Interdefinability of Parallel Operations in PCF

Allen Stoughton
Computer Science and Artificial Intelligence
School of Cognitive and Computing Sciences
University of Sussex
Falmer, Brighton BN1 9QH, England

Abstract. It is shown that the “parallel or” and “parallel conditional” operations are interdefinable elements of the continuous function model of the programming language PCF.

In his seminal paper [3], which the reader is assumed to be familiar with, Plotkin showed that the continuous function model of the programming language PCF contains certain “parallel” elements that are not definable by terms. The most famous of these elements is the “parallel or” operation, \( \text{por}: o \rightarrow o \rightarrow o \), which is defined by

\[
\text{por} \perp \text{tt} = \text{tt}, \quad \text{por} \text{tt} \perp = \text{tt}, \quad \text{por} \text{ff} \perp \text{ff} = \text{ff}.
\]

But two “parallel conditionals” are also of interest, \( \text{pif}_0: o \rightarrow o \rightarrow o \rightarrow o \) (over the booleans) and \( \text{pif}_1: o \rightarrow i \rightarrow i \rightarrow i \) (over the natural numbers), which are defined by

\[
\text{pif}_0 x \perp x = x, \quad \text{pif}_0 \text{tt} x \perp x = x, \quad \text{pif}_0 \text{ff} \perp x = x,
\]

for \( \kappa = o, i \). Note that \( \text{por} \) and \( \text{pif}_0 \) are finite (isolated), whereas \( \text{pif}_1 \) is infinite.

Plotkin showed that if constants \( \text{Plf}_0 \) and \( \text{Plf}_1 \) denoting \( \text{pif}_0 \) and \( \text{pif}_1 \) are added to PCF, then all finite elements of the continuous function model are denotable. Actually, only \( \text{Plf}_0 \) is needed to define \( \text{POr} = \lambda xy. \text{Plf}_0 x \text{tt} y \). Furthermore, given \( \text{Plf}_0 \), we can define

\[
\text{Plf}_1 = \lambda xyz. \text{Eq} 1 (\text{Plf}_0 x (\perp, y 1 0) (\perp, z 1 0)),
\]

where \( \text{Eq}: t \rightarrow i \rightarrow o \) is the easily definable equality test over the natural numbers (strict in both arguments), and we have written 0 and 1 instead of Plotkin’s numerals \( k_0 \) and \( k_1 \).

Abramsky [1] and Curien [2] independently sharpened Plotkin’s definability theorem by showing that simply adding a constant denoting \( \text{por} \) to PCF is enough to make all finite elements denotable. In particular, their theorem shows that \( \text{pif}_0 \) can be defined from \( \text{por} \). But this leaves open the question of whether \( \text{pif}_1 \) can be defined from \( \text{por} \). The answer is “yes”, as the following proposition shows, with the consequence that an element of the continuous function model is definable in PCF plus a constant denoting \( \text{por} \) iff it is definable in PCF plus a constant denoting \( \text{pif}_1 \).

\[\text{Appears in Theoretical Computer Science, 79:357-368, 1991.}\]
Proposition. The operations \( \mathsf{por} \), \( \mathsf{pif}_o \) and \( \mathsf{pif}_i \) are interdefinable elements of the continuous function model of PCF.

Proof. We have already shown that \( \mathsf{por} \) is definable from \( \mathsf{pif}_i \), which in turn is definable from \( \mathsf{pif}_o \). It thus remains to show that \( \mathsf{pif}_o \) is definable from \( \mathsf{por} \). Suppose that \( \mathsf{POr} \) is a constant denoting \( \mathsf{por} \). Define

\[
\mathsf{PIf}_o = \mathsf{Y}_o \cdot F \cdot 0,
\]

where \( \sigma = \iota \rightarrow o \rightarrow \iota \rightarrow \iota \rightarrow \iota \) and \( F: \sigma \rightarrow \sigma \) is defined by

\[
F = \lambda f n x y z. \ 
\begin{array}{l}
(\mathsf{POr} \ (\mathsf{PAnd} \ (\mathsf{Eq} \ y n) \ (\mathsf{Eq} \ z n))) \\
\quad (\mathsf{PAnd} \ x \ (\mathsf{Eq} \ y n)) \\
\quad (\mathsf{PAnd} \ (\mathsf{Not} \ x) \ (\mathsf{Eq} \ z n))
\end{array}
\]

\[
(f (+1 n) x y z).
\]

Here, \( \mathsf{Not}: o \rightarrow o \) is \( \lambda x. \mathsf{O}_s x f f t \), we have extended \( \mathsf{POr} \) to three arguments in the obvious way, and \( \mathsf{PAnd}: o \rightarrow o \rightarrow o \) is the “parallel and” operation, dual to \( \mathsf{POr} \):

\[
\mathsf{PAnd} = \lambda x y. \mathsf{Not} \ (\mathsf{POr} \ (\mathsf{Not} \ x) \ (\mathsf{Not} \ y)).
\]

The reader will have no trouble verifying that \( \mathsf{PIf}_o \) does in fact denote \( \mathsf{pif}_o \). \( \Box \)

The proof of this proposition makes use of ideas from the proofs by Abramsky and Curien of their definability theorem. Combining our proposition with Plotkin’s definability theorem, we have an alternative proof of Abramsky and Curien’s theorem.

References

