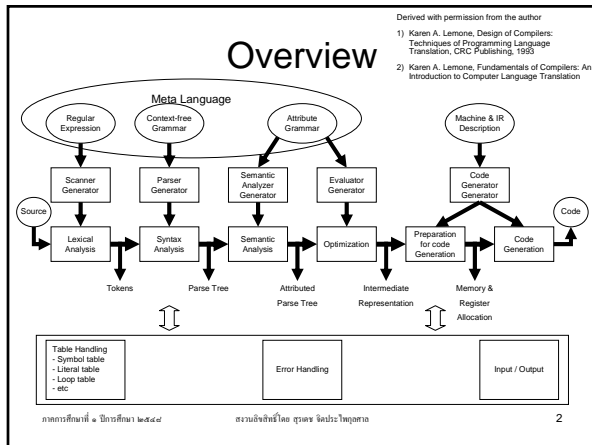


Compiler Construction

Suradet Jitprapaikularn




Lexical Analysis


Scanning or Tokenizing

- Break sequence of characters into a series of tokens

Lexical Analysis



```
x1 := a + b*4;
y2 := c/2 + b*4;
```



```
78 31 20 3A 3D 20 61 20
2B 20 62 2A 34 3B 0D 0A
79 31 20 3A 3D 20 63 2F
32 20 2B 20 62 2A 34 3B
0D 0A
```

ภาคการศึกษาที่ ๑ ปีการศึกษา ๒๕๖๔ ระบบอัตโนมัติของ ฮาร์ดแวร์ อิเล็กทรอนิกส์ 4

Lexical Analysis

```
78 31 20 3A 3D 20 61 20
2B 20 62 2A 34 3B 0D 0A
79 31 20 3A 3D 20 63 2F
32 20 2B 20 62 2A 34 3B
0D 0A
```

**Lexical
Analysis**

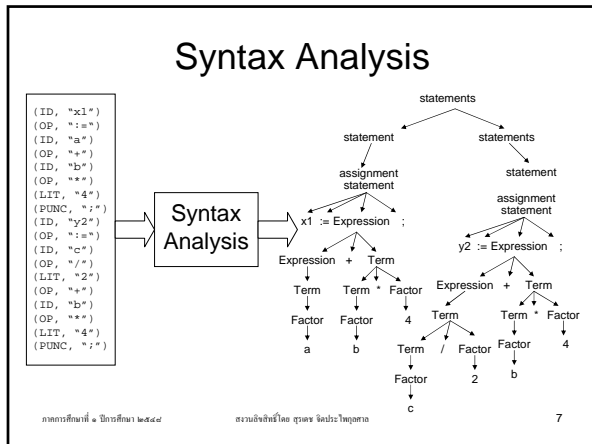
```
(ID, "x1")
(OP, ":=")
(ID, "a")
(OP, "+")
(ID, "b")
(OP, "**")
(LIT, "4")
(PUNC, ";")
(ID, "y2")
(OP, ":=")
(ID, "c")
(OP, "/")
(LIT, "2")
(OP, "+")
(ID, "b")
(OP, "**")
(LIT, "4")
(PUNC, ";")
```

ภาคการศึกษาที่ ๑ ปีการศึกษา ๒๕๖๔ ระบบอัตโนมัติของ ฮาร์ดแวร์ อิเล็กทรอนิกส์ 5

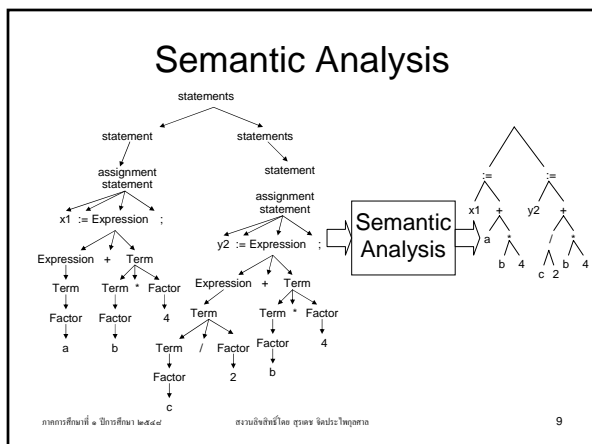
Syntax Analysis

- Determine the structure of the program and of the individual statement
- Determine if and how tokens can be generated by the grammar

ภาคการศึกษาที่ ๑ ปีการศึกษา ๒๕๖๔ ระบบอัตโนมัติของ ฮาร์ดแวร์ อิเล็กทรอนิกส์ 6



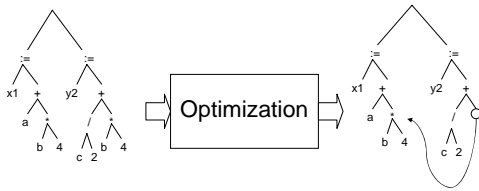
- ## Semantic Analysis
- Determine the meaning of the program
 - Static checking or type checking
 - Generation of Intermediate Representation (IR) or Intermediate Language (IL) or Intermediate Code (IC)
- ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี



Optimization

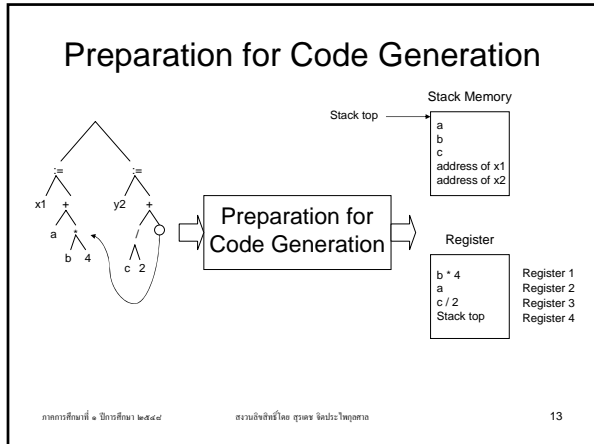
- Changes the IR so that the changed code will execute faster or take up less space or both
 - Local optimization
 - Loop optimization
 - Global optimization
 - Peep-hole optimization

Optimization



Preparation for Code Generation

- Assign and maintain space in memory for holding the value of variables and expressions
 - Memory allocation
 - Register allocation

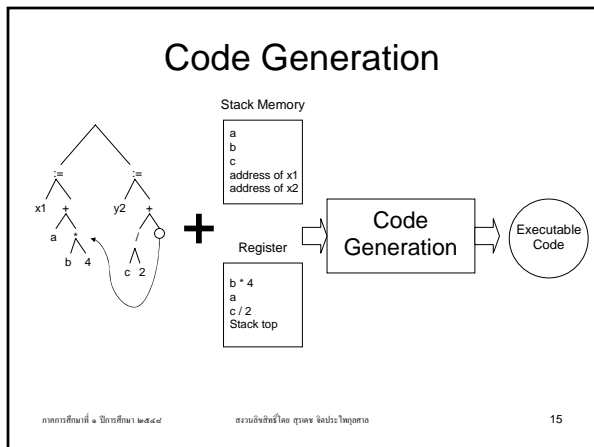


Code Generation

- Translates the IR to target language

ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

14



Supporting tools

- Table Handling
- Error Handling
- I/O Handling
- Loader
- Linker

Writing a Translator

Several ways to write a translator

- Machine code implementation
- High-level implementation
- Bootstrapping
- Cross-translator

Type of translator

- Assembler
- Compiler
- Interpreter
- Interpretive compiler
- Emulator
- Pre-processor
- Disassembler
- Decompiler
