

Milestone 3

Interpreter for μ -Opal

Assignment Interpreter

Implement an evaluation function for μ -Opal programs that realizes the transformation from an expression to a value as described below.

Your interpreter must provide an implementation of the primitive functions. Take care that numerically problematic situations like overflow, underflow or division by zero are properly reported as runtime errors (not as a Scala exceptions).

Submit the source tree of your μ -Opal compiler implementation including your implementation of the interpreter.

File to modify: `Interpreter.scala`

Evaluation

(Repeated from milestone 1a.)

A μ -Opal program returns the value of the right-hand side of the definition `MAIN`.

The value of an expression is defined as follows:

- A number or boolean denotes its own value.
- Assume a function call `id(expr1, ..., exprn)` and let val_i be the values of the argument $expr_i$.
 - If `id` is a primitive function the value of the call is the result returned by the predefined implementation of the primitive function with val_1, \dots, val_n as input.
 - If `id` is a defined function with right-hand side $expr$ and parameters x_1, \dots, x_n the value of the function call is $[x_1 \mapsto val_1, \dots, x_n \mapsto val_n]expr$.
- A conditional `IF expr1 THEN expr2 ELSE expr3 FI` yields the value of $expr_2$ if $expr_1$ has the value `true`. Otherwise, it yields the value of $expr_3$.

If the `ELSE`-branch is missing and $expr_1$ has the value `false` this is an error.