Memoryful GoI with recursion

Koko Muroya (Univ. Tokyo)  Naohiko Hoshino (Kyoto Univ.)  Ichiro Hasuo (Univ. Tokyo)
Memoryful Golf with recursion

Koko Muoya (Univ. Tokyo)  Naohiko Hoshino (Kyoto Univ.)  Ichiro Hasuo (Univ. Tokyo)
Memoryful GoI
with recursion

Koko Muroya  Naohiko Hoshino  Ichiro Hasuo
(Univ. Tokyo)  (Kyoto Univ.)  (Univ. Tokyo)
Memoryful GoI  [Hoshino, —, Hasuo ’14]

- **effectful terms**: 
  \[(\lambda x : \text{nat}. \, x + x) \, (3 \downarrow 5) : \text{nat}\]

- sound translation
  - based on Geometry of Interaction
  - via coalgebraic component calculus

- transducers
Geometry of Interaction (GoI)

- semantics of linear logic proofs [Girard '89],
- functional programming languages
- token machine representation [Mackie '95]
- compilation techniques and implementations [Mackie '95] [Pinto '01] [Ghica '07]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x)\ 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) \ 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) \ 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) \, 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) 1\]
Geometry of Interaction (GoI)

- token machine representation of GoI interpretation

\[(\lambda x. x + x) 1\]
Memoryful GoI [Hoshino, —, Hasuo ’14]

- **Effectful terms**
  
  $$(\lambda x : \text{nat}. x + x) (3 \sqcup 5) : \text{nat}$$

- Sound translation
  - based on Geometry of Interaction
  - via coalgebraic component calculus

- **Transducers**
Memoryful GoI — Input

CBV $\lambda$-terms with algebraic effects

- nondeterministic choice $M \uplus N$
- probabilistic choice $M \uplus_p N$
- actions on global states $\text{update}_{l,v}(M)$

algebraic operations [Plotkin, Power ’01]

effectful terms

transducers
stream transducers (Mealy machines)

\[(X, c: X \times A \to T(X \times B), x_0 \in X): A \to B\]
Memoryful GoI — Output

stream transducers (Mealy machines)

\[(X, c: X \times A \rightarrow T(X \times B), \, x_0 \in X): A \rightarrow B\]

\(\text{Res}(T)\)

- objects: sets
- arrows: transducers modulo behavioral equivalence

\[[(X, c: X \times A \rightarrow T(X \times B), \, x_0 \in X)]_{\sim}: A \rightarrow B\]
Memoryful GoI — Output

stream transducers (Mealy machines)

\[(X, c: X \times A \rightarrow T(X \times B), x_0 \in X): A \rightarrow B\]

\[T = \mathcal{P} \quad (x_0, a_0) \longmapsto \{(x_1, b_1), (x_2, b_2)\}\]

\[T = \mathcal{D} \quad (x_0, a_0) \longmapsto \left[\begin{array}{c}
(x_1, b_1) \mapsto 1/4, \\
(x_2, b_2) \mapsto 3/4
\end{array}\right]\]
Memoryful GoI [Hoshino, —, Hasuo ’14]

- **Effectful terms**
  - $(\lambda x : \text{nat}. x + x) (3 \square 5) : \text{nat}$

- Sound translation
  - based on Geometry of Interaction
  - via coalgebraic component calculus
    - [Barbosa ’03] [Hasuo, Jacobs ’11]

- **Transducers**
  - composition operators for software components
  - (many-sorted) process calculus
Memoryful GoI — Translation

Def. (component calculus)

\[ C \circ D \quad C \oplus D \quad \text{Tr}(C) \quad F(C) \quad \bar{\alpha}(\{C_i\}_{i \in I}) \]
Memoryful GoI — Translation

Def. (component calculus)

\[ C \circ D \]

\[
\left( Y \times B \xrightarrow{d} T(Y \times C) \right) \circ \left( X \times A \xrightarrow{\xi} T(X \times B) \right) = \left( X \times Y, \right.
\]
\[
\left. (x_0, y_0) \in X \times Y \right)
\]

\[ C \boxdot D \]

\[
\left( X \times A \xrightarrow{\xi} T(X \times B) \right) \boxdot \left( Y \times C \xrightarrow{d} T(Y \times D) \right) = \left( X \times Y, \right.
\]
\[
\left. (x_0, y_0) \in X \times Y \right)
\]
Memoryful GoI — Translation

Def. (component calculus)

\[
\begin{align*}
\text{Tr}(C) & \quad \text{trace operator} \\
F(C) & \quad \text{countable copy operator} \\
\overline{\alpha}\left(\{C_i\}_{i \in I}\right) & \quad \text{lifted algebraic operation}
\end{align*}
\]

\[
\text{Tr}\left(\begin{array}{c}
B \\
\hline \\
A \\
\end{array}\right) \begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

\[
\begin{array}{c}
B \\
\hline \\
A \\
\end{array}
\begin{array}{c}
C \\
\hline \\
C \\
\end{array}
\]

(\(\alpha\): I-ary algebraic operation)
Memoryful GoI — Translation

Def. (component calculus)

\[ \text{Tr}(C) \]

\[ \text{F}(C) \]

\[ \bar{\alpha} \left( \{ C_i \}_{i \in I} \right) \]

\[ \lambda x. x + x \]

\[ (\alpha: I\text{-ary algebraic operation}) \]
Memoryful GoI — Translation

Def. (component calculus)

\[
\text{Tr}(C) \quad F(C) \quad \overline{\alpha}(\{C_i\}_{i \in I})
\]

- \(\text{Tr}(C)\): trace operator
- \(F(C)\): countable copy operator
- \(\overline{\alpha}(\{C_i\}_{i \in I})\): lifted algebraic operation

\[
\begin{align*}
\text{Tr}(c) & \mapsto B \quad F(c) & \mapsto \text{countable copy operator} \\
\overline{\alpha}(\{C_i\}_{i \in I}) & \mapsto \text{lifted algebraic operation}
\end{align*}
\]

(\(\alpha\): \(I\)-ary algebraic operation)

Muroya (U. Tokyo)
Memoryful GoI — Translation

Def. (translation(\(\Gamma \vdash M : \tau\)))

For a type judgement \((\Gamma \vdash M : \tau)\) \((\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n)\)

we inductively define

\[
(\Gamma \vdash M : \tau) = (\Gamma \vdash M : \tau) \cdot (\Gamma \vdash M : \tau) \cdot \cdots \cdot (\Gamma \vdash M : \tau).
\]
Memoryful GoI — Translation

Def. (translation $(\Gamma \vdash M : \tau)$)

$(\Gamma \vdash \lambda x : \sigma. M : \sigma \rightarrow \tau) = h$

$(\Gamma \vdash M \ N : \tau) = (\Gamma \vdash M : \sigma \rightarrow \tau) \ (\Gamma \vdash N : \sigma) \ (\Gamma, x : \sigma \vdash M : \tau)$
Def. (translation$(\Gamma \vdash M : \tau)$)

\[ (\Gamma \vdash x_i : \tau_i) = \]

\[ (\Gamma \vdash \text{op}^+(M_1, \ldots, M_{\text{ar}(\text{op})}) : \tau) = \]

\[ (\Gamma \vdash \text{op}^0() : \tau) = \]
Memoryful GoI — Translation

### Def. (translation(Γ ⊢ M : τ))

1. \((Γ ⊢ n : \text{nat}) = \)

2. \((Γ, x : \text{nat}, y : \text{nat} ⊢ x + y : \text{nat}) = \)

   (if \(x \neq y\))

3. \((Γ, x : \text{nat} ⊢ x + x : \text{nat}) = \)
Memoryful GoI — Translation

**Theorem III.3** (soundness of \([-\cdot]\)). *For closed terms* $M$ and $N$ of the base type nat, $\vdash M = N : \text{nat}$ implies $\langle\vdash M : \text{nat}\rangle \simeq \langle\vdash N : \text{nat}\rangle$.

- (almost full fragment of) Moggi’s equations for computational lambda-calculus
- equations for algebraic operations

\[
M \sqcup M = M
\]

\[
E[\text{opr}(M_1, \ldots, M_n)] = \text{opr}(E[M_1], \ldots, E[M_n])
\]

\[
(\lambda x. M) (N_1 \sqcup N_2) = (\lambda x. M) N_1 \sqcup (\lambda x. M) N_2
\]
Memoryful GoI [Hoshino, —, Hasuo ’14]

- **Effectful terms**
  - \((\lambda x : \text{nat}. x + x)(3 \triangleleft 5) : \text{nat}\)

- **Sound translation**
  - Based on Geometry of Interaction
  - Via coalgebraic component calculus

- **Transducers**

![Diagram of transducers and memoryful GoI](image)
Memoryful GoI with recursion

- **effectful terms**
  - $(\lambda x : \text{nat}. x + x)(3 \Box 5) : \text{nat}$

- **translation**
  - based on Geometry of Interaction
  - via coalgebraic component calculus

---

**transducers**
Memoryful GoI with recursion

- **Effectful terms**
- **Recursion**

\[(\lambda x : \text{nat.} \ x + x) (3 \equiv 5) : \text{nat}\]

- **Translation**
  - Based on Geometry of Interaction
  - Via coalgebraic component calculus

**Transducers**

- Calculational proofs of meta-theoretical properties
- Executed transducers
- A prototype Haskell program

**Motivation**

- Insight into higher-order computation with effects
- Automation of transducer translation
- Meta-theoretical properties
- Haskell program

**Geometry of Interaction (GoI)**

- Helps us to scrutinize step-by-step interactions
- Derived and executed transducers

**TtT Compiler**

- Transducers, Derived and Executed
- Parametric in a monad
- Computation of a state space
- Transducer translation is based on denotational semantics

**Simulation results**

- Simulation results for simulating a token machine
- A transducer is given by a Haskell program

**Examples**

- Example of the limitations of memoryless GoI
- Example of the limitations of memoryful GoI
Memoryful GoI with recursion

effectful terms

(\lambda x : \text{nat}. x + x)(3 \sqcup 5) : \text{nat}

translation
- based on Geometry of Interaction
- via coalgebraic component calculus

transducers

\emph{fix}(F) = F(F(F(\cdots))))
Memoryful GoI with recursion

effectful terms

\[(\lambda x : \text{nat}. \ x + x) (3 \sqcup 5) : \text{nat}\]

translating
- based on Geometry of Interaction
- via coalgebraic component calculus

transducers

\[
\text{fix}(F) = F(F(F(F(\cdots))))
\]
Memoryful GoI with recursion

**effectful terms**

\[(\lambda x : \text{nat}. x + x) (3 \sqcup 5) : \text{nat}\]

**translation**
- based on Geometry of Interaction
- via coalgebraic component calculus

**transducers**

\[\text{fix}(F) = F(F(F(F(\cdots))))\]
Memoryful GoI with recursion

- **effectful terms**
  - $(\lambda x : \text{nat}. x + x) (3 \sqcup 5) : \text{nat}$

- **translation**
  - based on Geometry of Interaction
  - via **extended** coalgebraic component calculus

- **transducers**

  $\text{fix}(F) = F(F(F(F(\cdots))))$

  $\lambda x. x + x$

  $1$

  $c$

  $\ldots$
Memoryful GoI with recursion

Def. (component calculus over transducers)

\[ \mathcal{C} \circ \mathcal{D} \]

\[ \mathcal{C} \boxplus \mathcal{D} \]

\[ \text{Tr}(\mathcal{C}) \]

\[ F(\mathcal{C}) \]

\[ \overline{\alpha}(\{\mathcal{C}_i\}_{i \in I}) \]

- Sequential composition
- Parallel composition
- Trace operator
- Countable copy operator
- Lifted algebraic operation

(\(\alpha\): \(I\)-ary algebraic operation)
Memoryful GoI with recursion

Def. (extended component calculus)

\[
\bigoplus_{i \in I} C_i
\]

\[
\begin{array}{ccc}
B_1 & B_2 & \cdots \\
\downarrow & \downarrow & \downarrow \\
c_1 & c_2 & \cdots \\
\downarrow & \downarrow & \downarrow \\
A_1 & A_2 & \cdots
\end{array}
\]

“fixed point” operator

\[
\text{Fix}(C)
\]

\[
\begin{array}{c}
\text{Fix}
\end{array}
\begin{array}{c}
\bigoplus_{i \in I} C_i
\end{array}
\]

\[
\begin{array}{ccc}
A & N \times A & A \\
\downarrow & \downarrow & \downarrow \\
c & & \cdots
\end{array}
\]

\[
\begin{array}{ccc}
A & N \times A & A \\
\downarrow & \downarrow & \downarrow \\
c & & \cdots
\end{array}
\]

\[
\begin{array}{ccc}
A & N \times A & A \\
\downarrow & \downarrow & \downarrow \\
c & & \cdots
\end{array}
\]

Muroya (U. Tokyo)
Memoryful GoI with recursion

Def. (extended component calculus)

\[
\bigoplus_{i \in I} C_i
\]

countable parallel composition

\[
\text{Fix}(C)
\]

“fixed point” operator

\[
\text{Fix}(A \times A) \quad \text{N \times A}
\]
“Fixed point” operator

Lemma (Fix as a fixed point operator)

\[
\text{Fix}(c) \quad \text{satisfies} \quad \text{Fix}(\text{Fix}(c)) = \text{Fix}(c).
\]
"Fixed point" operator

Lem. (two styles of "implementation")

\[ \text{Fix}(c) \]

\[ \Rightarrow \]

\[ \text{Girard style} \]

\[ \Rightarrow \]

\[ \text{Mackie style} \]
“Fixed point” operator

Lem. (domain-theoretic characterization of $\text{Fix}$)

Under the assumption that

- $\textbf{Set}_T$ is a $\textbf{Cppo}$-enriched category with $\textbf{Cppo}$-enriched (countable) cotu- plings

- compositions $\circ_T$ of $\textbf{Set}_T$ is strict in the restricted form: $f \circ_T \bot = \bot$ and $\bot \circ_T (\eta_Y \circ g) = \bot$ hold for any $f : X \to TY$ and $g : X \to Y$ in $\textbf{Set}$

- premonoidal structures $X \otimes -$ and $- \otimes X$ of $\textbf{Set}_T$ is locally continuous and strict for any $X$ in $\textbf{Set}$

it holds that:
“Fixed point” operator

Lem. (domain-theoretic characterization of \( \text{Fix} \))

\[
\text{Fix} \left( A \xrightarrow{N \times A} A \right) = \text{is a supremum of an } \omega\text{-chain}
\]

where \( (X, c: X \times A \to T(X \times B), x_0 \in X) \preceq (Y, c: Y \times A \to T(Y \times B), y_0 \in Y) \)

\[
\overset{\text{def}}{\iff} X = Y \land x = y \land c \subseteq d \text{ in } \text{Set}_T(X \times A, X \times B)
\]
Memoryful GoI with recursion

effectful terms

\[(\lambda x : \text{nat}. x + x)(3 \Box 5) : \text{nat}\]

translation

- based on Geometry of Interaction
- via extended coalgebraic component calculus

transducers

\[\begin{array}{c}
\psi \\
\phi \\
(3)_{\mathbb{N}, \mathbb{N}}(5) \\
\end{array}\]

\[\begin{array}{c}
h \\
\text{sum} \\
\text{cpy} \\
\end{array}\]
Memoryful GoI — Translation

Def. \((\text{translation}(\Gamma \vdash M : \tau))\)

For a type judgement \((\Gamma \vdash M : \tau)\) \((\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n)\)

we inductively define

\[
(\Gamma \vdash M : \tau) = \begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\vdots \\
N \\
N \\
\vdots \\
N
\end{array}
\end{array}
\end{array}.
\]
Memoryful GoI — Translation

Def. (translation(Γ ⊢ M : τ))

(Γ ⊢ λx : σ. M : σ → τ) = h

(Γ ⊢ M N : τ) =

\[ \lambda x. x + x \]
Memoryful GoI — Translation

Def. (translation($\Gamma \vdash M : \tau$))

$\langle \Gamma \vdash x_i : \tau_i \rangle = \begin{array}{c}
\hline
h \\
\vdots \\
w \\
w' \\
\vdots \\
\hline
\end{array}$

$\langle \Gamma \vdash \text{op}^+(M_1, \ldots, M_{\text{ar}(\text{op})}) : \tau \rangle = \begin{array}{c}
\hline
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\vdots \\
\hline
\langle \Gamma \vdash M_1 : \tau \rangle \\
\langle \Gamma \vdash M_2 : \tau \rangle \\
\langle \Gamma \vdash M_3 : \tau \rangle \\
\langle \Gamma \vdash M_{\text{ar}(\text{op})} : \tau \rangle \\
\vdots \\
\vdots \\
\end{array}$

$\langle \Gamma \vdash \text{op}^0() : \tau \rangle = \begin{array}{c}
\hline
\text{op}^0 \\
w \\
w' \\
\vdots \\
\hline
\end{array}$

Muroya (U. Tokyo)
Memoryful GoI — Translation

Def. (translation ($\Gamma \vdash M : \tau$))

$\Gamma \vdash n : \text{nat}$

$\Gamma, x : \text{nat}, y : \text{nat} \vdash x + y : \text{nat}$ (if $x \neq y$)

$\Gamma, x : \text{nat} \vdash x + x : \text{nat}$
Memoryful GoI with recursion

Def. (translation(Γ ⊢ M : τ))

(Γ ⊢ rec(f : σ → τ, x : σ). M : σ → τ) =

Mackie style

\[
\begin{align*}
&\text{c'} \\
&\text{u} \\
&\text{d} \\
&\phi \\
&\psi \\
&\text{v} \\
&\text{c} \\
&\text{d}' \\
&\ldots
\end{align*}
\]
Memoryful GoI — Translation

**Theorem III.3** (soundness of \([-\cdot-]\)). For closed terms \(M\) and \(N\) of the base type \(\text{nat}\), \(\vdash M = N : \text{nat}\) implies \(\langle \vdash M : \text{nat} \rangle \simeq \langle \vdash N : \text{nat} \rangle\).

- (almost full fragment of) Moggi’s equations for computational lambda-calculus
- equations for algebraic operations

\[
M |- M = M \\
E[\text{opr}(M_1, \ldots, M_n)] = \text{opr}(E[M_1], \ldots, E[M_n]) \\
(\lambda x. M) (N_1 |- N_2) = (\lambda x. M) N_1 |- (\lambda x. M) N_2
\]
Memoryful GoI recursion

sound translation
• based on Geometry of Interaction
• via extended coalgebraic component calculus

transducers

effectful terms

(\lambda x : \text{nat}. x + x)(3 \parallel 5) : \text{nat}
Examples

\(((\lambda x.x) (3 \sqcap_{0.4} 5))\) =

Muroya (U. Tokyo)
Examples

\( \lambda x . x \) = 3 ∥ 0.4 5 = \)

\( \lambda x . x ) (3 \big\|_{0.4} 5) = \)
Examples

\(((\text{rec}(f,x).fx)\;0)\) =
Examples

\[
\text{case(inl}_{1,1}(\ast) \sqcup \text{inr}_{1,1}(\ast), \ y.1, \ z.f\ x)\\
\]

\[
\|(\text{rec}(f, x).\text{if true \sqcup false then 1 else } f\ x)\ 0\| =
\]
Memoryful Geometry of Interaction

Effectful terms

\((\lambda x : \text{nat}. x + x)(3 \sqcup 5) : \text{nat}\)

Sound (& adequate) translation

- based on Geometry of Interaction
- via extended coalgebraic component calculus

Transducers

\(\Gamma \vdash M : \tau\)
Memoryful GoI recursion

- **effectful terms**
  - \((\lambda x : \text{nat}. x + x)(3 \sqcup 5) : \text{nat}\)

- sound (& adequate) translation
  - based on Geometry of Interaction
  - via **extended** coalgebraic component calculus

- **transducers**