Lecture 1: Strings and Languages

Dr Kieran T. Herley

Department of Computer Science
University College Cork

2018/19

Summary

Formal languages and their role in the compilation process. Languages and strings. Set operations and use of same for pattern specification.

Formal Languages

- Compilers rely on *formal specifications* for various aspects source language syntax:
  - *lexical structure* –format of identifiers, numbers etc.
  - *syntactic structure* –format of if statements, while-loops etc.
-
  - A *formal language* is a set of strings that conform to some pattern/structure
  - Typically described in some mathematical notation (regular expressions, context-free grammars)
- Notation exploited within compilers to facilitate construction of elements of the compiler (e.g. grammar for parsing).

Symbols, Alphabets and Strings

Definition

An alphabet is a finite set of *symbols*.

Example

- $\Sigma_1 = \{0, 1\}$
- $\Sigma_2 = \{a, b, c, \cdots, z\}$

Symbols, Alphabets and Strings cont'd

Definition

A string over alphabet Σ is a finite sequence of zero or more symbols drawn from Σ.

Example

Some strings over Σ_1

- 0
- 1
- 10010
- ϵ

Last one denotes the empty string (zero symbols). Note ϵ is a *metasymbol* not a member of Σ_1.

More examples

Alphabet $\Sigma_1 = \{0, 1\}$

What about the following (wrt Σ_1)?

- 012

More examples

Alphabet $\Sigma_1 = \{0, 1\}$

What about the following (wrt Σ_1)?

- 012
- x10101

More examples

Alphabet $\Sigma_1 = \{0, 1\}$

What about the following (wrt Σ_1)?

- 012
- x10101
- "10101"

More examples

Alphabet $\Sigma_1 = \{0, 1\}$

What about the following (wrt Σ_1)?

- 012
- x10101
- "10101"
- 10001111.11101111.11010011.11100101

More examples

Alphabet $\Sigma_1 = \{0, 1\}$

What about the following (wrt Σ_1)?

- 012
- x10101
- "10101"
- 10001111.11101111.11010011.11100101
- 10001111 11101111 11010011 11100101

More Examples cont'd

Alphabet $\Sigma_2 = \{a, b, c, \cdots, z\}$

- Strings over Σ_2(?)
  - ϵ
  - `a`
  - `xyz`
  - `fiddlesticks`
  - `Tuesday`

More Examples cont'd

Alphabet $\Sigma_2 = \{a, b, c, \cdots, z\}$

- Strings over Σ_2(?)
  - ϵ
  - `a`
  - `xyz`
  - `fiddlesticks`
  - `Tuesday` No!
  - `The rain in Spain falls mainly in the plain.`

More Examples cont'd

Alphabet $\Sigma_2 = \{a, b, c, \cdots, z\}$

- Strings over Σ_2(?)
  - ϵ
  - `a`
  - `xyz`
  - `fiddlesticks`
  - `Tuesday` No!
  - `The rain in Spain falls mainly in the plain.` No!

String Length and Equality

Definition

The length of a string (denoted $|\alpha|$ for string α) is the number of symbols it contains.

1

Example

- $|\epsilon| = 0$
- $|xyz| = 3$

[1] For now, I'll use the Roman alphabet to denote individual symbols and the Greek to denote strings over that alphabet.

String Length and Equality cont'd

Definition

Two strings are equal if they contain exactly the same sequence of symbols.

Example

$$\mathsf{xyz} = \mathsf{xyz}, \quad \mathsf{xyz} \neq \mathsf{yxz}$$

String Concatenation

Definition

The concatenation of two strings α and β (denoted $\alpha \cdot \beta$) consists of the symbols of α followed by those of β.

[2]

Example

If $\alpha = abcd, \beta = efg$, then

$$\overbrace{abcd}^{\alpha} \cdot \overbrace{efg}^{\beta} = \overbrace{abcdefg}^{\alpha \cdot \beta}$$

[2] $\cdot$ is another metasymbol.

String Concatenation

Definition

The concatenation of two strings α and β (denoted $\alpha \cdot \beta$) consists of the symbols of α followed by those of β.

2

Example

If $\alpha = abcd, \beta = efg$, then

$$\overbrace{abcd}^{\alpha} \cdot \overbrace{efg}^{\beta} = \overbrace{abcdefg}^{\alpha \cdot \beta}$$

Example

$$\epsilon \cdot xyz = xyz = xyz \cdot \epsilon$$

[2] $\cdot$ is another metasymbol.

String Concatenation

Definition

The concatenation of two strings α and β (denoted $\alpha \cdot \beta$) consists of the symbols of α followed by those of β.

2

Example

If $\alpha = abcd, \beta = efg$, then

$$\overbrace{abcd}^{\alpha} \cdot \overbrace{efg}^{\beta} = \overbrace{abcdefg}^{\alpha \cdot \beta}$$

Example

$$\epsilon \cdot xyz = xyz = xyz \cdot \epsilon$$

[2] $\cdot$ is another metasymbol.

Notes on Concatenation

- The $\cdot$ operator often omitted: $abcd \cdot efg$ may be simplified to $abcdefg$.
- Concatenation is not *commutative*:

$$x \cdot y \neq y \cdot x.$$

- Concatenation is *associative*:

$$x \cdot (y \cdot z) = (x \cdot y) \cdot z.$$

Some Useful Notation

- If x is a symbol, then x^n denotes n repetitions of symbol x concatenated together:

$$x^n = \underbrace{x \cdot x \cdot x \cdot \cdots \cdot x}_{n \text{times}}$$

-

$$\begin{array}{rcl} x^0 & = & \epsilon \\ x^1 & = & x \\ x^2 & = & x \cdot x = xx \\ x^3 & = & x \cdot x \cdot x = xxx \end{array}$$

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_1 = \{0, 1\}$

- $\emptyset$ – the empty language
- $\{\epsilon, 0, 1\}$
- $\{0, 1, 00, 10, 11, 100, 101\}$
- binary representations of prime numbers

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_1 = \{0, 1\}$

- $\emptyset$ – the empty language
- $\{\epsilon, 0, 1\}$
- $\{0, 1, 00, 10, 11, 100, 101\}$
- binary representations of prime numbers

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_2 = \{a, b, c, \cdots, z\}$

- $\{a, aa, aaa, aaaa\}$
- $\{\text{one}, \text{two}, \text{three}\}$
- palindromes

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_2 = \{a, b, c, \cdots, z\}$

- $\{a, aa, aaa, aaaa\}$
- $\{\text{one}, \text{two}, \text{three}\}$
- palindromes
- legal Java reserved words?

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_2 = \{a, b, c, \cdots, z\}$

- $\{a, aa, aaa, aaaa\}$
- $\{\text{one}, \text{two}, \text{three}\}$
- palindromes
- legal Java reserved words?
- legal Java identifiers?

Definition of a Language

Definition

A language over alphabet Σ is a set of strings over Σ.

Example

Some languages over $\Sigma_2 = \{a, b, c, \cdots, z\}$

- $\{a, aa, aaa, aaaa\}$
- $\{\text{one}, \text{two}, \text{three}\}$
- palindromes
- legal Java reserved words?
- legal Java identifiers?
- English-language words (as per OED)?

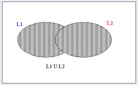
Set Operations

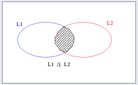
Definition

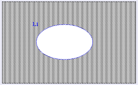
Intersection A string belongs to the union $L_1 \cup L_2$ if it belongs either to L_1 or to L_2.

Intersection A string belongs to the intersection $L_1 \cap L_2$ is it belongs to both L_1 and L_2.

Complement A string belongs to the complement $\bar{L_1}$ if it does not belong to L_1.







Example

$A = \{1, 2, 4\}$, B $= \{2, 3, 5\}$.

$$\begin{aligned} A \cup B &= \{1, 2, 3, 4, 5\} \\ A \cap B &= \{2\} \end{aligned}$$

Set Operations cont'd – Concatenation

Definition

A string x belongs to the concatenation $L_1 \cdot L_2$ if it has the form $x = x_1 \cdot x_2$, where x_1 and x_2 belong to L_1 and L_2 resp.

Set Operations cont'd – Concatenation

Definition

A string x belongs to the concatenation $L_1 \cdot L_2$ if it has the form $x = x_1 \cdot x_2$, where x_1 and x_2 belong to L_1 and L_2 resp.

Example

Let

$$\begin{aligned} L_1 &= \{\epsilon, 0, 1\} \\ L_2 &= \{00, 01, 10, 11\}, \end{aligned}$$

then

$$\begin{aligned} L_1 \cdot L_2 = \; & \{00, 01, 10, 11, \\ & 000, 001, 010, 011, \\ & 100, 101, 110, 111\} \end{aligned}$$

Examples

Example

If $B = \{0, 1\}$, then $B \cdot B = B^2 = \{00, 01, 10, 11\}$ *i.e.* all two bit binary strings.

Examples

Example

If $B = \{0, 1\}$, then $B \cdot B = B^2 = \{00, 01, 10, 11\}$ *i.e.* all two bit binary strings.

Example

If $D = \{0, 1, \cdots, 9\}$ and $L = \{a, b, c, \cdots, z\}$ then $L \cdot L \cdot D \cdot D \cdot D = L^2 D^3$

Examples

Example

If $B = \{0, 1\}$, then $B \cdot B = B^2 = \{00, 01, 10, 11\}$ *i.e.* all two bit binary strings.

Example

If $D = \{0, 1, \cdots, 9\}$ and $L = \{a, b, c, \cdots, z\}$ then $L \cdot L \cdot D \cdot D \cdot D = L^2 D^3$ all alphanumeric strings consisting of two letters followed by three digits.

Set Closure

Intuitively The closure L^* of L is the set of strings that can be expressed as the concatenation of zero or more (possibly different) strings from L.

Formally

Definition

Let

$$\begin{aligned} L^k &= \overbrace{L \cdot L \cdots L}^{k} \\ L^0 &= \{\epsilon\} \end{aligned}$$

then

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cdots = \bigcup_{k=0}^{\infty} L^k$$

Example

$L = \{0, 1\})$, $L^* =$

Set Closure

Intuitively The closure L^* of L is the set of strings that can be expressed as the concatenation of zero or more (possibly different) strings from L.

Formally

Definition

Let

$$\begin{aligned} L^k &= \overbrace{L \cdot L \cdots L}^{k} \\ L^0 &= \{\epsilon\} \end{aligned}$$

then

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cdots = \bigcup_{k=0}^{\infty} L^k$$

Example

$L = \{0, 1\})$, $L^* =$all binary strings (inluding ϵ)

Closure Examples

- $L = \{a, bb, ccc\}$
- L^* is an infinite set containing
  - ϵ
  - a, bb, ccc
  - a, aa, aaa, $aaaa$
  - ccc, $cccccc$, $ccccccccc$
  - abb, bba, $acccbb$
  - $ccc \cdot a \cdot a \cdot bb$

Closure Examples

- $L = \{a, bb, ccc\}$
- L^* is an infinite set containing
  - ϵ
  - a, bb, ccc
  - a, aa, aaa, $aaaa$
  - ccc, $cccccc$, $ccccccccc$
  - abb, bba, $acccbb$
  - $ccc \cdot a \cdot a \cdot bb$
  - What about bbb?

Closure Examples

- $L = \{a, bb, ccc\}$
- L^* is an infinite set containing
  - ϵ
  - a, bb, ccc
  - a, aa, aaa, $aaaa$
  - ccc, $cccccc$, $ccccccccc$
  - abb, bba, $acccbb$
  - $ccc \cdot a \cdot a \cdot bb$
  - What about bbb?
  - What about bab?

Set Expressions

- Let

$$
\begin{aligned}
L &= \{a, b, \cdots, z, A, B, \cdots, Z\} \\
D &= \{0, 1, 2, \cdots, 9\}
\end{aligned}
$$

- Pascal identifiers:

$$\overbrace{L}^{1} \cdot \overbrace{(L \cup D)^*}^{2}$$

  - 1: A single letter
  - 2: (followed by) zero or more letters or digits

Set Expressions

- Let

$$\begin{aligned} L &= \{a, b, \cdots, z, A, B, \cdots, Z\} \\ D &= \{0, 1, 2, \cdots, 9\} \end{aligned}$$

- Pascal identifiers:

$$\overbrace{L}^{1} \cdot \overbrace{(L \cup D)^*}^{2}$$

  - 1: A single letter
  - 2: (followed by) zero or more letters or digits
- Succint specification of lexical structure of Pascal identifiers!

What's Next?

- We introduce a concept called regular expressions akin to set expressions that can succintly capture certain kinds of patterns
- We develop techniques for the detection within text of patterns specified by regular expression
- Use these ideas to perorm lexical analysis in a compiler context.