The Lambda-Calculus with Multiplicities

(abstract)

Gérard Boudol

INRIA-Sophia Antipolis

BP 93, 06902 SOPHIA-ANTIPOLIS CEDEX (FRANCE)

email: gbo@sophia.inria.fr

The λ-calculus with multiplicities is a refinement of the usual λ-calculus, inspired by the encoding of the lazy λ-calculus into the π-calculus given by Milner in [Milner 1992]. The basic observation is this: in a reduction step \((\lambda x. M) N \rightarrow M[N/x]\), the argument \(N\) is copied as many times as we need, that is, as much as there are free occurrences of \(x\) in \(M\). One could say that \(N\) is infinitely available. On the other hand, the π-calculus provides means, namely parallel composition and replication (or "bang"), for controlling the number of copies of an agent. One can show that this allows for distinguishing terms that are not distinguished in the λ-calculus.

In the refinement of the λ-calculus we propose, the argument of a function is a bag of resources, that is more precisely a multiset of terms. Then each term in the bag comes with an explicit, possibly infinite multiplicity, indicating how many copies of it are available. One recovers the usual λ-calculus when the bags consist of just one term with an infinite multiplicity. We shall write a bag as a parallel composition \(P = (M_1^{m_1} \mid \cdots \mid M_k^{m_k})\) of terms with multiplicities (where \(m_i\) is a non-negative integer, or \(\infty\)). The parallel composition is intended to be commutative and associative, with a neutral element \(1\), denoting the empty multiset. Then, besides the variables \(x, y, z\ldots\) and the abstractions \(\lambda x. M\), the syntax of our calculus includes applications of the form \((MP)\), where \(M\) is any term, and \(P\) a bag of terms. The management of the resources is done by means of explicit substitutions. That is, we use \(M[P/x]\) not just as a notation for the meta-operation of substitution, but as

This work has been partly supported by the BRA CONFÉR, and by the PRC-CNRS "Modèles Logiques de la Programmation".
an explicit syntactic construct, as in [Abadi et al. 1990], which binds the variable $x$ in $M$. Summarizing, the syntax of the $\lambda$-calculus with multiplicities is as follows:

$$M ::= x \mid \lambda x.M \mid (MP) \mid (M[P/x])$$

$$P ::= 1 \mid M \mid (P | P) \mid M^\infty$$

The terms with finite multiplicity may be defined by:

$$M^0 = 1$$

$$M^{m+1} = (M | M^m)$$

The usual $\lambda$-calculus is obtained by restricting the syntax of bags of resources to $P ::= M^\infty$.

The evaluation rules are given in the appendix. The evaluation process is basically the lazy $\beta$-reduction, but we also have to incorporate the substitution process as a part of the evaluation. To evaluate $M[P/x]$, one fetches a resource from the bag $P$, if any, when one actually needs it, that is when the variable $x$ is in the head position in $M$. Typically, we have:

$$xP_1 ... P_n \rightarrow (M | P)_x \rightarrow (MP_1 ... P_n)[P/x]$$

The operational semantics of our calculus is defined as the usual Morris' preorder, relying on the notion of value. A value is any functional closure, that is any term given by the grammar:

$$V ::= \lambda x.M \mid (V[P/x])$$

We write $M \downarrow$ whenever $M$ has a value, that is $\exists V. M \rightarrow^* V$. Then the operational preorder is given by:

$$M \subseteq N \iff \forall C. \ C[M] \downarrow \Rightarrow C[N] \downarrow$$

where $C$ denotes any context (a term with a hole) and $C[M]$ is the term obtained by placing $M$ in the hole of $C$.

It is worth emphasizing two points here. First, if the bag $P$ in $M[P/x]$ is empty, nothing can be fetched out of it. Then a term like $x[1/x]$ is deadlocked. It is a closed normal form w.r.t. the evaluation process, but not a value. Second, since parallel composition is commutative and associative, any resource from the bag can be selected in the fetch operation. Then we can define a non-deterministic choice $(M \oplus N)$ as follows, provided $x$ is not free in $M$ or $N$:

$$(M \oplus N) =_{def} x[(M | N)/x]$$

One can see that $(M \oplus N) \rightarrow M[N/x]$ and $(M \oplus N) \rightarrow N[M/x]$. Moreover, the two terms $M[N/x]$ and $N[M/x]$ may be regarded as identical to $M$ and $N$ respectively, up to some garbage collection. These two features of deadlock and non-determinism do not arise in the usual $\lambda$-calculus. The non-deterministic choice allows one to deal with parallel functions, as in [Boudol 1990] (see also [Boudol 1991]).