Semantics for first-order affine inductive datatypes via slice categories

Vladimir Zamdzhiev

Inria/LORIA
Nancy, France

Concurrent Games Cafe
02.04.2021

Introduction

- Inductive datatypes are an important programming concept.
  - Data structures such as natural numbers, lists, etc.; manipulate variable-sized data.
- Affine types: non-linear variables may be copied; all variables may be discarded.
- How to interpret an affine type system with inductive datatypes?
  - **Standard approach:** Find a subcategory in which to interpret the *values* of the language and where the tensor unit is terminal.
  - **Problem:** What if we cannot find such a subcategory?
- **This talk:** a new semantic technique for the construction of discarding maps for first-order inductive datatypes.
- Paper presented at CMCS 2020 (Coalgebraic Methods in Computer Science).

Overview

- Let $\mathbf{C}$ be a categorical model with sufficient structure to interpret inductive types.
- Open types are interpreted as ω-cocontinuous functors $[\![\Theta \vdash A]\!] : \mathbf{C}^{|\Theta|} \to \mathbf{C}$.
- Closed types are interpreted as objects $[\![A]\!] \in \mathrm{Ob}(\mathbf{C})$.
- Consider the slice category $\mathbf{C}/I$.
- $\mathbf{C}/I$ inherits sufficient structure from $\mathbf{C}$ to interpret inductive types.
- Open types can *also* be interpreted as ω-cocontinuous functors $\llparenthesis \Theta \vdash A \rrparenthesis : (\mathbf{C}/I)^{|\Theta|} \to \mathbf{C}/I$.
- The *affine* interpretation $\llparenthesis - \rrparenthesis$ satisfies some coherence properties w.r.t $[\![-]\!]$.
- It follows $\llparenthesis A \rrparenthesis = ([\![A]\!], \diamond_A : [\![A]\!] \to I)$.
- One can also show that the interpretations of values are discardable.

Syntax

Type Variables	X, Y, Z		
Term Variables	x, y, b, u		
Atomic Types	$\mathbf{A} \in \mathcal{A}$		
Types	A, B, C	$::=$	$X \mid I \mid \mathbf{A} \mid A + B \mid A \otimes B \mid \mu X.A$
Terms	M, N	$::=$	$\textbf{new unit } u \mid \textbf{discard } x \mid M; N \mid \textbf{skip} \mid$ $\textbf{while } b \textbf{ do } M \mid x = \textbf{left}_{A,B} M \mid x = \textbf{right}_{A,B} M \mid$ $\textbf{case } y \textbf{ of } \{\textbf{left } x_1 \to M \mid \textbf{right } x_2 \to N\} \mid$ $x = (x_1, x_2) \mid (x_1, x_2) = x \mid y = \textbf{fold } x \mid y = \textbf{unfold } x$
Type contexts	Θ	$::=$	$X_1, X_2, \ldots, X_n$
Variable contexts	Γ, Σ	$::=$	$x_1 : A_1, \ldots, x_n : A_n$
Type Judgements	$\Theta \vdash A$		
Term Judgements	$\vdash \langle \Gamma \rangle \; M \; \langle \Sigma \rangle$		

Type Formation Rules

$$\frac{\vdash \Theta}{\Theta \vdash \Theta_i} \quad \frac{\vdash \Theta}{\Theta \vdash I} \quad \frac{\vdash \Theta}{\Theta \vdash \mathbf{A}} \quad \frac{\Theta \vdash A \qquad \Theta \vdash B}{\Theta \vdash A \star B} \star \in \{+, \otimes\} \quad \frac{\Theta, X \vdash A}{\Theta \vdash \mu X.A},$$

where **A** ranges over a set of atomic types.

Syntax : discarding

$$\frac{}{\vdash \langle \Gamma, x : A \rangle\ \mathbf{discard}\ x\ \langle \Gamma \rangle}\ (\text{discard})$$

- Term formation rules are standard (omitted here).

Operational Semantics

- A *configuration* is a tuple (M, V), where:
  - M is a well-formed term $\Pi \vdash \langle \Gamma \rangle\ M\ \langle \Sigma \rangle$.
  - V is a *value assignment*, such that each input variable of M is assigned a value.
  - This data is subject to some well-formedness conditions (omitted).
- Program execution is modelled as a reduction relation on configurations $(M, V) \rightsquigarrow (M', V')$.
- This is pretty standard.

Slice categories for affine types

- Let $\mathbf{C}$ be a category and $I \in \mathrm{Ob}(\mathbf{C})$.
- Consider $\mathbf{C}/I$, the slice category of $\mathbf{C}$ with the fixed object I.
- Objects of $\mathbf{C}/I$ are pairs $(A, \diamond_A : A \to I)$.
- Morphisms $f : (A, \diamond_A) \to (B, \diamond_B)$ are the morphisms $f : A \to B$ of $\mathbf{C}$, s.t. $\diamond_B \circ f = \diamond_A$.
- Forgetful functor $U : \mathbf{C}/I \to \mathbf{C}$.
- **Proposition:** The functor $U : \mathbf{C}/I \to \mathbf{C}$ reflects small colimits.

Slice categories for affine types (monoidal structure)

Proposition

The category $\mathbf{C}/I$ *inherits a (symmetric) monoidal structure from* $(\mathbf{C}, \otimes, I)$.

$$(A, \diamond_A) \boxtimes (B, \diamond_B) \coloneqq (A \otimes B, \lambda_I \circ (\diamond_A \otimes \diamond_B))$$
$$f \boxtimes g \coloneqq f \otimes g$$

$U : \mathbf{C}/I \to \mathbf{C}$ *is a strict monoidal functor and:*

$$\otimes \circ (U \times U) = U \circ \boxtimes : \mathbf{C}/I \times \mathbf{C}/I \to \mathbf{C}.$$

Slice categories for affine types (coproducts)

Proposition

The category $\mathbf{C}/I$ inherits finite coproducts from $\mathbf{C}$.

$$(A, \diamond_A) \amalg (B, \diamond_B) := (A + B, [\diamond_A, \diamond_B])$$

Moreover:

$$+ \circ (U \times U) = U \circ \amalg : \mathbf{C}/I \times \mathbf{C}/I \to \mathbf{C}.$$

Parameterised Initial Algebras

- To interpret *mutual* type induction, we need *parameterised* initial algebras.
- Let $T : \mathbf{A} \times \mathbf{B} \to \mathbf{B}$ a functor. A *parameterised initial algebra* for T is a pair $(T^\dagger, \tau)$, such that:
  - $T^\dagger : \mathbf{A} \to \mathbf{B}$ is a functor;
  - $\tau : T \circ \langle \mathrm{Id}, T^\dagger \rangle \Rightarrow T^\dagger : \mathbf{A} \to \mathbf{B}$ is a natural transformation;
  - For every $A \in \mathrm{Ob}(\mathbf{A})$, the pair $(T^\dagger A, \tau_A)$ is an initial $T(A, -)$-algebra.
- When $\mathbf{A} = \mathbf{1}$ we recover the usual notion of initial algebra.
- **Proposition:** If $\mathbf{B}$ has an initial object and all ω-colimits, then any ω-cocontinuous functor $T : \mathbf{A} \times \mathbf{B} \to \mathbf{B}$ has a parameterised initial algebra $(T^\dagger, \tau)$. Moreover, $T^\dagger$ is also ω-cocontinuous.

Slice categories for affine types (initial algebras)

Proposition

$\mathbf{C}/I$ inherits an initial object and all ω-colimits from $\mathbf{C}$. Moreover, the forgetful functor $U : \mathbf{C}/I \to \mathbf{C}$ preserves and reflects them.

Theorem

Let $H : (\mathbf{C}/I)^n \to \mathbf{C}/I$ be a functor and $T : \mathbf{C}^n \to \mathbf{C}$ an ω-cocontinuous functor, such that the diagram:

$$\begin{tikzcd} (\mathbf{C}/I)^n \arrow[r, "U^{\times n}"] \arrow[d, "H"'] & \mathbf{C}^n \arrow[d, "T"] \\ \mathbf{C}/I \arrow[r, "U"'] & \mathbf{C} \end{tikzcd}$$

commutes. Then, H is also ω-cocontinuous.

Slice categories for affine types (initial algebras)

Theorem

Let H and T be ω-cocontinuous functors, such that (1) commutes.

$$\begin{tikzcd}
(\mathbf{C}/I)^{n+1} \arrow[r, "U^{\times(n+1)}"] \arrow[d, "H"'] \arrow[dr, phantom, "(1)"] & \mathbf{C}^{n+1} \arrow[d, "T"] \\
\mathbf{C}/I \arrow[r, "U"'] & \mathbf{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{C}/I^{n} \arrow[r, "U^{\times n}"] \arrow[d, "H^{\dagger}"'] \arrow[dr, phantom, "(2)"] & \mathbf{C}^{n} \arrow[d, "T^{\dagger}"] \\
\mathbf{C}/I \arrow[r, "U"'] & \mathbf{C}
\end{tikzcd}$$

Then (2) also commutes, where $(T^{\dagger}, \phi)$ and $(H^{\dagger}, \psi)$ are the respective parameterised initial algebras.

Slice categories for affine types (initial algebras)

Theorem

Let H and T be ω-cocontinuous functors, such that (1) commutes.

$$
\begin{tikzcd}
(\mathbf{C}/I)^{n+1} \arrow[r, "U^{\times(n+1)}"] \arrow[d, "H"'] \arrow[dr, phantom, "(1)"] & \mathbf{C}^{n+1} \arrow[d, "T"] \\
\mathbf{C}/I \arrow[r, "U"'] & \mathbf{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
(\mathbf{C}/I)^{n} \arrow[r, "U^{\times n}"] \arrow[d, bend right, "H \circ \langle Id, H^{\dagger} \rangle"'] \arrow[d, bend left, "H^{\dagger}"] \arrow[d, phantom, "\overset{\psi}{\Longrightarrow}"] \arrow[dr, phantom, "(2)"] & \mathbf{C}^{n} \arrow[d, bend right, "T \circ \langle Id, T^{\dagger} \rangle"'] \arrow[d, bend left, "T^{\dagger}"] \arrow[d, phantom, "\overset{\phi}{\Longrightarrow}"] \\
\mathbf{C}/I \arrow[r, "U"'] & \mathbf{C}
\end{tikzcd}
$$

Then the 2-categorical diagram (2) also commutes, where $(T^{\dagger}, \phi)$ and $(H^{\dagger}, \psi)$ are the respective parameterised initial algebras.

Slice categories for affine types (summary)

- $\mathbf{C}/I$ inherits a monoidal structure from $\mathbf{C}$.
- $\mathbf{C}/I$ inherits coproducts from $\mathbf{C}$.
- $\mathbf{C}/I$ inherits ω-colimits from $\mathbf{C}$.
- Parameterised initial algebras in $\mathbf{C}/I$ are constructed in the same way as in $\mathbf{C}$ for functors that may be lifted to $\mathbf{C}/I$.
- Therefore, $\mathbf{C}/I$ has sufficient structure to interpret inductive datatypes.

Categorical Model

A categorical model of our language (with recursion) is given by the following data:

1. A symmetric monoidal category $(\mathbf{C}, \otimes, I)$ with finite coproducts $(\mathbf{C}, +, \varnothing)$.
2. The tensor product $\otimes$ distributes over $+$.
3. For each atomic type $\mathbf{A} \in \mathcal{A}$, an object $\mathbf{A} \in \mathrm{Ob}(\mathbf{C})$ together with a discarding map $\diamond_{\mathbf{A}} : \mathbf{A} \to I$.
4. $\mathbf{C}$ has all ω-colimits and $\otimes$ is an ω-cocontinuous functor.

(5.) (The above data is $\mathbf{DCPO}_{\perp!}$-enriched.)

Proposition

The induced functors $\boxtimes : \mathbf{C}/I \times \mathbf{C}/I \to \mathbf{C}/I$ and $\amalg : \mathbf{C}/I \times \mathbf{C}/I \to \mathbf{C}/I$ are both ω-cocontinuous.

Interpretation of Types

$$
\begin{aligned}
\llbracket \Theta \vdash A \rrbracket &: \mathbf{C}^{|\Theta|} \to \mathbf{C} \\
\llbracket \Theta \vdash \Theta_i \rrbracket &= \Pi_i \\
\llbracket \Theta \vdash I \rrbracket &= K_I \\
\llbracket \Theta \vdash \mathbf{A} \rrbracket &= K_{\mathbf{A}} \\
\llbracket \Theta \vdash A + B \rrbracket &= + \circ \langle \llbracket \Theta \vdash A \rrbracket, \llbracket \Theta \vdash B \rrbracket \rangle \\
\llbracket \Theta \vdash A \otimes B \rrbracket &= \otimes \circ \langle \llbracket \Theta \vdash A \rrbracket, \llbracket \Theta \vdash B \rrbracket \rangle \\
\llbracket \Theta \vdash \mu X.A \rrbracket &= \llbracket \Theta, X \vdash A \rrbracket^{\dagger}
\end{aligned}
$$

$$
\begin{aligned}
\lBrack \Theta \vdash A \rBrack &: (\mathbf{C}/I)^{|\Theta|} \to \mathbf{C}/I \\
\lBrack \Theta \vdash \Theta_i \rBrack &= \Pi_i \\
\lBrack \Theta \vdash I \rBrack &= K_{(I, \mathrm{id}_I)} \\
\lBrack \Theta \vdash \mathbf{A} \rBrack &= K_{(\mathbf{A}, \diamond_{\mathbf{A}})} \\
\lBrack \Theta \vdash A + B \rBrack &= \amalg \circ \langle \lBrack \Theta \vdash A \rBrack, \lBrack \Theta \vdash B \rBrack \rangle \\
\lBrack \Theta \vdash A \otimes B \rBrack &= \boxtimes \circ \langle \lBrack \Theta \vdash A \rBrack, \lBrack \Theta \vdash B \rBrack \rangle \\
\lBrack \Theta \vdash \mu X.A \rBrack &= \lBrack \Theta, X \vdash A \rBrack^{\dagger}
\end{aligned}
$$

Proposition

$\llbracket \Theta \vdash A \rrbracket$ and $\lBrack \Theta \vdash A \rBrack$ are both (well-defined) ω-cocontinuous functors.

Relationship between type interpretations

Theorem

For any type $\Theta \vdash A$, the following diagram

$$\begin{tikzcd}
(\mathbf{C}/I)^{|\Theta|} \arrow[r, "U^{\times|\Theta|}"] \arrow[d, "(\!|\Theta \vdash A|\!)"'] & \mathbf{C}^{|\Theta|} \arrow[d, "[\![\Theta \vdash A]\!]"] \\
\mathbf{C}/I \arrow[r, "U"'] & \mathbf{C}
\end{tikzcd}$$

commutes. Therefore, for any closed type $\cdot \vdash A$, we have $[\![A]\!] = U(\!|A|\!)$.

Remark

This shows that $(\!|A|\!)$ gives us both $[\![A]\!]$ and a discarding map $\diamond_A : [\![A]\!] \to I$.

Folding/Unfolding of Types

- Easy to prove type substitution lemma:
  - $\llbracket \Theta \vdash A[B/X] \rrbracket = \llbracket \Theta, X \vdash A \rrbracket \circ \langle \mathsf{Id}, \llbracket \Theta \vdash B \rrbracket \rangle.$
  - $\lBrack \Theta \vdash A[B/X] \rBrack = \lBrack \Theta, X \vdash A \rBrack \circ \langle \mathsf{Id}, \lBrack \Theta \vdash B \rBrack \rangle.$
- Now, we can define folding/unfolding maps:
  - $\mathrm{fold}_{\mu X.A} : \llbracket A[\mu X.A/X] \rrbracket = \llbracket X \vdash A \rrbracket \llbracket \mu X.A \rrbracket \cong \llbracket \mu X.A \rrbracket : \mathrm{unfold}_{\mu X.A}$
  - $\mathfrak{fold}_{\mu X.A} : \lBrack A[\mu X.A/X] \rBrack = \lBrack X \vdash A \rBrack \lBrack \mu X.A \rBrack \cong \lBrack \mu X.A \rBrack : \mathfrak{unfold}_{\mu X.A}$

Theorem

Given a closed type $\cdot \vdash \mu X.A$, then the following diagram

$$\begin{tikzcd}
U\lBrack A[\mu X.A/X] \rBrack \arrow[r, "U\mathfrak{fold}"] \arrow[d, equal] & U\lBrack \mu X.A \rBrack \arrow[d, equal] \\
\llbracket A[\mu X.A/X] \rrbracket \arrow[r, "\mathrm{fold}"'] & \llbracket \mu X.A \rrbracket
\end{tikzcd}$$

commutes. Therefore, folding/unfolding is a discardable isomorphism.

Soundness and Adequacy

- Values, terms and configurations may be interpreted in the standard way.
  - Remark: we have to show values are discardable. Hard part is folding/unfolding.
- **Soundness:** If $\mathcal{C} \rightsquigarrow \mathcal{D}$, then $[\![\mathcal{C}]\!] = [\![\mathcal{D}]\!]$.
- **Adequacy:** In any adequate categorical model ($\mathrm{id}_I \neq \perp$), for any closed term M:

$$[\![M]\!] \neq \perp \text{ iff } M \Downarrow .$$

Thank you for your attention!