A Semantics for Static Type Inference in a Nondeterministic Language

Martín Abadi
Systems Research Center
Digital Equipment Corporation

November 6, 1992

Abstract

Plotkin used the models of reduction in order to obtain a semantic characterization of static type inference in the pure λ-calculus. Here we apply these models to the study of a nondeterministic language, obtaining results analogous to Plotkin's.

1 Introduction

The models of reduction are a generalization of the usual syntactic λ-models for the pure λ-calculus (see [Plo92] and the references therein). If a term M reduces to a term N then its interpretation in a model of reduction is "smaller than" or equal to the interpretation of N (and not necessarily equal as in λ-models). Plotkin obtained a series of soundness and completeness results for static type inference with respect to models of reduction. With type inference in mind, it seems natural that M and N be interpreted differently, since it may be possible to infer a type for N but not for M.

The study of nondeterministic languages gives rise to an alternative motivation for considering models of reduction. In nondeterministic languages, the reduction from M to N may involve a sequence of choices. Hence the two terms may not behave equivalently in all contexts, and then we must interpret them differently, with M being "less determined" than N. (For example, M may equal $N + N'$, where N' is a third term and $+$ represents nondeterministic choice.)

This note extends the models of reduction to the interpretation of a simple nondeterministic λ-calculus. The results obtained are soundness and completeness theorems for reduction and for static type inference. For concreteness, we focus on a calculus with the β rule and with rules for nondeterministic choice; but there should not be any difficulty in treating the η rule too.

The next section reviews the language studied, with evaluation and typing rules. Section 3 describes the models of reduction. The following two sections include the results. As much as possible, we assume familiarity with Plotkin's paper.

2 The Language

The language is the usual untyped λ-calculus extended with the choice operation $+$, so a term can be: a variable x, an abstraction $\lambda x.M$, an application MN, or a binary sum $M+N$. Sharma and de' Liguoro studied this language in their theses [Sha84, dL92], obtaining interesting results about its operational and denotational semantics.

The basic reduction relation $\twoheadrightarrow_{\beta+}$ is axiomatized by a set of rules that contains the expected rules from [Plo92], in particular the β rule:

$$(\lambda x.M)N \twoheadrightarrow_{\beta+} [N/x]M$$

The novelties are the rules for choice:

$$M+N \twoheadrightarrow_{\beta+} M \qquad M+N \twoheadrightarrow_{\beta+} N$$

$$\frac{M \twoheadrightarrow_{\beta+} M' \quad N \twoheadrightarrow_{\beta+} N'}{M+N \twoheadrightarrow_{\beta+} M'+N'}$$

Some of the propositions below concern an extended reduction relation, $\twoheadrightarrow_{\beta+c}$, obtained reproducing the rules of $\twoheadrightarrow_{\beta+}$ with the subscript $\beta{+}c$ instead of $\beta+$, and adding the rule:

$$M \twoheadrightarrow_{\beta+c} M+M$$

This relation turns out to have semantic properties somewhat simpler than those of $\twoheadrightarrow_{\beta+}$.

The type inference rules are Curry's:

$$\Gamma \vdash x : \alpha \qquad (\text{if } x : \alpha \text{ is in } \Gamma)$$

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}$$

$$\frac{\Gamma \vdash M : \alpha \to \beta \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash MN : \beta}$$

with one additional rule, introduced by Hennessy and Ashcroft in the study of a nondeterministic typed λ-calculus [HA80]:

$$\frac{\Gamma \vdash M : \alpha \qquad \Gamma \vdash N : \alpha}{\Gamma \vdash M + N : \alpha}$$

This rule says that if M and N both have the type α then so does their sum. With the Curry-Howard isomorphism in mind, we may read this rule: if one can prove the proposition α with either of M and N, then one can prove α by first choosing one of them and then using it.

A more sophisticated rule is possible, using union types:

$$\frac{\Gamma \vdash M : \alpha \qquad \Gamma \vdash N : \beta}{\Gamma \vdash M + N : \alpha \cup \beta}$$

For simplicity, we do not consider union types; they seem to lead to significant complications.

A similar rule appears in Boudol's work [Bou91]:

$$\frac{\Gamma \vdash M : \alpha \qquad \Gamma \vdash N : \beta}{\Gamma \vdash M || N : \alpha \wedge \beta}$$

Here α and β are logical assertions (rather than types), $\alpha \wedge \beta$ is their conjunction, and $||$ is a parallel-composition operator. As the rule suggests, parallel composition is rather different from the traditional nondeterministic composition.

3 Models of Reduction

A model of β-reduction is a triple:

$$\mathcal{P} = \langle P, \cdot, [\![\ ]\!] \rangle$$

that satisfies certain conditions; in particular P is a partial order.

We define the models of β+c-reduction by imposing the additional conditions that P be a lower semi-lattice and that:

$$[\![M + N]\!]_\rho = [\![M]\!]_\rho \wedge [\![N]\!]_\rho \tag{1}$$

where the $\wedge$ on the right is the meet operation on P. The models of β+c-reduction resemble de' Liguoro's syntactical models. There are two important differences between them. First, de' Liguoro's definition is based on that of the usual syntactic λ-models (rather than on that of the models of β-reduction). Second, it includes a semilinearity condition, that

$$(a+b)\cdot c = (a\cdot c)+(b\cdot c)$$

Sharma gave an analogous condition in his study of conversion. This condition does not hold in the models constructed in the completeness proofs of section 5.

The models of β+-reduction have an even looser definition than the models of β+c-reduction. They are quadruples:

$$\mathcal{P} = \langle P, \cdot, \wedge, [\![\ ]\!]\rangle$$

where P is just a partial order (not necessarily a semi-lattice), with a monotonic, binary operation $\wedge$, and with property (1) and the new property:

$$a\wedge b \leq a \qquad a\wedge b\leq b \tag{2}$$

for all $a, b \in P$. Obviously every model of β+c-reduction with its $\wedge$ operation is also a model of β+-reduction.

In models of reduction, types are interpreted as upper-closed subsets. Here, in addition, we require that they be closed under $\wedge$. For models of β+c-reduction, this requirement implies that types are filters (possibly empty). The conditions on $\rightarrow$ are unchanged.

4 Reduction

This section contains soundness and completeness results for the evaluation rules. As could be expected, the models of β+-reduction correspond to the relation $\twoheadrightarrow_{\beta+}$, and the models of β+c-reduction correspond to $\twoheadrightarrow_{\beta+c}$.

Proposition 1 (Soundness) *If $M \twoheadrightarrow_{\beta+} N$ then $[\![M]\!]_\rho \leq [\![N]\!]_\rho$ in all models of β+-reduction. If $M \twoheadrightarrow_{\beta+c} N$ then $[\![M]\!]_\rho \leq [\![N]\!]_\rho$ in all models of β+c-reduction.*

Proof The arguments proceed by induction on the proofs of the formal systems for reduction. All cases are evident, except that of the β rule, treated in the original paper, and that of the + rules. Condition (2) and the monotonicity of $\wedge$ are semantic paraphrases of the rules for + included in $\twoheadrightarrow_{\beta+}$; with condition (1), these properties suffice to guarantee the soundness of $\twoheadrightarrow_{\beta+}$ in models of $\beta+$-reduction. The rule $M \twoheadrightarrow_{\beta+c} M+M$ is validated by models of $\beta+c$-reduction because $[\![M+M]\!]_\rho = [\![M]\!]_\rho \wedge [\![M]\!]_\rho$ by condition (1) and $[\![M]\!]_\rho \wedge [\![M]\!]_\rho = [\![M]\!]_\rho$ whenever $\wedge$ is the meet operation in a semi-lattice. □

Proposition 2 (Completeness) *If $[\![M]\!]_\rho \leq [\![N]\!]_\rho$ in all models of $\beta+$-reduction then $M \twoheadrightarrow_{\beta+} N$. If $[\![M]\!]_\rho \leq [\![N]\!]_\rho$ in all models of $\beta+c$-reduction then $M \twoheadrightarrow_{\beta+c} N$.*

Proof The argument consists in constructing a particular model of reduction, a term model, such that if $[\![M]\!]_\rho \leq [\![N]\!]_\rho$ then it must be that M reduces to N. Recall that $[M]$ is the set of terms interreducible with M, here with $\twoheadrightarrow_{\beta+}$ or $\twoheadrightarrow_{\beta+c}$ as reduction relation; and $[M] \leq [N]$ if M reduces to N.

To extend Plotkin's argument to $\twoheadrightarrow_{\beta+}$, we define $[M] \wedge [N] = [M+N]$, trivially satisfying condition (1). It remains to check:

- $\wedge$ is well defined: If M and N are interreducible with M' and N', then $M+N$ is interreducible with $M'+N'$.
- $\wedge$ is monotonic: If $M \twoheadrightarrow_{\beta+} M'$ and $N \twoheadrightarrow_{\beta+} N'$ then $M+N \twoheadrightarrow_{\beta+} M'+N'$.
- $[M+N] \leq [M]$ and $[M+N] \leq [N]$ follow from $M+N \twoheadrightarrow_{\beta+} M$ and $M+N \twoheadrightarrow_{\beta+} N$.

To extend Plotkin's argument to $\twoheadrightarrow_{\beta+c}$, we have to check that $T = \{[M] \mid M \text{ a term}\}$ is a lower semi-lattice and that in fact $[M+N] = [M] \wedge [N]$ defines its meet operation. Any partial order with a monotone operation $\wedge$ such that $a \wedge b \leq a$, $a \wedge b \leq b$, and $c \leq c \wedge c$ is a lower semi-lattice, with meet $\wedge$. All the conditions but the last one, $c \leq c \wedge c$, are checked in the argument for $\twoheadrightarrow_{\beta+}$. The last condition is guaranteed by the special clause in the definition of $\twoheadrightarrow_{\beta+c}$, since $M \twoheadrightarrow_{\beta+c} M+M$ yields $[M] \leq [M] \wedge [M]$. □

5 Type Inference

Now we can give analogues of Plotkin's results for type inference. From the point of view of type inference, models of $\beta+$-reduction and models of

β+c-reduction turn out to be identical, so we present our soundness result in terms of the larger class and our completeness results in terms of the smaller one. We write $\models_{\beta+}$ and $\models_{\beta+c}$ for the corresponding semantic entailment relations.

Proposition 3 (Soundness) *If* $\Gamma \vdash M : \alpha$ *then* $\Gamma \models_{\beta+} M : \alpha$.

Proof As usual the argument goes by induction on the type-assignment proofs, with a new case for the added rule. This rule is justified using requirement (1) and the closure of types under $\wedge$: if $[\![M]\!]_\rho, [\![N]\!]_\rho \in \alpha$ then $[\![M + N]\!]_\rho = [\![M]\!]_\rho \wedge [\![N]\!]_\rho$ by requirement (1), and then $[\![M + N]\!]_\rho \in \alpha$ by the closure of types under $\wedge$. □

Proposition 4 (Completeness) *If* $\Gamma \models_{\beta+c} M : \alpha$ *then* $\Gamma \vdash M : \alpha$.

We give two proofs for this result. The first one requires a Subject Reduction result:

Proposition 5 (Subject Reduction) *If* $M \twoheadrightarrow_{\beta+c} N$ *and* $\Gamma \vdash M : \alpha$ *then* $\Gamma \vdash N : \alpha$.

Proof The argument is by induction on reduction derivations. The only novelties concern the rules for $+$. For the cases of $M + N \twoheadrightarrow_{\beta+c} M$ and $M + N \twoheadrightarrow_{\beta+c} N$, we assume that there is a proof of $\Gamma \vdash M + N : \alpha$; the last step in the proof must be the typing rule for $+$, which has as hypotheses $\Gamma \vdash M : \alpha$ and $\Gamma \vdash N : \alpha$, so these judgements must be provable too. The case of $M \twoheadrightarrow_{\beta+c} M + M$ is easy: if $\Gamma \vdash M : \alpha$ then $\Gamma \vdash M + M : \alpha$, by the typing rule for $+$. The remaining reduction rule for $+$ is handled like the usual rule for application. □

The first proof of Proposition 4 is based on the model of β+c-reduction defined in Proposition 2.

Proof 1 We construct a particular type interpretation for the model of β+c-reduction given in the proof of Proposition 2. The interpretation X_α of the type expression α is the set of classes $[M]$ such that $\mathcal{B} \vdash M : \alpha$, where $\mathcal{B}$ is a certain fixed extension of Γ, defined as in [Plo92]. (This is a good definition by Subject Reduction.)

We need to check that these interpretations of types are closed under $\wedge$: suppose that $[M] \in X_\alpha$ and $[N] \in X_\alpha$, then $\mathcal{B} \vdash M : \alpha$ and $\mathcal{B} \vdash N : \alpha$; the type inference rule for $+$ then yields $\mathcal{B} \vdash M{+}N : \alpha$, and hence $[M{+}N] \in X_\alpha$.

The result follows from $[M + N] = [M] \wedge [N]$, an equality established in Proposition 2. $\square$

The second proof relies on a type-expression-model construction.

Proof 2 In a type-expression model, the semantics of a term is defined as a set of type expressions. The sets of type expressions form a lattice, with $\cap$ for $\wedge$. Therefore $[\![M + N]\!]_\rho$ can be defined as $[\![M]\!]_\rho \cap [\![N]\!]_\rho$, and condition (1) is satisfied. (Intuitively, this reflects that the set of types for $M + N$ is the intersection of the sets of types for M and for N.)

The type expression α is interpreted as the set X_α of sets of types a such that $\alpha \in a$. The set of types is the set of all X_α. It is obvious then that if a and b are both elements of X_α then so is $a \cap b$, and hence types are closed under meets as required.

One of Plotkin's lemmas states that $[\![M]\!]_{\hat{\Gamma}} = \{\alpha \mid \Gamma \vdash M : \alpha\}$ for all M, with $\hat{\Gamma}(x) = \{\alpha \mid x : \alpha \in \Gamma\}$. Now we can show that this lemma can be extended with a trivial case for $+$. This case goes: $[\![M + N]\!]_{\hat{\Gamma}}$ equals $[\![M]\!]_{\hat{\Gamma}} \cap [\![N]\!]_{\hat{\Gamma}}$ by definition, $\{\phi \mid \Gamma \vdash M : \phi\} \cap \{\psi \mid \Gamma \vdash N : \psi\}$ by induction hypothesis, and $\{\alpha \mid \Gamma \vdash M + N : \alpha\}$ by the type rules.

The desired completeness result follows immediately. $\square$

The type-expression model constructed in the second proof is also a model of η, and so the corresponding type interpretation is both simple and an F-interpretation. The model also satisfies an additional law:

$$[\![\lambda x.(M + N)]\!]_\rho = [\![(\lambda x.M) + (\lambda x.N)]\!]_\rho \tag{3}$$

This condition makes some sense, as the terms $\lambda x.(M + N)$ and $(\lambda x.M) + (\lambda x.N)$ behave identically in every context and are equivalent for type inference too. Thus, we have soundness and completeness results for the models of reduction that satisfy (3).

Note that the analogous condition for $M(N + N')$ and $(MN) + (MN')$ is not sensible, since these two terms behave differently (at least for some calling mechanisms; see [Sha84]). And the condition for $(M + M')N$ and $(MN) + (M'N)$ would not work, since although these two terms behave identically the latter can be easier to type. For example, if the environment Γ contains $x : (s \to s) \to \alpha$ and $y : (t \to t) \to \alpha$ for s and t different type variables and α any type expression, then $\Gamma \vdash x(\lambda z.z) + y(\lambda z.z) : \alpha$ but $\Gamma \not\vdash (x + y)(\lambda z.z) : \alpha$. The type-expression model distinguishes $(x + y)(\lambda z.z)$ and $x(\lambda z.z) + y(\lambda z.z)$.

Acknowledgements

Gordon Plotkin provided much useful help. In particular he argued in favor of the use of models of $\beta +$-reduction and suggested looking at condition (3) and similar ones. Ugo de' Liguoro gave me guidance in understanding previous work on nondeterministic λ-calculi.

References

[Bou91] Gérard Boudol. Lambda-calculi for (strict) parallel functions. To appear in Information and Computation, 1991.

[dL92] Ugo de' Liguoro. *Nondeterministic untyped λ-calculus: A study about explicit non determinism in higher-order functional calculi.* PhD thesis, Università di Roma "La Sapienza", 1992.

[HA80] M.C.B. Hennessy and E.A. Ashcroft. A mathematical semantics for a nondeterministic typed λ-calculus. *Theoretical Computer Science*, 11:227–245, 1980.

[Plo92] Gordon Plotkin. A semantics for static type inference. *Information and Computation*, 1992. In this issue. A preliminary version appeared in *Theoretical Aspects of Computer Software*, Springer-Verlag LNCS 526.

[Sha84] Keshav Sharma. Syntactic aspects of the non-deterministic lambda calculus. Technical Report CS-84-127, Washington State University, September 1984. Thesis.

See also the references in [Plo92] and [dL92].