (C, \otimes, I)\) is a traced symmetric monoidal category (TSMC);
- \(F : C \to C\) is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  
  \[
  \begin{align*}
  e : FF \triangleleft F & : e' \quad \text{Comultiplication} \\
  d : \text{id} \triangleleft F & : d' \quad \text{Dereliction} \\
  c : F \otimes F \triangleleft F & : c' \quad \text{Contraction} \\
  w : K_I \triangleleft F & : w' \quad \text{Weakening}
  \end{align*}
  \]

  Here \(K_I\) is the constant functor into the monoidal unit \(I\);

- \(U \in C\) is an object (called *reflexive object*), equipped with the following retractions.

  \[
  \begin{align*}
  j : U \otimes U \triangleleft U & : k \\
  I \triangleleft U & \\
  u : FU \triangleleft U & : v
  \end{align*}
  \]
GoI situation

**Defn.** (GoI situation [AHS02])

A GoI situation is a triple \((C, F, U)\) where

- \(C = (C, \otimes, I)\) is a traced symmetric monoidal category (TSMC);
- \(F : C \to C\) is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  
  \[ e : FF \triangleleft F : e' \quad \text{Comultiplication} \]
  \[ d : \text{id} \triangleleft F : d' \quad \text{Dereliction} \]
  \[ c : F \otimes F \triangleleft F : c' \quad \text{Contraction} \]
  \[ w : K_I \triangleleft F : w' \quad \text{Weakening} \]

Here \(K_I\) is the constant functor into the monoidal unit \(I\);

- \(U \in C\) is an object (called **reflexive object**), equipped with the following rejections.
  
  \[ j : U \otimes U \triangleleft U : k \]
  \[ I \triangleleft U \]
  \[ u : FU \triangleleft U : v \]

**Monoidal category** \((C, \otimes, I)\)

**String diagrams**

\[
\begin{align*}
A & \xrightarrow{f} B \quad B \xrightarrow{g} C \\
A & \xrightarrow{g \circ f} C
\end{align*}
\]

\[
\begin{align*}
A & \xrightarrow{f} B \quad C \xrightarrow{g} D \\
A \otimes C & \xrightarrow{f \otimes g} B \otimes D
\end{align*}
\]

\[ h \circ (f \otimes g) \]
Defn. (GoI situation [AHS02])

A GoI situation is a triple \((C, F, U)\) where

- \(C = (C, \otimes, I)\) is a traced symmetric monoidal category (TSMC);
- \(F : C \rightarrow C\) is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  
  \[
  \begin{align*}
  e & : FF \otimes F \rightarrow e' \\
  d & : \text{id} \otimes F \rightarrow d' \\
  c & : F \otimes F \otimes F \rightarrow c' \\
  w & : K_I \otimes F \rightarrow w'
  \end{align*}
  \]

  - \(e\): Comultiplication
  - \(d\): Dereliction
  - \(c\): Contraction
  - \(w\): Weakening
  
  Here \(K_I\) is the constant functor into the monoidal unit \(I\);

- \(U \in C\) is an object (called reflexive object), equipped with the following retractions.
  
  \[
  \begin{align*}
  j & : U \otimes U \rightarrow k \\
  I & \rightarrow U \\
  u & : FU \rightarrow U \rightarrow v
  \end{align*}
  \]

That is

\[
\begin{align*}
A \otimes C & \xrightarrow{f} B \otimes C \\
\underbrace{A} & \xrightarrow{\text{tr}(f)} \underbrace{B}
\end{align*}
\]
String Diagram vs. “Pipe Diagram”

* I use two ways of depicting partial functions \( \mathbb{N} \rightarrow \mathbb{N} \)

Pipe diagram

String diagram

In the monoidal category \((\text{Pfn}, +, 0)\)
Traced Sym. Monoidal Category

(Pfn, +, 0)

Category **Pfn** of partial functions

**Obj.** A set \( X \)

**Arr.** A partial function \( f: X \rightarrow Y \) in **Pfn**

\[
\begin{align*}
X & \rightarrow Y \quad \text{in Pfn} \\
X & \rightarrow Y, \text{ partial function}
\end{align*}
\]

is traced symmetric monoidal
Traced Sym. Monoidal Category
(Pfn, +, 0)

\[
\begin{align*}
X + Z & \xrightarrow{f} Y + Z \quad \text{in Pfn} \\
X & \xrightarrow{\text{tr}(f)} Y \quad \text{in Pfn}
\end{align*}
\]

\[f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\
\bot & \text{o.w.} \end{cases} \]

Similar for \(f_{XZ}, f_{ZY}, f_{ZZ}\)

* Execution formula (Girard)
* Partiality is essential (infinite loop)

\[
\text{tr}(f) = f_{XY} \sqcup \left( \bigsqcup_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)
\]
**GoI situation**

A GoI situation is a triple $(C, F, U)$ where

- $C = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : C \to C$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  - $e : FF \otimes F \to F$ : $e'$ (Comultiplication)
  - $d : id \otimes F \to d'$ (Dereliction)
  - $c : F \otimes F \otimes F \to c'$ (Contraction)
  - $w : K_I \otimes F \to w'$ (Weakening)

Here $K_I$ is the constant functor into the monoidal unit $I$;

- $U \in C$ is an object (called reflexive object), equipped with the following retraction.
  - $j : U \otimes U \otimes U \to k$ (Reflexivity)
  - $I \otimes U$ (Identity)
  - $u : FU \otimes U \to v$ (Weakening)

**Traced sym. monoidal cat.**

**Where one can “feedback”**

**Why for GoI?**
\[ [M N] \]

\[ = \]

\[ [M] \quad [N] \]

... in string diagram
**GoI situation**

*Traced sym. monoidal cat.*

*Where one can “feedback”*

*Why for GoI?*

*Leading example: Pfn*
GoI situation

**Defn. (GoI situation [AHS02])**
A GoI situation is a triple \((C, F, U)\) where

- \(C = (C, \otimes, I)\) is a traced symmetric monoidal category (TSMC);
- \(F : C \to C\) is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  - \(e : FF \otimes F \to e'\) Comultiplication
  - \(d : \text{id} \otimes F \to d'\) Dereliction
  - \(c : F \otimes F \otimes F \to c'\) Contraction
  - \(w : K_I \otimes F \to w'\) Weakening

Here \(K_I\) is the constant functor into the monoidal unit \(I\);
- \(U \in C\) is an object (called reflexive object), equipped with the following retractions.
  - \(j : U \otimes U \to k\)
  - \(I \to k\)
  - \(u : FU \otimes U \to v\)

---

**Defn. (Retraction)**
A retraction from \(X\) to \(Y\),

\[ f : X \otimes Y : g, \]

is a pair of arrows

\[ \begin{array}{ccc}
id & \circlearrowleft \ & X \\
\downarrow \ & f \downarrow & \downarrow \ g \\
Y & \circlearrowright & Y \\
\end{array} \]

such that \(g \circ f = \text{id}_X\).

---

**Functor** \(F\)

**For obtaining** \(! : A \to A\)
GoI situation

The reflexive object $U$

Retr. $U \otimes U \triangleleft U$

with

\[ \begin{array}{c}
\downarrow j \\
\downarrow k
\end{array} \quad \begin{array}{c}
j \\
k
\end{array} \quad = \begin{array}{c}
id
\end{array} \]

Defn. (GoI situation [AHS02])

A GoI situation is a triple $(C, F, U)$ where

- $C = (C, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : C \to C$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  
  - $e : FF \triangleright F : e'$, Comultiplication
  - $d : \text{id} \triangleright F : d'$, Dereliction
  - $c : F \otimes F \triangleright F : c'$, Contraction
  - $w : K_I \triangleright F : w'$, Weakening

Here $K_I$ is the constant functor into the monoidal unit $I$;

- $U \in C$ is an object (called reflexive object), equipped with the following retractions.
  
  - $j : U \otimes U \triangleleft U : k$
  - $I \triangleleft U$
  - $u : FU \triangleleft U : v$
**GoI situation**

- The reflexive object $U$
- Why for GoI?
- Example in Pfn:

\[
\begin{align*}
N \in \text{Pfn}, \quad 
N + N \cong N, \\
N \cdot N \cong N
\end{align*}
\]

---

**Defn. (GoI situation [AHS02])**

A GoI situation is a triple $(C, F, U)$ where

- $C = (C, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : C \to C$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  
  \[
  \begin{align*}
  e & : FF \otimes F : e' \\
  d & : \text{id} \otimes F : d' \\
  c & : F \otimes F \otimes F : c' \\
  w & : K_I \otimes F : w'
  \end{align*}
  \]

  Here $K_I$ is the constant functor.

- $U \in C$ is an object (called reflexive object), equipped with the following retractions.
  
  \[
  \begin{align*}
  j & : U \otimes U \otimes U : k \\
  I & : U \\
  u & : FU \otimes U : v
  \end{align*}
  \]
A GoI situation is a triple $(C, F, U)$ where

- $C = (C, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : C \to C$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations):
  - $e : FF \otimes F : e'$
  - $d : id \otimes F : d'$
  - $c : F \otimes F \otimes F : c'$
  - $w : K_I \otimes F : w'$

Here $K_I$ is the constant functor into the monoidal unit $I$.
- $U \in C$ is an object (called reflexive object), equipped with the following retractions:
  - $j : U \otimes U \otimes U : k$
  - $I : U$
  - $u : FU \otimes U : v$

**Example:**

$\mathbf{(Pfn, N \cdot - , N)}$
Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation \((\mathcal{C}, F, U)\), the homset
\[ \mathcal{C}(U, U) \]
carries a canonical LCA structure.

- Applicative str. \(\cdot\)
- ! operator
- Combinators B, C, I, ...

\[ g \cdot f := \text{tr} \left( (U \otimes f) \circ k \circ g \circ j \right) \]

Hasuo (NII, JP)
Summary:
Categorical GoI

**Defn.** (GoI situation [AHS02])
A GoI situation is a triple \((C, F, U)\) where

- \(C = (C, \otimes, I)\) is a traced symmetric monoidal category (TSMC);
- \(F : C \to C\) is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).
  - \(e : FF \triangleleft F : e'\) Comultiplication
  - \(d : \text{id} \triangleleft F : d'\) Dereliction
  - \(c : F \otimes F \triangleleft F : c'\) Contraction
  - \(w : K_I \triangleleft F : w'\) Weakening
- Here \(K_I\) is the constant functor into the monoidal unit \(I\);
- \(U \in C\) is an object (called reflexive object), equipped with the following retractions.
  - \(j : U \otimes U \triangleleft U : k\)
  - \(I \triangleleft U\)
  - \(u : FU \triangleleft U : v\)

**Thm.** ([AHS02])
Given a GoI situation \((C, F, U)\), the homset \(C(U, U)\) carries a canonical LCA structure.
Outline

Coalgebra meets higher-order computation in Geometry of Interaction [Girard, LC'88]

"GoI Animation"

Categorical GoI
[Abramsky, Haghverdi & Scott, MSCS'02]

GoI w/ $T$-branching
[IH & Hoshino, LICS'11]
Why Categorical Generalization?

Examples Other Than \( \text{Pfn} \)

\[ \text{Pfn} \] (partial functions)

\[ \begin{align*}
X &\to Y \text{ in } \text{Pfn} \\
X &\to Y, \text{ partial function} \\
X &\to \text{LY in Sets}
\end{align*} \]

where \( \text{LY} = \{ \bot \} + Y \)

(Potential) non-termination

\[ \begin{align*}
\text{Rel} \text{ (relations)} &\\
X &\to Y \text{ in } \text{Rel} \\
R &\subseteq X \times Y, \text{ relation} \\
X &\to \mathcal{P}Y \text{ in Sets}
\end{align*} \]

where \( \mathcal{P} \) is the powerset monad

Non-determinism

\[ \begin{align*}
\text{DSRel} &\\
X &\to Y \text{ in } \text{DSRel} \\
X &\to \mathcal{D}Y \text{ in Sets}
\end{align*} \]

where \( \mathcal{D}Y = \{ d : Y \to [0, 1] \mid \sum_y d(y) \leq 1 \} \)

Probabilistic branching

\( \text{Kl}(T) \) for different branching monads \( T \)
Different Branching in The GoI Animation

- **Pfn** (partial functions)
  - Pipes can be stuck

- **Rel** (relations)
  - Pipes can branch

- **DSRel**
  - Pipes can branch probabilistically
**Thm.** ([Jacobs,CMCS10])

Given a “branching monad” $T$ on $\textbf{Sets}$, the monoidal category

$$(\mathcal{Kl}(T), +, 0)$$

is

- a *unique decomposition category* [Haghverdi,PhD00], hence is
- a traced symmetric monoidal category.

**Cor.**

$(\mathcal{Kl}(T), +, 0), \mathbb{N} \cdot \bot, \mathbb{N})$ is a GoI situation.

**Monads in** [Hasuo,Jacobs&Sokolova07]

- $\mathcal{Kl}(T)$ is $\mathbb{Cpo}_\bot$-enriched

**Particle-style: trace via the execution formula**

$$\text{tr}(f) = f_{XY} \sqcup \bigg( \bigsqcup_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \bigg)$$
The Categorical GoI Workflow

Branching monad $B$  
Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$  
+ other constructs $\Rightarrow$ "GoI situation" [AHS02]

Categorical GoI [AHS02]
Linear combinatory algebra
Realizability

Fancy monad
Fancy TSMC
Fancy LCA

Model of fancy language
Model for (a variant of) the Selinger-Valiron quantum $\lambda$-calculus
(linear $\lambda$ + prep./Unitary/meas.)
[Hasuo & Hoshino, LICS’11 & APAL’16]

via the quantum branching monad
… with considerable complication :(

$$[[\Gamma \vdash M : \tau]] : [[\Gamma]] \rightarrow ([[\tau] \rightarrow R) \rightarrow R$$

where

$$R = \left\{ \begin{array}{ll}
 p_0 & q_0 \\
p_1 & q_1 \\
\vdots & \vdots
\end{array} \right| p_\alpha, q_\alpha \in [0, 1]$$

Records measurement outcomes
$R$ as a suitable final coalgebra in the realizability category

Realizability

Linear category

Model of fancy language
Challenge: Memorizing Effects

Already w/ nondeterminism!
Challenge: Memorizing Effects

\[(\lambda x. x + x)(3 \sqcup 5)\]

Already w/ nondeterminism!

- Query \((\lambda x. x + x)(3 \sqcup 5)\)
- Query \(x\)
- Answer \(3\) or \(5\)
- Query \(x\)
- Answer \(3\) or \(5\)
- Answer \(3 + 3, 3 + 5, 5 + 3\) or \(5 + 5\)
The GoI Animation

$$[M] = (\mathbb{N} \rightarrow \mathbb{N}, \text{a partial function})$$

= “piping”

\[0 \quad 1 \quad 2 \quad 3 \quad \ldots \quad \text{(countably many)}\]
An Idea

* Let a traversing token rearrange piping!
Memoryful GoI

* Let a traversing token rearrange piping!

PythagoraSwitch, NHK Education
(Created by (another) Masahiko Sato)
Outline

Coalgebra meets higher-order computation in Geometry of Interaction [Girard, LC’88]

"GoI Animation"

Categorical GoI
[Abramsky, Haghverdi & Scott, MSCS’02]

GoI w/ T-branching
[IH & Hoshino, LICS’11]

Memoryful GoI
[Hoshino, Muroya & IH, CSL-LICS’14 & POPL’16]
Memoryful GoI

* Equip piping with internal states, or memory

* not \([3 \sqcup 5]: \mathbb{N} \rightarrow \mathcal{P}\mathbb{N}, \quad q \mapsto \{3, 5\}\]

but a transducer (Mealy machine)

\([3 \sqcup 5]: X \times \mathbb{N} \rightarrow \mathcal{P}(X \times \mathbb{N}), \quad q/3 \xrightarrow{s_1} q/3 \xrightarrow{s_0} q/5 \xrightarrow{s_r} q/5\]

* Not a new idea:

* Slices in GoI for additives [Laurent, TLCA’01]

* Resumption GoI [Abramsky, CONCUR’96]
Memoryful GoI

* We introduce memory in a structured manner...

→

the “traced monoidal category” of transducers

<table>
<thead>
<tr>
<th>Trans(T)</th>
<th>Objects:</th>
<th>sets $A, B, \ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrows:</td>
<td>$A \to B$ in Trans(T)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(X, X \times A \xrightarrow{c} T(X \times B), x_0 \in X)$, $T$-transducer</td>
<td></td>
</tr>
</tbody>
</table>

* with operations like

- composition $\circ$
- tensor $\otimes$
- trace

Hasuo (NII, JP)
Trans($T$) by Coalgebraic Component Calculus

[Barbosa '03][IH & Jacobs '11]

| Trans($T$) | Objects: sets $A, B, \ldots$
| Arrows: | $A \rightarrow B$ in Trans($T$)
| | ($X, X \times A \xrightarrow{c} T(X \times B), x_0 \in X$), $T$-transducer

\[ X \times Y, \stackrel{c \times X}{\rightarrow} T(X \times B) \times Y \]

\[ \xrightarrow{T} T((X \times B) \times Y) \]

\[ \xrightarrow{T \str} T(X \times T(Y \times C)) \]

\[ \xrightarrow{T \str} TT(X \times (Y \times C)) \]

\[ \xrightarrow{\mu^T} T(X \times (Y \times C)) \]

\[ \xrightarrow{\alpha} T((X \times Y) \times C) \]

\[ \otimes \]

Composition $\circ$

Tensor $\otimes$

Trace

Hasuo (NII, JP)
**Trans(\(T\)) by Coalgebraic Component Calculus**

[Barbosa ’03][IH & Jacobs ’11]

<table>
<thead>
<tr>
<th>(\text{Trans}(T))</th>
<th><strong>Objects:</strong></th>
<th>sets (A, B, \ldots)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arrows:</strong></td>
<td>(A \rightarrow B) in (\text{Trans}(T))</td>
<td></td>
</tr>
<tr>
<td>((X, X \times A \xleftarrow{x_0} T(X \times B), x_0 \in X)), (T)-transducer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
A \xrightarrow{(X,c,x)} B \\
B \xrightarrow{(Y,d,y)} C \\
C \xrightarrow{(Z,e,z)} D
\end{align*}
\]

- **composition**
- **tensor**
- **trace**

\* **Trans(\(T\)) is a “category”…**

\* **Fix: quotient modulo behavioral equivalence**

(homomorphisms of \(T\)-transducers) \(\rightarrow\) **resumptions** [Abramsky]

Hasuo (NII, JP)
The Memoryful GoI Framework

* Given:
  * a monad $T$ on Sets, s.t. $\text{Kl}(T)$ is Cppo-enriched
  * an alg. signature $\Sigma$ with algebraic operations on $T$
    [Plotkin & Power]

* For the calculus: $\lambda_e + \text{(alg. opr. from } \Sigma) + \text{(co)products + arith.}$

* We give

$\begin{array}{c}
|\Gamma| \\
\vdash \\
\downarrow \\
\Gamma \vdash M : \tau \\
\end{array}$

in $\text{Trans}(T)$

- Exception $1 + E + (\_)$
  - with 0-ary opr. $\text{raise}_e (e \in E)$
- Nondeterminism $\mathcal{P}$
  - with binary opr. $\sqcup$
- Probability $\mathcal{D}$, where
  $\mathcal{D} X = \{ d: X \rightarrow [0, 1] | \sum_x d(x) \leq 1 \}$
  - with binary opr. $\sqcup_p (p \in [0, 1])$
- Global state $(1 + S \times \_)^S$
  - with $|V|$-ary $\text{lookup}_l$ and unary $\text{update}_{l, v}$
Theorem (Adequacy)

Let \( \vdash M : \text{nat} \). Then, as elem. of \( T(\mathbb{N}) \),

\[
\left( \begin{array}{c}
\vdash M : \text{nat} \\
\mathbb{N}
\end{array} \right)
\]

I

feeding a query

and observing

the outcome

Theorem

The two coincide. (for any suitable \( T \))

Interpretation

\( [\_ ] : \text{EffVal}_{\mathbb{N}}^\Sigma \rightarrow T(\mathbb{N}) \)

(exploiting free conti. \( \Sigma \)-alg.)

Opr. sem.: Plotkin-Power

effect-value. E.g.

\[
| 3 \uplus (5 \uplus \text{div}) | = 3 \uplus 5 \uplus \bot
\]
Our Tool TtT

Developed by Koko Muroya
http://koko-m.github.io/TtT/

TtT (Terms to Transducers)

Enter a term, or type "ex" to select one from 13 examples. [read documents]

This is a simulation tool of the memoryful Go framework.
Implemented by Koko Muroya, using Processing.js v1.4.8 and PEG.js v0.8.0.
Summary

Coalgebra meets higher-order computation in Geometry of Interaction [Girard, LC’88]

“GoI Animation”

GoI w/ $T$-branching [IH & Hoshino, LICS’11]

Categorical GoI [Abramsky, Haghverdi & Scott, MSCS’02]

Memoryful GoI [Hoshino, Muroya & IH, CSL-LICS’14 & POPL’16]
Abstraction in a Theory:

“Categorical Transfer”

Categorical Geometry of Interaction
[Abramsky, Haghverdi, Scott]

Abstract Technique
$T[\_ ]$

Coalgebra
[Jacobs, Rutten, …]

Identify “mathematical essence”

Choose parameter $e_1$

Choose parameter $e_2$

Existing Technique
$T[e_1]$

Novel Technique
$T[e_2]$

Automata, Mealy machines, …

Geometry of Interaction
[Girard]

Gol with nondeterministic, probabilistic, quantum, and other effects

⋯→ Memoryful Geometry of Interaction
[Hoshino, Muroya, Hasuo, …]
Retracing some paths in Process Algebra

Samson Abramsky
Laboratory for the Foundations of Computer Science
University of Edinburgh

1 Introduction

The very existence of the CONCUR conference bears witness to the fact that “concurrency theory” has developed into a subject unto itself, with substantially different emphases and techniques to those prominent elsewhere in the semantics of computation.

Whatever the past merits of this separate development, it seems timely to look for some convergence and unification. In addressing these issues, I have found it instructive to trace some of the received ideas in concurrency back to their origins in the early 1970’s. In particular, I want to focus on a seminal paper by Robin Milner [Mil75], which led in a fairly direct line to his enormously influential work on CCS [Mil80, Mil89]. I will take (to the extreme) the liberty of applying hindsight, and show how some different paths could have been taken, which, it can be argued, lead to a more unified approach to the semantics of computation, and moreover one which may be better suited to modelling today’s concurrent, object-oriented languages, and the type systems and logics required to support such languages.

2 The semantic universe: transducers

Milner’s starting point was the classical automata-theoretic notion of transducers, i.e. structures

\((Q, X, Y, q_0, \delta)\)
Outline

* Abstraction in a Theory I
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02] [Hasuo & Hoshino, LICS’11]
  [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]

* Abstraction in a Theory II
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* Abstraction in a Project
  Introducing ERATO MMSD Project:
  from categorical foundations to automated driving

* Abstraction in an Inspiring Mind
Abstraction in a Theory II
Codensity Bisimulation Games

\[ f^* Q \xrightarrow{\varphi(f^* Q)} \varphi Q \]
\[ (Ff)^* (\varphi Q) \xrightarrow{Ff(\varphi Q)} \varphi Q \]

\[
\begin{array}{|c|c|c|}
\hline
\text{position} & \text{player} & \text{possible moves} \\
\hline
(x, y) \in X^2 & S & Z \subseteq X \text{ s.t. } c(x)(Z) \neq c(y)(Z) \\
\hline
Z \subseteq X & D & (x', y') \in X^2 \text{ s.t. } x' \in Z \land y' \not\in Z \\
\hline
\end{array}
\]

[Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
[Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]
"Categorical Transfer"

Coalgebra [Jacobs, Rutten, …]

Codensity bisimilarity in a fibration

Codensity games

Identify “mathematical essense”

Automata, Mealy machines, …

Choose parameter $e_1$

Choose parameter $e_2$

Abstract Technique $T[\_\_]$  

$T_1 = T[e_1]$  

Novel Technique $T[e_2]$  

Games for bisimulation metric, topology, …

Defn. $T(e) =$ theory of coalgebras in $K$, where $B$ is a monad suited for $e$

Thm. Coalgebraic proof methods (such as (bi)simulations) are valid in $T(e)$ too.

Proof. By the following characterizations in $K$, where $B$ is a monad suited for $e$:

$\begin{aligned}
F X & \xrightarrow{F \text{beh}_c} F Z \\
X & \xrightarrow{\text{beh}_c} Z
\end{aligned}$

system behavior simulation

$\begin{aligned}
F X & \xrightarrow{F f} F Y \\
X & \xrightarrow{f} Y
\end{aligned}$

$A ::= \text{true} | \text{false} | A_1 \land A_2 | \neg A | a_1 < a_2 | \forall x \in \mathbb{N}. A | \forall x \in \mathbb{R}. A$
LaTeX slides
“Categorical Transfer”

Coalgebra
[Jacobs, Rutten, …]

Codensity bisimilarity
in a fibration

Codensity games

Identify
“mathematical
essense”

Automata,
Mealy machines, …

Abstract Technique

Choose parameter $e_1$

$T_1 = T[e_1]$

Bisimulation games

Choose parameter $e_2$

Novel Technique

Games for bisimulation
metric, topology, …

A ::= \text{true} | \text{false} | A_1 \land A_2 | \neg A | a_1 < a_2 | 
\forall x \in \mathbb{N}. A | \forall x \in \mathbb{R}. A

Defn. $T(e) = \text{theory of coalgebras in } K$, where $Be$ is a monad suited for $e$

Thm. Coalgebraic proof methods (such as (bi)simulations) are valid in $T(e)$ too.

Proof. By the following characterizations in $K$, $Be$.

$FX \xrightarrow{F_{beh_c}} FZ$
$X \xrightarrow{\text{beh}_c} Z$

system behavior

$FX \xrightarrow{Ff} FY$
$X \xrightarrow{f} Y$

simulation
Perspectives

* Games ➔ algorithms
  * Infinite state?
    ➔ CEGAR, template-based symbolic presentation, ...

* The roles of **observations** and **indistinguishability** in bisimulation notions, formalized
  * relational
  * metric
  * topological ➔ domain theory!
    * open = observable
      continuous = computable
Outline

* Abstraction in a Theory I
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02] [Hasuo & Hoshino, LICS’11] [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]

* Abstraction in a Theory II
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* Abstraction in a Project
  Introducing ERATO MMSD Project:
  from categorical foundations to automated driving

* Abstraction in an Inspiring Mind
The Power of Abstraction

... and in an inspiring mind
Outline

* Abstraction in a Theory I
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02] [Hasuo & Hoshino, LICS’11] [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]

* Abstraction in a Theory II
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* Abstraction in a Project
  Introducing ERATO MMSD Project:
  from categorical foundations to automated driving

* Abstraction in an Inspiring Mind
On ERATO MMSD

- JST ERATO Project, 2016/10-2022/03.
  Several faculty members,
  15+ researchers, 20+ students, 6 sites

- Our goal: **formal methods** for cyber-physical systems (CPS)
  - Extend **formal methods**, from software to CPS
  - Safety, reliability, V&V (Verification & Validation).
    “Check if a system behaves as expected”
  - **Automated driving** as a strategic target domain.
    Collaboration with U Waterloo: www.autonomoose.net

- Our principle: **broaden** the realm of CPS research
  - **Theory**: abstract mathematical **metatheory**
    → scale out to diverse applications
  - **Practice**: real-world systems (not only toy examples)
Cyber-Physical Systems: Control Theory and Formal Methods/Software Science

- Cyber-Physical System (CPS)
  - “A mechanism that is controlled or monitored by computer-based algorithms, tightly integrated with the Internet and its users” (Wikipedia)
  - Physical plant (continuous) + Digital control (discrete)
  - In US: NSF Key Area of Research (2006-)

- Formal methods: Logical proofs for “correctness” of (discrete) programs
  - Model checking [Pnueli, Clarke, Emerson, Sifakis, …]
  - Theorem Proving (Coq, Agda, …) [Milner, Coquand, Leroy, Voevodsky, …]

- Control Theory: Analysis of continuous dynamics
  - Stability, Lyapunov function, …

- Their similarity is widely recognized
  - e.g. HSCC, one of the main conferences of annual CPS Week
CPS Research, So Far (the V&V Aspect)

- **Challenge:** scalability, esp. for real-world CPSs
  - Require complete understanding of a white-box model
  - Insist on being absolutely sound and correct
  - Little tolerance to uncertainty and noise
    → don’t get along with statistical machine learning
CPS Research: Our Comprehensive Approach

Control Theory

Formal Methods

Mathematical Metathtery

Statistical Machine Learning

Software Engineering
ERATO 蓮尾メタ数理システムデザインプロジェクト
National Institute of Informatics & Japan Science and Technology Agency
$A ::= \text{true} \mid \text{false} \mid A_1 \land A_2 \mid \neg A \mid a_1 < a_2 \\
\forall x \in \ast. \mathbb{N}. A \mid \forall x \in \ast. \mathbb{R}. A$

**Example 1: Coalgebraic Unfolding**
[Hasuo, Urabe, Shimizu et al.]

**Example 2: Nonstandard Transfer**
[Suenaga, Hasuo, Sekine, Kido et al.]
Our Organization

International and multi-disciplinary. “creative chaos”

Group 0 @ NII: Metatheoretical Integration
Leader: Shin-ya Katsumata
Topics: Programming Languages, Formal Semantics, Categorical Models, Mathematical Logic, …

Group 1 @ NII: Heterogeneous Formal Methods
Leader: Ichiro Hasuo
Topics: Automata Theory, Formal Verification, Proof Assistants, Automated Deduction, Runtime Verification

Group 2 @ U Waterloo: Formal Methods in Industry
Leader: Krzysztof Czarnecki
Topics: Automated Driving, Software Engineering, Formal Modeling, Testing, Safe & Explainable AI

Group 3 @ NII: Formal Methods and Intelligence
Leader: Fuyuki Ishikawa
Topics: Software Engineering, Formal Modeling, Testing, Safe & Explainable AI

Kyoto U RIMS Site: Categorical Infrastructure
Leader: Masahito Hasegawa

Kyushu U Site: Optimization for CPS V&V
Leader: Hayato Waki

Osaka U Site: Control Theory for CPS
Leader: Toshimitsu Ushio
Our Categorical Team: ERATO MMSD Group 0

- **Shin-ya Katsumata** (Group Leader, PhD (Edinburgh)):
  fibration, lambda-calculus, categorical semantics, monoidal category, programming language, program verification

- **Jérémy Dubut** (PhD (ENS Paris Saclay, 2017)):
  concurrency, directed topology, topos theory, coalgebra, verification, hybrid system

- **David Sprunger** (PhD (Indiana, 2017)):
  coalgebra, monoidal category, categorical logic, fibration, machine learning

- **Clovis Eberhart** (PhD (U Savoie Mont Blanc, 2018)):
  game semantics, nominal set, verification, machine learning

- **Kenta Cho** (PhD expected (Nijmegen)):
  categorical logic, quantum mechanics & logic, verification, machine learning

- **Soichiro Fujii** (PhD confirmed (Tokyo)):
  Lawvere theory, higher-dimensional category, algebraic effect

- **Natsuki Urabe** (PhD confirmed (Tokyo)):
  coalgebra, concurrency, model checking, game, verification, probabilistic systems

- **Yuichi Komorida** (PhD student (NII/Sokendai)):
  fibration, coalgebra, game

- ... and at **Kyoto Site** (Masahito “Hassei” Hasegawa, Naohiko Hoshino, Koko Munoya, and more)
Control Theory

(Meta-)Theoretical Foundation

Software Science
Formal Verification
Programming Languages

Cyber-Physical Systems, esp. Safety of Automated Driving

Statistical ML/AI

Software Engineering

Group 0

Group 1

Group 3

PJ Status

Defn. $T(e) = \square$, where $B$ is a monad suited for $e$

Thm. Coalgebraic proof methods (such as (bi)simulations) are valid in $T(e)$ too.

Proof. By the following characterizations in $K(B)$.

System behavior simulation

$F\rightarrow X \leftarrow \int \dot{x}_i dt \rightarrow x_i$

$V(x_i)$
Outline

* **Abstraction in a Theory I**
  Categorical GoI (geometry of interaction)
  [Abramsky, Haghverdi & Scott, MSCS’02] [Hasuo & Hoshino, LICS’11] [Hoshino, Muroya & Hasuo, CSL-LICS’14]
  [Muroya, Hoshino & Hasuo, POPL’16] [Hasuo & Hoshino, APAL’17]

* **Abstraction in a Theory II**
  Codensity Bisimulation Games
  [Katsumata & Sato, CALCO’15] [Sprunger, Katsumata, Dubut & Hasuo, CMCS’18]
  [Komorida, Katsumata, Hu, Klin & Hasuo, submitted (soon on arXiv)]

* **Abstraction in a Project**
  Introducing **ERATO MMSD** Project:
  from categorical foundations to automated driving

* **Abstraction in an Inspiring Mind**
The Power of Abstraction

To come any close to an inspiring mind...

individual efforts

... and our collective effort