PFPL Supplement: Comparing \texttt{fix} and \texttt{self}

Robert Harper

October 2018

General recursion, \texttt{fix} : \( \tau \) is \( e \), is only sensible in a by-name dynamics for \texttt{PCF}, because it steps to \([\texttt{fix} x : \tau \texttt{is} e/x] e\), which substitutes a non-value for the variable \( x \) in the expression \( e \). In a by-value dynamics general recursion is, for this reason, not sensible and must be replaced by type-specific forms of self-reference, such as the self-referential function form \texttt{fun}\{\( \tau_1; \tau_2 \)}\( (f.x.e) \), which is postulated to be a value of type \( \tau_1 \rightarrow \tau_2 \). Such a function is a value that is unrolled on application, substituting the recursive function value itself for the recursive variable, \( f \), in the body of the function.

In \texttt{FPC} an alternative account of self-reference is provided by the recursive type \( \tau \) \texttt{self}, the type of self-referential values \texttt{self} \( x \) is \( e \). Within \( e \) the self-reference, \( x \), must be unrolled, writing \texttt{unroll}(\( x \)), to unroll the recursion and access the underlying expression \( e \). More precisely,

\[
\texttt{unroll}(\texttt{self} \ x \ \texttt{is} \ e) \mapsto [\texttt{self} \ x \ \texttt{is} \ x/\texttt{e}] \ e,
\]

which makes sense in either a by-name or a by-value dynamics.

Using \texttt{self} types the recursive factorial function has type \( (\texttt{nat} \rightarrow \texttt{nat}) \) \texttt{self}, which reveals in its type that it is self-referential. To call such a function, either externally or internally within its definition, it is necessary to first \texttt{unroll} the self-reference and apply it to an argument, which may be either a value or a computation, depending on the dynamics. In any case the self-referential variable is only ever replaced by a value.

Curiously, in \texttt{FPC} it is possible to define \texttt{fix} from \texttt{self}, obtaining the expected dynamics, even by value! The “trick” is to anticipate the need to unroll any self-reference within \texttt{self} \( x \) is \( e \) by forming \( \hat{e} \) to be \([\texttt{unroll}(x)/x] e\), and then defining \texttt{fix} : \( \tau \) is \( e \) to be the expression \texttt{unroll}(\texttt{self} \( x \) is \( \hat{e} \)). Observe that

\[
\texttt{fix} x : \tau \texttt{is} e = \texttt{unroll}(\texttt{self} \ x \ \texttt{is} \ \hat{e}) \\
\mapsto [\texttt{self} \ x \ \texttt{is} \ \hat{e}/x] \ \hat{e} \\
= [\texttt{self} \ x \ \texttt{is} \ \hat{e}/x] [\texttt{unroll}(x)/x] e \\
= [\texttt{unroll}(\texttt{self} \ x \ \texttt{is} \ \hat{e})/x] e \\
= [\texttt{fix} x : \tau \texttt{is} e/x] e.
\]

The penultimate line summarizes the result of the iterated substitution; it does not arise in the dynamics itself as a substitution of a non-value for a variable, which would be disallowed in the by-value case.
References