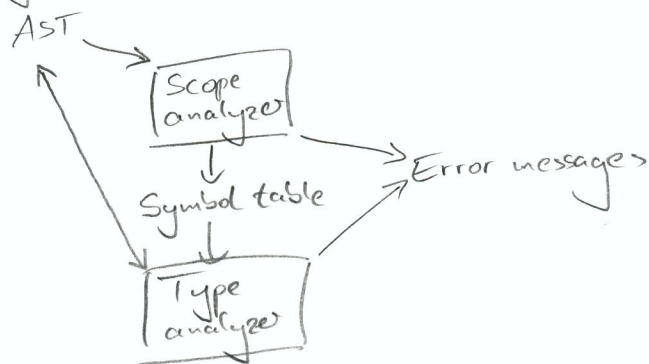


Wed 10/26/2011: Name and type analysis (p.1)

Lecture topics

- I Scopes and definitions
- II Scope analyzer
- III Types and their relations
- IV Type analyzer

Big picture



I Scopes and definitions

Definition	Example	Entity
FunDef	main = fun() → int { → 0; }	function main
VarDef	v = 1; # def v := v + 1; # assign	variable v
FieldLit	r = (f = 1, g = 2); print(r.f : string);	fields f, g
FieldType	r = (f = 1, g = 2) : (f : int, g : int);	fields f, g
FieldType	f = fun(p : int, q : int) → int { → p + q; }	variables (params p, q)
ForStmt	for i in [1, 2, 3] { print(i : string); }	variable (iterator i)

Lexical scoping example:

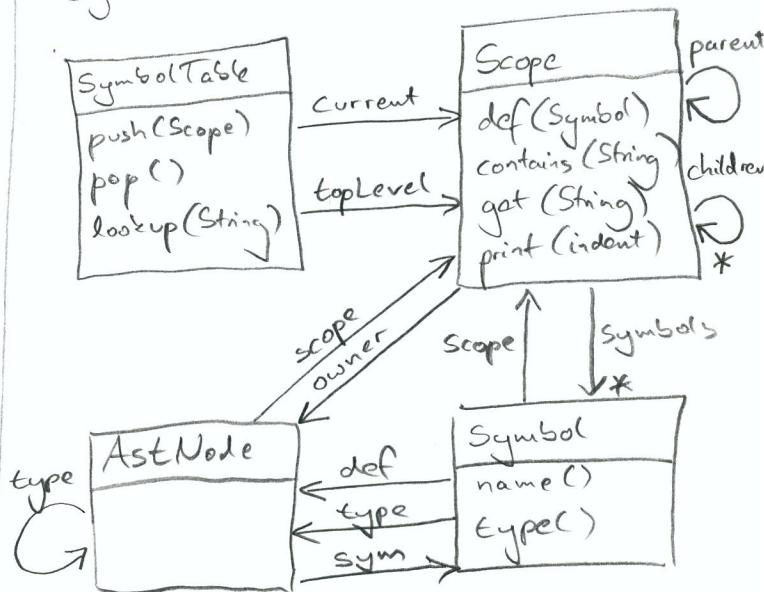
```

{ x = 1;
  { x = 2;
    x := x + 1;
  }
  print(x : string); # prints 1
}
    
```

Scope owner	Example
BlockStmt	{ x = "hi"; print(x); }
ForStmt	for j in ["a", "b"] { ... }
RecordLit	(f = 1, g = "two")
RecordType	(f : int, g : string)
FunDef	f = fun(p : int, q : int) { ... }
Program	f = fun ... main = fun ...

II Scope analyzers

Symbol table:



Translation scheme:

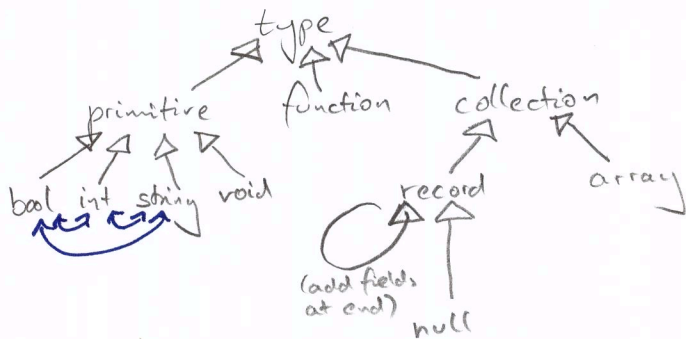
$B \rightarrow \{$ L $\}$ $L \rightarrow SL \epsilon$ $S \rightarrow B D U$ $D \rightarrow id = E;$ $U \rightarrow id;$	$B.scope = new Scope(B, current)$ $push(B.scope)$ $pop()$ $D.sym = new VarSym(D, current)$ $current.def(D.sym)$ (error if duplicate)
---	---

See also: Dragon book Figure 2.38 (page 30)

III Types and their relations

TACK: Subtype hierarchy

↑ subtype
↔ castable



Example:

```

a = (x=1, y="two");
b = (x=3);
b := a; # subtype
a := b : (x:int, y:string); # cast
print ("3");
print (3 : string); # cast
print ("" + 3); # coercion

```

IV Type analyzer

Translation scheme:

B → {	push (B.scope)
L	
}	pop ()
L → SLIE	
S → BIDIU	
D → id = E;	D.sym.type = E.type
U → id ;	U.sym = lookup(id) (error if not found or not var)

Example Java implementation:

see pr3.pdf, last page

Reminders

- pr 2 due Fr 10/28
- hw 6 due Fr 11/4