Relit: Implementing Typed Literal Macros in Reason

CHARLES CHAMBERLAIN, University of Chicago
CYRUS OMAR, University of Chicago

...

1 INTRODUCTION

Motivation similar to that in the ICFP paper – but compare more explicitly to how people use ppx to rewrite string literals to support notation, camlp4, and Reason's built in HTMLish notation.

2 OVERVIEW BY EXAMPLE

Cite the ICFP paper, give a simple example, list the reasoning principes

3 IMPLEMENTATION

(1) Extending the Reason parser
(2) Using singleton signatures to encode definitions
(3) Using exceptions for application + typechecking inside a PPX (but really we only need to signature check but thats not possible). Is it bad?
(4) Using ocamldep to determine module dependencies
(5) Awkwardness with packaging the parser
(6) Speculation: issues with OCaml + Reason support together (use menhir's parameterization?)

4 DISCUSSION

Successes:

(1) ocamldep was a straightforward way to do context independence
(2) can do typechecking and compile-time code execution in a ppx

Challenges:

(1) packaging / loading parsers at compile-time is a huge hassle
(2) jbuilder treats ppx differently
(3) ordering of ppx matters
(4) ocamldep can't see into spliced terms – how to fix?
(5) we need to extend merlin and other tools still...
(6) issues with using OCaml + Reason together

briefly: relationship between this stuff and Scheme/Racket/Scala-style macros

REFERENCES

Authors' addresses: Charles Chamberlain, University of Chicago, cchamberlain@uchicago.edu; Cyrus Omar, University of Chicago, comar@cs.uchicago.edu.