Technische Universität Berlin
Compiler Construction and Programming Languages
Prof. Dr. Peter Pepper · Secr. TEL12-2 · www.uebb.tu-berlin.de

Compiler Construction 1
Pepper · Lorenzen

Winter term 2014/2015
Handout: Dec 4, 2014
Submission: Dec 17, 2014

Milestone 2

Context checker for µ-Opal

Assignment Context checker
Implement the context checker for µ-Opal. Your context checker must be able to detect more than one error in the input program and give descriptive error messages and precise source locations.
Submit the source tree of your µ-Opal compiler implementation including your implementation of the context checker.

File to modify: `ContextChecker.scala`

Conditions for context correctness
(Repeated from milestone 1a.)

- There exists exactly one definition for `MAIN`.
- The names of all defined functions and the primitive functions are disjoint.
- The names of the parameters of the left-hand side of a definition are disjoint.
- The type of right-hand side of a definition is the same as the declared result type of its left-hand side.
- All expressions are well-typed:
  - A number is well-typed and has type `nat`.
  - `true` and `false` are well-typed and have type `bool`.
  - A variable `id` is well-typed if `id` is a parameter in the current context. Its type is the declared type of `id`.
  - A function call `id`($expr_1$,…,$expr_n$) is well-typed if `id` is a defined or primitive function of n parameters of types $type_1, \ldots, type_n$ and $expr_i$ is of type $type_i$. The type of the function call is the return type of `id`.
  - A conditional `IF` $expr_1$ `THEN` $expr_2$ `ELSE` $expr_3$ `FI` is well-typed if $expr_1$ has type `bool` and $expr_2$ and $expr_3$ have the same type. The type of the conditional is the common type of $expr_2$ and $expr_3$.

    An assertion `IF` $expr_1$ `THEN` $expr_2$ `FI` is well-typed if $expr_1$ has type `bool`. The type of the assertion is the type of $expr_2$.