COBRA 2011
First Workshop on Compilers by Rewriting, Automated

`http://www.rdp2011.uns.ac.rs/workshops/cobra.html`

May 29, 2011

Program

10:00–10:30. Coffee Break & Welcome

10:30–11:30. James Cheney (U of Edinburgh)
Towards mechanized verification of XQuery rewrite rules

As with many relational database systems, XQuery is often implemented by translating to a simpler core language and then applying rewrite rules to try to find a low-cost plan to evaluate the query. However, rules that seem obviously valid in the relational case can easily become invalid (or only conditionally valid) in XQuery, due to subtleties such as node identity. These problems get worse when extensions such as updates (XQuery Update Facility) or higher-order functions (XQuery 1.1) are considered; neither XQuery nor XQuery Update have a formal semantics that fully specifies the behavior of node identity. I will argue that such a semantics (ideally, an executable specification) is needed and that this can provide a foundation for verifying whether (and when) rewriting rules are safely applicable.

11:30-12:00. Takahide Nogayama (IBM Tokyo Research Lab)
Eclipse Plugin for Authoring and Debugging Higher-order Rewriting (contributed talk)

We present an Eclipse "plugin" for authoring rules in the CRSX system, which includes simple facilitites for breaking at a given rule and stepping through single rewrites.

12:00-14:00. Lunch break

14:00-14:45. Barry Jay and Jose Vergara (U of Tech., Sydney)
Growing a Language by Pattern Matching (contributed talk)

Pattern calculus treats all computation as pattern matching, which is, in turn, central to the implementation of programming languages. Hence, its realisation in the general-purpose language bondi provides a natural host in which to develop domain-specific languages (DSLs). bondi is a strongly typed language that supports mixing of all the programming styles required for language implementation, including imperative programming (for handling input streams), functional programming (for transforming abstract syntax trees), generic queries (for performing traversals) and object-orientation (for growing DSLs).

The approach is illustrated by growing a small combinatory language in which each language feature, i.e. each production of each grammar, is isolated within a single object-oriented class that captures the rules for parsing, type inference, evaluation and printing. Further, growth is used to support lambda-abstraction and pattern matching, providing good evidence that this combinatory language could grow to support all of bondi.

14:45-15:30. Kristoffer Rose (IBM Watson Research Center)
Strategies Considered Harmful

We present how the notations that CRSX use for specifying executable formal specifications, and explain how explicit reduction strategies can be avoided in the formalism by careful use of dependencies and the occasional "forcing" of a particular subtree to evaluate to a data value. We argue that this allows a simpler analysis for how parallel execution can be achieved based only on dependencies even in the higher order rewriting case.

15:30-16:00. Coffee break

16:00-17:00. Panel: Should we have reusable rewriters and other formal "meta-tools" for compiler writers?

We hope that the assembled researchers will share experiences with and requirements for "the ideal rewriting-based compiler-compiler;" hopefully we can all leave with a sense of purpose for creating it!