Memoryful GoI with Recursion

Koko Muroya  
University of Tokyo  
Email: muroykk@is.s.u-tokyo.ac.jp

Naohiko Hoshino  
RIMS, Kyoto University  
Email: naophiko@kurims.kyoto-u.ac.jp

Ichiro Hasuo  
University of Tokyo  
Email: ichiro@is.s.u-tokyo.ac.jp

Abstract—In this preliminary report we extend our framework of memoryful Geometry of Interaction (mGoI) [Hoshino, Muroya & Hasuo, CSL-LICS 2014] by recursion. The mGoI framework provides a sound translation from λ-terms to transducers; notably it accommodates algebraic effects introduced by Plotkin and Power; and the translation, defined in terms of a coalgebraic component calculus, is extracted from categorical semantics (hence correct-by-construction). In our current extension, recursion is additionally accommodated by introducing a new “fixed point” operator in the component calculus.

I. GoI INTERPRETATION

Girard’s Geometry of Interaction (GoI) [1] is originally introduced as semantics of linear logic proofs and, via the Curry-Howard correspondence (and the Girard transformation), it has been successfully applied to denotational semantics of higher-order functional programs. The resulting semantics give so-called “GoI interpretation” of programs; one of its notable features is that GoI interpretation of function application is given by interactions of a function and its arguments.

Many representations of GoI interpretation have been studied so far: the original one by elements of a C*-algebra (or a dynamic algebra) that can be seen as “valid paths” on type derivation trees [1]; the one by token machines [2]; and the categorical one by arrows in a traced symmetric monoidal category [3]. The second one by token machines plays an important role in bridging the gap between mathematical interpretation and low-level implementation. Namely it provides techniques of compilation and high-level synthesis, such as a compilation technique [2] and a high-level synthesis technique [4] that enables hardware acceleration of programs by FPGA.

We wish to contribute to this sequence of work by enable GoI interpretation to accommodate computational effects.

II. MEMORYFUL GOI

In the previous work [5] we developed the memoryful GoI (mGoI) framework that extends GoI interpretation of programs. Notably it accommodates algebraic effects—computational effects with algebraic operations as a syntactic interface, introduced by Plotkin and Power [6], [7]. Their examples include: nondeterminism, with a nondeterministic choice operation ⊗ as an algebraic operation; probability, with a probabilistic choice operation Lp for any p ∈ [0, 1]; and global states, with operations lookup and update.

A. Component Calculus over Transducers

The mGoI interpretation of a program is given by T-transducers—an extension of Mealy machines (or sequential machines) by effects specified by a monad T. Here we follow [8] and model algebraic effects by a monad T on the category Set of sets and functions.

Definition II.1 (T-transducers [5, Definition 4.1]). For sets A and B, a T-transducer (X, c, x) from A to B (written as (X, c, x): A → B) consists of a set X, a function c: X × A → T(X × B) and an element x ∈ X.

A T-transducer (X, c, x): A → B can be seen as an (T-effectful) transition function c with input A, output B, a set of internal states X and an initial state x. It shall be presented, in diagrams, as in Fig. 1.

In the mGoI framework, T-transducers are combined via a component calculus over them. It consists of primitive T-transducers (as basic building blocks) and the following operators on T-transducers: a) sequential composition ⊗; b) binary parallel composition ⨿; c) the trace operator Tr; d) the countable copy operator F; e) the operator Tr for each algebraic operation α on T.1 On top of these operators an auxiliary operator is defined: f) binary application •.2 The last is a well-known construction called parallel composition and hiding and is used here to translate function application. In Fig. 2 are graphical presentation of these operators; we refer readers to [5] for their precise definitions.

1We identify algebraic operations with their interpretations, as in [6].
2Binary application • presented here is an adaptation of that in [5].
B. Translation from Terms to Transducers

In our mGoI framework, to be precise, the provided interpretation \( \langle - \rangle \) is from a type judgment \( \Gamma \vdash M : \tau \) to a \( T \)-transducer

\[
\langle \Gamma \vdash M : \tau \rangle : \prod_{i=0}^{m} \mathbb{N} \to \prod_{i=0}^{m} \mathbb{N}.
\]

Here \( \mathbb{N} \) is the set of natural numbers. The interpretation is defined inductively on the type derivations, using the component calculus introduced in the above.

In [9] we presented a prototype implementation—\( TrT \), short for “Terms to Transducers”—of the translation \( \langle - \rangle \). Given a closed term \( M \) of type \( \tau \), the tool first generates a Haskell program that implements a transition function of the \( T \)-transducer \( \langle \vdash M : \tau \rangle \); and then it produces a simulation result of the execution of the transducer. We believe that the tool serves as a first step towards high-level synthesis (that translates a \( \lambda \)-term to hardware design like on FPGA)—much like in [4] but now with algebraic effects.

Some further comments are in order on: 1) a categorical model behind the translation \( \langle - \rangle \); and 2) prospects of accommodating recursion. In fact the translation \( \langle - \rangle \) is extracted from a categorical model \( \text{Per}_\Phi \)—a Kleisli category of a strong monad \( \Phi \) on a cartesian closed category \( \text{Per} \)—built on \( T \)-transducers and the component calculus. It is an instance of the class of models, that is provided in [6], of the Moggi’s computational \( \lambda \)-calculus [8] with algebraic operations and arithmetic primitives. In [6] a class of models that accommodates recursion is studied as well; the key is a fixed point operator on a categorical model. However it was not clear, at the time of writing our previous paper [5], how to obtain a fixed point operator on the categorical model \( \text{Per}_\Phi \) and extend the translation \( \langle - \rangle \) to recursion.

III. Translation of Recursion

Here we report our ongoing work that introduces recursion to the mGoI framework in [5].

A. Extension of Component Calculus and Translation

Our approach is to extend the component calculus shown in Fig. 2: binary parallel composition \( \Box \) is extended to a countable one \( \Box_i \); and on top of the calculus, a “fixed point” operator \( \text{Fix} \) is introduced. It is presented in Fig. 3.

<table>
<thead>
<tr>
<th>A</th>
<th>( \mathbb{N} \times A )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{N} \times A )</td>
<td>A</td>
</tr>
</tbody>
</table>

\[ \text{Fix}(X, c, x) : A \to A. \]

Here one dashed box means countable duplication of a component.

It indeed gives a fixed point with respect to binary application \( \bullet \).

Lemma III.1. Let \( (X, c, x) : A + \mathbb{N} \times A \to A + \mathbb{N} \times A \) be a \( T \)-transducer. The \( T \)-transducer \( \text{Fix}(X, c, x) : A \to A \) satisfies the behavioral equivalence

\[
(X, c, x) \cdot \text{Fix}(X, c, x) \simeq \text{Fix}(X, c, x).
\]

Here the behavioral equivalence \( \simeq \) [5, Def. 5.2] is used for (equational) reasoning on \( T \)-transducers; it enables us to abstract away from internal state spaces of \( T \)-transducers.

With this extension of the component calculus the translation \( \langle - \rangle \) can be extended to recursion: the following definition is precisely what is given in [5], except recursion that is new.

Definition III.2 (translation \( \langle - \rangle \)). For each type judgment \( \Gamma \vdash M : \tau \) where \( \Gamma = x_1 : \tau_1, \ldots, x_m : \tau_m \), we inductively define a \( T \)-transducer

\[
\langle \Gamma \vdash M : \tau \rangle = \prod_{i=0}^{m} \mathbb{N} \to \prod_{i=0}^{m} \mathbb{N}
\]

as in Fig. 4. In Fig. 4, \( \alpha \) is an \( n \)-ary algebraic operation on \( T \) that is the interpretation of \( op \); and all the \( T \)-transducers other than those in the form \( \langle \vdash M : \tau \rangle \) are primitives (see [5] for their definitions).

The translation \( \langle - \rangle \) is sound with respect to the equational theory given in [6]. The latter is (an almost full fragment of) the Moggi’s equational theory of computational \( \lambda \)-calculus, extended by algebraic operations, arithmetic primitives and recursion.

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

For simplicity we have restricted to algebraic operations with finite arities; accommodating countable arities is straightforward (much like in [5], [10]). On top of soundness, we expect adequacy to hold too, against the operational semantics in [6].

Extension of our implementation tool \( TtT \) with recursion is future work, too.

B. The Categorical Model

The translation \( \langle - \rangle \) extended with recursion (Def. III.2) is backed up by a categorical model, too—this fact underlies Thm. III.3. Starting from the model \( \text{Per}_\Phi \) used in [5], we use its modification \( \text{Per}_\Phi' \) (whose details we do not describe here); then we can show that the construction \( \text{Fix} \) in Lem. III.1 indeed yields a (categorical) fixed point operator in \( \text{Per}_\Phi' \). In showing the latter, the following is a key technical lemma.

Lemma III.4. Let \( \text{Cppo} \) be the category of pointed \( \omega \)-cpos (i.e. with the least element \( \bot \)) and continuous maps. Assume that the Kleisli category \( \text{Set}_T \) satisfies the following:

- it is \( \text{Cppo} \)-enriched (with a partial order \( \sqsubseteq \)) and has \( \text{Cppo} \)-enriched (countable) cotupling;
- its compositions \( \circ_T \) is strict, in the restricted sense as in [5, Lem. 4.3];
its premonoidal structures \( X \otimes - , - \otimes X \) are locally continuous and strict, for any \( X \in \text{Set} \).

The Cppo-enrichment of \( \text{Set}_T \) induces the following \( \omega \)-cpo structure on \( T \)-transducers. A partial order \( \leq \) on \( T \)-transducers \( (X, c, x), (Y, d, y) : A \rightarrow B \) is defined by

\[
(X, c, x) \leq (Y, d, y) \overset{\text{def}}{\Rightarrow} X = Y \wedge x = y \wedge c \sqsubseteq d.
\]

Minimal \( T \)-transducers with respect to \( \leq \) are given by \((Z, \perp, z)\) for any set \( Z \). Now for a \( T \)-transducer \((X, c, x) : A + N \times A \rightarrow A + N \times A\), the \( T \)-transducer \( \text{Fix}(X, c, x) : A \rightarrow A \) is a supremum of the following \( \omega \)-chain.

ACKNOWLEDGMENT

The authors are supported by the JSPS-INRIA Bilateral Joint Research Project CRECOGI; K.M. and I.H. are supported by Grants-in-Aid No. 24680001 & 15K11984, JSPS; and N.H. is supported by Grant-in-Aid No. 26730004, JSPS.

REFERENCES


\[
\begin{align*}
(\Gamma \vdash x : \sigma, M : \sigma \Rightarrow \tau) &= \\
(\Gamma \vdash x : \sigma, y : \sigma \vdash x + y : \sigma) &= \\
(\Gamma \vdash \text{op}(M_1, \ldots, M_n) : \tau) &= \\
(\Gamma \vdash \text{rec}(f : \sigma \Rightarrow \tau, x : \sigma, M) : \sigma \Rightarrow \tau) &= \\
\end{align*}
\]

Fig. 4. inductive definition of the translation \( \langle - \rangle \)