CS480

Compilers

Chapter 1, 2 (and section 7.6)

Pages 1-51, 60 - 62, 429 - 440

February 9, 2009
Compilers

• Definition (again)

• Difference between compiler and
  – Eclipse
  – MS Visual Studio
  – Parts of each?

• Other source code analysis tools?
Two Jobs

• Analysis

• Synthesis/Generation
Compiler, in detail

- Lexical Analysis
- Syntactic Analysis
  - AST
- Semantic Analysis

taxes = salary * 0.30 + 500;
Compiler, in detail

• Intermediate code generator

  \[
  \text{taxes} = \text{salary} \times 0.30 + 500;
  \]

• Code generation
  – Optimization

• What's the difference between assembly lang and machine lang?
Example?

What does the compiler need to do here?

```c
#include <stdio.h>

#if SYS == RH5.1
#define LIB "rh51.h"
#elif SYS == RH6.0
#define LIB "rh60.h"
#else
#define LIB "rh62.h"
#endif

#include LIB

#define MAX(x,y) (((x) > (y) ? (x) : (y))

main ()
{
    int i = 3, j = 7;
    printf ("%d", MAX(i,j));
}
```
Compiler Terms

• Front End/Back End

• Passes
Linker/Loader

• What's a link error?
Ch 2: One-Pass Compiler

• Convert infix to postfix (RPN) expressions
  – why is RPN useful?

• Infix: $5 + 2 / 4 + 1$

• Postfix:
Language

• Context Free Grammar
  – components?

• Can you give a grammar for postfix notation?
  – Assume single digit ints

• Parse Tree?
  – Components?
  – Parse tree for 1 2 + 2 -
Grammars

\[
\begin{align*}
\text{bin} & \rightarrow \text{bin} + \text{bin} \\
\text{bin} & \rightarrow \text{bin} - \text{bin} \\
\text{bin} & \rightarrow 0 \mid 1
\end{align*}
\]

- Ambiguous?

- Associativity?
  - Is + left or right associative?
Grammars

\[
\begin{align*}
\text{expr} & \rightarrow \text{expr} + \text{term} \mid \text{expr} - \text{term} \mid \text{term} \\
\text{term} & \rightarrow \text{term} \ast \text{factor} \mid \text{term} / \text{factor} \mid \text{factor} \\
\text{factor} & \rightarrow \text{digit} \mid ( \ \text{expr} \ )
\end{align*}
\]

- Ambiguous?
- Left or Right associative
  \[
  + \quad - \quad \ast \quad / 
  \]
- \textbf{9 + 2 * 5}
Syntax Directed Translation

• Grammar for infix
• Grammar of postfix
• ???
• Translation
Annotated Parse Trees

• Semantic rules
• Synthesized Attributes
• Traversal
• Emit Translation
Parsing!

- Top Down

- Bottom Up
Symbol Tables (p 60-62, 429-440)

• Lexeme

• Runtime location
  – Where could a variable be stored at runtime?

• Type

• Level/Scope

• Array dimension/Number of parameters
Symbol Table

• Cross Reference Information

• Data structures
  – Unordered linear list
  – Ordered linear list
  – Binary search tree
  – AVL tree
  – advantages/disadvantages
  – Insert/Delete/Search O(??)