Handout Phase 2

Interpretation and Compilation

Luis Caires

Abstract Syntax

EE -> EE **;** EE | EE **:=** EE

| **num** | **id** | **bool** | **let** (**id** = EE)+ **in** EE **end**

| **new** EE | **<!>** EE

| **if** EE **then** EE **else** EE **end**

| **while** EE **do** EE **end**

| EE **binop** EE

| **unop** EE

Concrete Syntax

```
EM -> E(<;>EM)*                          ASTSeq(E1,E2)
E -> EA(< == > EA)?                      ASTEq(EA,EA)
EA -> T(<+>EA)*                          ASTAdd(E1,E2)
T -> F ( (<*>T)*                         ASTMul(F,T)
       | (<(>AL<)>)*                     ASTApply(F,AL)
       | <:=> E)                         ASTAssign(F,E)
AL -> (EM(<,>EM)*)?
PL -> (id(<,>id)*)?
F -> num | id | bool | let (id = EM)+ in EM end
   | <(> EM <)>
   | new F | <!> F
   | if EM then EM else EM end               ASTIf(EM,EM,EM)
   | while EM do EM end                      ASTWhile(EM,EM)
```

Basic operations

Arithmetic operations (on integer values)

E+E, E-E, E*E, E/E, -E

Relational operations

E==E, E>E, E<E, E<=E, E>=E

Logical operations (on boolean values)

E && E, E || E, ~E

IValues (schematic)

```
interface IValue { /* represents values */
void show();
}

// IValue eval(Environment env) { ... }

//Value constructors
VInt(n)
VBool(t)
VCell(value)
```

IValues (schematic)

```
class VInt implements IValue {
int v;
VInt(int v0) { v = v0; }
int getval() { return v;}
}
```

IValues (schematic)

```
class VCell implements IValue {
IValue v;
VCell(IValue v0) { v = v0; }
IValue get() { return v;}
void set(IValue v0) { v = v0;}
}
```

Interpreter with Dynamic Type Checking (idea)

```
class ASTAdd implements ASTNode {

IValue eval(Environment env) {
v1 = left.eval(env);
if (v1 instanceof  VInt) {
  v2 = right.eval(env)
  if (v2 instanceof VInt) {
      return new Vint((VInt)v1).getval()+((VInt)v2).getval())
}
throw TypeError("illegal arguments to + operator");
}
```

Examples

```
(new 3) := 6;;

let a = new 5 in a := !a + 1; !a end;;

let x = new 10
      s = new 0  in
while !x>0 do
      s := !s + !x ; x := !x – 1
end; !s
end;;
```