Interpretation
and
Compilation

TEST 1

Luis Caires

Universidade Nova de Lisboa
14 Octover 2014

Test Statement

The goal of the exercise first phase is to extension an interpreter for a simple expression language with constructs for pairs.

The basic project files will be available in the CLIP when the test starts.

The language includes arithmetic operations, identifiers and declaration blocks, as defined by the following grammar:

```
exp ::= exp + exp |
        exp - exp |
        exp * exp |
        exp / exp |
        ( exp )   |
        num |
        id  |
        decl (id = exp)+ in exp end
```

Test Statement

The grammar is to be extended with

```
exp ::= ... |
        (exp , exp) |
        fst exp |
        snd exp
```

A pair `(e1,e2)` denotes a pair of (any) values, and the expressions `fst` e and `snd` e are intended to respectively evaluate to the first and second component of the pair denoted by e,

In the next page we illustrate the evaluation of some sample expressions.

Examples

```
(2+3,6);;      // input (black)
```
(5,6)   // evaluation result (red)
```
((2+1,3*2),2*2);;
```
((3,6),4)
```
decl x=1 y=2 in (x+x,y+x) end;;
```
(2,3)
```
decl x=(1,2) in (snd x,fst x) end;;
```
(2,1)
```
decl x=(1,2) y=(3,4) in (x, fst y + snd x) end;;
```
((1,2),6)
```
decl x=(1,(3,4)) in
   (fst x + snd snd x, fst x + fst snd x) end;;
```
(5,4)

Test Statement

What you are expected to do (extending the provided source files):

Extend the JavaCC parser. Extend the LL grammar to deal with the additional three constructs pairs, fst, and snd.

AST construction. Define the additional AST node classes.

Representation of pair values. Define a class to represent pair values with its required operations, extending the **IValue** interface.

Extend the interpreter. Define the **IValue eval(Environ e)** method in the new ASTNode classes.

Test your interpreter. It should give the right answers for the examples shown and any other.