CSc 453

Compilers and Systems Software

14 : Intermediate Code II

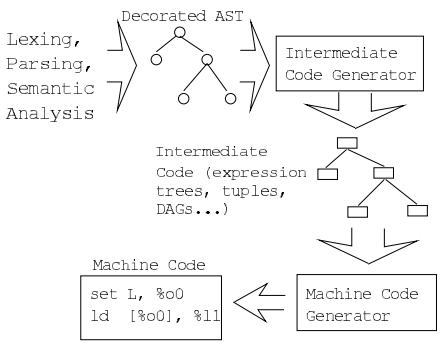
Department of Computer Science
University of Arizona

collberg@gmail.com

Copyright © 2009 Christian Collberg

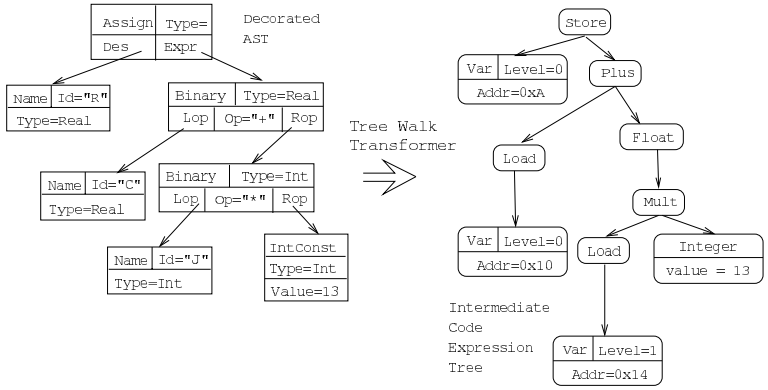
Intermediate Code Generation

Intermediate Code Generation



- After semantic analysis we traverse the AST and emit the correct intermediate code.

Generating Expression Trees



- An expression tree is generated from an AST. The `float` can easily be inserted since all types are known in the AST.

Generating Quadruples

- To generate quadruples from an AST we make an extra traversal of the tree after semantic analysis.
- Each AST node in an expression sub-tree is given an attribute ⇑Place:**SymbolT** which represents the name of the identifier or temporary in which the value of the subtree will be computed.

Generating Quadruples...

```
PROCEDURE Program (n:  Node);
   Decl(n.DeclSeq);
   Stat(n.StatSeq);
END;

PROCEDURE Decl (n:  Node);
   IF n.Kind = ProcDecl THEN
      Decl(n.Locals);
      Decl(n.Next);
      Stat(n.StatSeq);
   ENDIF
END;
```

Generating Quadruples...

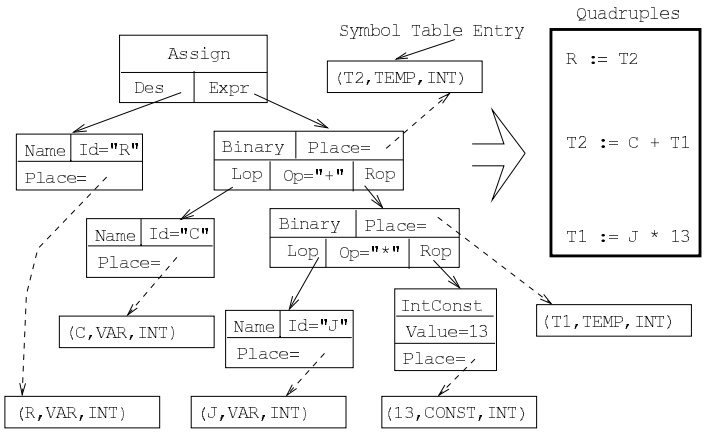
- NewTemp generates a new temporary var.

```
PROCEDURE Stat (n:  Node);
  IF n.Kind = Assign THEN
    Expr(n.Des);
    Expr(n.Expr);
    Emit(n.Des.Place ':=' n.Expr.Place);
    Stat(n.Next);
  ENDIF
END;
```

Generating Quadruples...

```
PROCEDURE Expr (n:  Node);
   IF n.Kind = Add THEN
      Expr(n.LOP);
      Expr(n.ROP);
      n.Place := NewTemp();
      Emit(n.Place ':=' n.LOP.Place '+' n.ROP.Place);
   ELSIF n.Kind = VarRef THEN
      n.Place := n.Symbol;
   ENDIF
END;
```

Generating Quadruples...



Attribute Grammar Notation

Attribute Grammar

- The tree-walker is better described using an attribute grammar notation.

Assign ::= Des:Expr Expr:Expr
{ Emit(n.Des.Place ':=' n.Expr.Place); }

Add ::= LOP:Expr ROP:Expr
{ n.Place := NewTemp();
Emit(n.Place ':=' n.LOP.Place '+' n.ROP.Place);
}

Name ::= Ident
{ n.Place := n.Symbol; }

Summary

Readings and References

- Read Louden:
  Generating Intermediate Code 407–410