Generic programming with XML

Barry Jay

The *pattern calculus* is a new foundation for computation in which computation is based on pattern-matching. While static patterns can be expressed in, say, λ-calculus, significant new expressive power is realised by allowing patterns to be first-class entities, that can be passed as parameters, evaluated and returned as results. In particular, it supports two new forms of polymorphism, namely *path polymorphism* and *pattern polymorphism*. Among other things, this new approach is able to represent generic queries for selecting and updating in an arbitrary data structure or database.

The pure or untyped pattern calculus has developed somewhat since its presentation at ESOP last year [JK06]. Its terms are now given by

$$t ::= x \mid \widehat{x} \mid t\ t \mid [\theta]\, t \to t$$

consisting of variables, constructors, applications and cases. The representation of a λ-abstraction $\lambda x.s$ is $[x]\,\widehat{x} \to s$. Earlier accounts used a free variable for the binder instead of a constructor but the new approach eliminates some pathological examples.

The approach to typing has also simplified since last year [Jay06] so that the types are now just those of System **F**

$$T ::= X \mid T \to T \mid \forall X.T$$

while the terms are given by those of System **F** plus constructors and extensions:

$$t ::= x^T \mid t\ t \mid \lambda x^T.t \mid t\ T \mid \Lambda X.t \mid \widehat{x}^T \mid [\theta]\, t \to t | t.$$

The evaluation rules are:

$$\begin{array}{rcll} (\lambda x.s)\ u & \longrightarrow & \{u/x\}s & (\beta 1) \\ (\Lambda X.s)\ U & \longrightarrow & \{U/X\}s & (\beta 2) \\ ([\theta]p \to s \mid r)\ u & \longrightarrow & \{u\,/[\theta]\,p\}\ s\ r\ u & \text{(match)}. \end{array}$$

In this talk, the basics of the pattern calculus will be introduced and used to program with XML paths. More precisely, there is an abstract data type of XML paths which can be passed as parameters to suitable generic queries.

References

[Jay06] B. Jay. Typing first-class patterns. *HOR 2006*.

[JK06] B. Jay and D. Kesner. Pure pattern calculus. *ESOP 2006*, p 100–114.