

Language translation

Language translation

- ▶ How to translate a C program to machine code?
 - ▶ How to know '123' is a variable or a number?
 - ▶ How to know `/* It's a comment */` is a comment?
 - ▶ How to know the parenthesis `'5*(((1+2)*3)+4)*6'` is balanced?
 - ▶ How to know the execution order of `'if(a==++i == 3)'`?

▶ Two types of translations

Interpreted code

```
Load r1, 0x00004
Load r2, 0x00008
Addi r1, r2, r3
Store r3, 0x00000
```

```
Load r1, 0x00000
Load r2, 0x00008
Addi r1, r2, r3
Store r3, 0x0000D
```

Source code

```
x = y + z;
w = x + z;
```

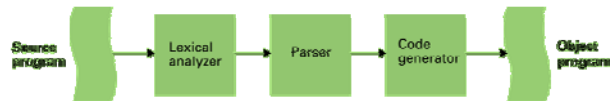
Compiled code

```
Load r1, 0x00004
Load r2, 0x00008
Addi r1, r2, r3
Addi r2, r3, r4
Store r4, 0x00000
```

Memory I/O (Load, Store) is much slower than computation.

Language implementations

- ▶ Interpreter: **interprets** and **executes** a program statement by statement
 - ▶ Perl, Matlab, JavaScript, BASIC, HTML ...
- ▶ Compiler: **translates** high level program primitives into machine codes.
 - ▶ C, C++, Fortran, Verilog ...
- ▶ The translation process



Lexical analyzer

- ▶ Breakdown a program into a list of tokens

`a = b + 32;`

Token	Type
a	Variable
=	Assignment operator
b	Variable
+	Addition operator
32	Integer
;	End of statement

- ▶ Ex: definition of variables

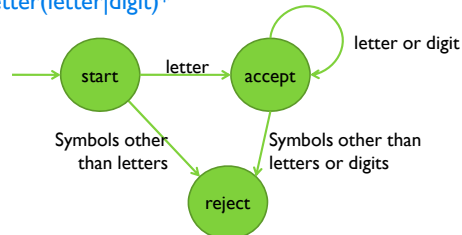
`letter(letter|digit)*`



- ▶ `a1234g5678t0000` is a variable
- ▶ `0abcdefg3` is not a variable

Regular expression and finite state machine (FSM)

- ▶ `letter(letter|digit)*` is a regular expression (正規表示法)
- ▶ Finite state machine (FSM) is a “machine” to recognize a regular expression
 - ▶ Ex: FSM for `letter(letter|digit)*`



- ▶ Try 'a123', 'a.123', '0a123'



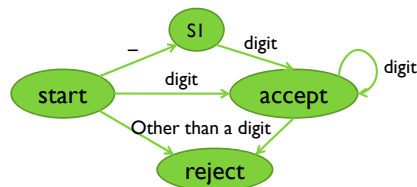
Some notations in a regular expression

Char	Behavior	Example
*	Matches the preceding character or expression zero or more times.	zo* matches "z" and "zoo".
+	Matches the preceding character or expression one or more times.	zo+ matches "zo" and "zoo", but not "z".
?	Matches the preceding character or expression zero or one time.	zo? matches "z" and "zo", but not "zoo".
()	Marks the start and end of an expression.	(0 1)? matches "0", "1", or ""
	Indicates a choice between two or more items.	z food matches "z" or "food". (z f)ood matches "zood" or "food".

- ▶ From <http://msdn.microsoft.com/en-us/library/ae5bf541.aspx>

Example: integer

- ▶ Description:
 - ▶ It may start with a negative sign: -
 - ▶ It has at least one digit: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- ▶ Regular expression for a floating point number
 - ▶ `-(0|1|2|3|4|5|6|7|8|9)+`
- ▶ The corresponding finite state machine



Parser

- ▶ Group tokens into meaningful structures
 - ▶ “meaningful” is defined by the **grammatical rules** of the programming language.
 - ▶ It can be hard even for human
 - ▶ *The man the horse that won the race threw was not hurt.*
- ▶ Three representations
 - ▶ **Grammar**: the rules to define the syntax
 - ▶ **Syntax diagram**: flow chart for a grammar
 - ▶ **Parse tree** (output of parser): the hierarchical structure of tokens



