Slides for

Compiler Construction Using Java, JavaCC, and Yacc

Chapter 1

Strings, Languages, and Compilers

Why study compilers?

- Compiler techniques have broad applicability.
- To program most effectively, you need to understand the compiling process.
- Language and language translation are at the very heart of computing.
- Not easy to learn compiler construction techniques "on the job."

Basic language concepts

A ***compiler*** is a translator. It translates the source program to the target program.

The ***source program*** typically is a high-level language.

The ***target program*** typically is machine language or assembly language.

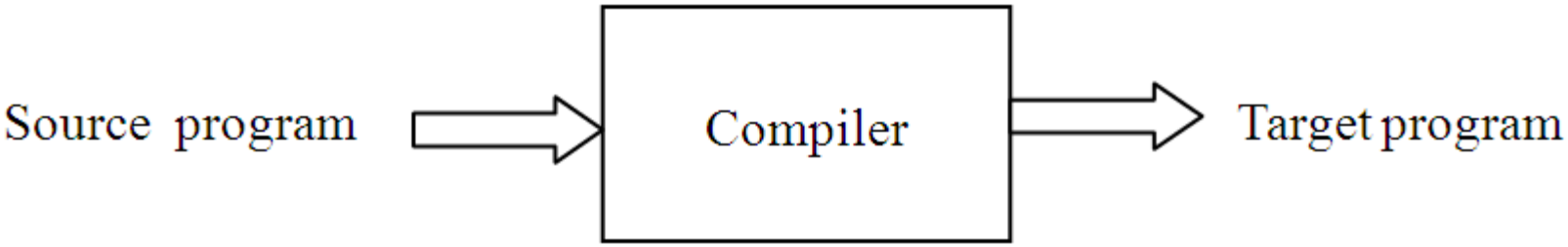
Basic compiler concepts

An ***alphabet*** is a finite set of symbols.

A ***string*** over an alphabet is a finite sequence of characters selected from that alphabet.

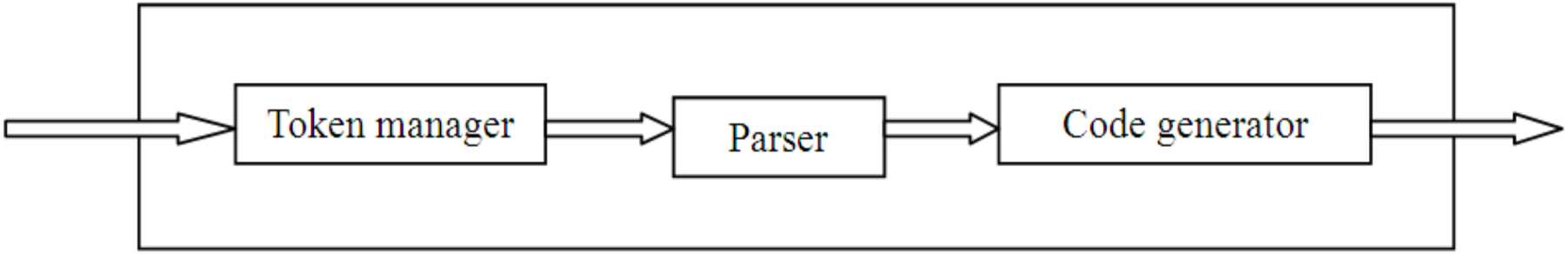
The ***length*** of a string is the number of characters it contains.

A ***language*** is a set of strings.


Source program
Compiler
Target program

Parts of a compiler

- Token manager (aka scanner, lexical analyzer)
- Parser
- Code generator


Token manager
Parser
Code generator

Null string

The string with zero characters

Represented with the Greek letter λ

As zero is useful in arithmetic, so is the null string in language theory.

Concatentation

Juxtapose two strings to get a new string.

bc concatentated with cc yields bccc.

Exponent notation

$b^5 = bbbbb$

$\{b^i : i \leq 3\} = \{b^0, b^1, b^2, b^3\} = \{\lambda, b, bb, bbb\}$

Star operator

Creates a set consisting of zero or more occurences of the starred item.

Resulting set always contains the null string. b* = {λ, b, bb, bbb, ...}

Can be applied to sets

{b,c} * = all strings over over alphabet {b, c}

Concatentating sets of strings

{b, cc} {bbb, c} = {bbbb, bc, ccbbb, ccc}

Plus operator

Creates a set contains one or mor occurrences of the plussed item.

b+ = {b, bb, bbb, ...}

Can be applied to sets.

{b, c} + = all strings over {b, c} of nonzero length

Question mark operator

Specifies optional item.

Creates set contain the null string and one occurrence of the marked item.

b? = {λ, b}

bc?b = {bb, bcb}

Operator precedence

Set complementation

Star, plus, question mark

Concatentation

Set intersection

set Union

Regular expressions

Expressions that use concatentation, star, set union, and parentheses.

Will use | to denote set union.

Each regular expression represents a language.

Examples of regular expressions:

$\varnothing$, λ, b, c, b*, bc, b|c, b*c*|(cc)*

Limitations of regular expressions

Cannot define *PAIRED* = $\{b^i c^i : i \geq 0\}$

b*c* does not equal *PAIRED*.

Structures like *PAIRED* appear in programming languages:

(x * (y + z))

{ { ...}}

begin begin ... end end