$$(0 \cup 1)01^*$$

Theory of Computation

Lesson 4

Regular Expressions

Regular Language Tools

There are three tools under the regular language:

- Deterministic finite automata (DFA)√
- Non-deterministic finite automata (NFA) √
- Regular Expressions (RegEx)

We'll first review this tool and apply it to problem solving, then focus on its conversion to other tools.

Regular Operations

In this section, we define three operations on languages. We know that the set of all regular languages is closed under these three operations. These are:

- Union
- Concatenation
- Star

Regular Operations

1. The *union* of A and B is defined as

$$A \cup B = \{w : \; w \in A \text{ or } w \in B\}.$$

2. The *concatenation* of A and B is defined as

$$AB = \{ww' : \; w \in A \text{ and } w' \in B\}.$$

3. The *star* of A is defined as

$$A^* = \{u_1 u_2 \ldots u_k : \; k \geq 0 \text{ and } u_i \in A \text{ for all } i = 1, 2, \ldots, k\}$$

An Example

In the example below, you can see these three operations:

- Union: (0 U 1) is an example of union operation. Here, you can choose 0 or 1 to write.
- Concatenation: The expression glues three symbols, so this can be an example of concatenation.
- Star: The last symbol 1 in the expression has a star operation. It can produce many 1 symbols in sequence.

$$(0 \cup 1)01^*$$

Formal Definition

Definition 2.7.1 Let Σ be a non-empty alphabet.

1. ϵ is a regular expression.
2. $\emptyset$ is a regular expression.
3. For each $a \in \Sigma$, a is a regular expression.
4. If R_1 and R_2 are regular expressions, then $R_1 \cup R_2$ is a regular expression.
5. If R_1 and R_2 are regular expressions, then $R_1 R_2$ is a regular expression.
6. If R is a regular expression, then R^* is a regular expression.

A first example

$$(0 \cup 1)01^*$$

The language described by this expression is the set of all binary strings

1. that start with either 0 or 1 (this is indicated by $(0 \cup 1)$),
2. for which the second symbol is 0 (this is indicated by 0), and
3. that end with zero or more 1s (this is indicated by 1^*).

That is, the language described by this expression is

$$\{00, 001, 0011, 00111, \ldots, 10, 101, 1011, 10111, \ldots\}.$$

A second example

The language {w : w contains exactly two 0s} is described by the following expression

1*01*01*

In this example, we should guarantee only two zeros in anywhere in the string. Each 1* can produce ε or only 1 or more 1s. Therefore, it can overcome the problem and generate all the following strings.

00,100,1100,…,1010,10110,…,10101,101011,…, 010, …

A second example

Let's take a look at how the same regular expression generates words. Here each star operation can turn into independent numbers.

1*01*01*

$1^a01^b01^c$

a	b	c	string
0	0	0	00
0	0	1	001
0	0	2	0011
0	1	0	010
0	2	0	0110
1	0	0	100
3	2	4	11101101111

A third example

The language {w : w contains at least two 0s} is described by the following expression

$$(0 \cup 1)^* \, 0 \, (0 \cup 1)^* \, 0 \, (0 \cup 1)^*$$

In this example, we should guarantee at least two zeros in anywhere in the string. In the answer of the previous question, 1*'s will be replaced with (0 U 1)*.

A fourth example

The language {w : 1011 is a substring of w} is described by the expression

$$(0 \cup 1)^* \, 1011 \, (0 \cup 1)^*$$

Here, when we guarantee 1011 as a substring, it doesn't matter. The string can start and also end with any substring.

A other examples

The language $\{w : \text{ the length of } w \text{ is even}\}$ is described by the expression

$$((0 \cup 1)(0 \cup 1))^* .$$

The language $\{w : \text{ the length of } w \text{ is odd}\}$ is described by the expression

$$(0 \cup 1)\,((0 \cup 1)(0 \cup 1))^* .$$

The language $\{1011, 0\}$ is described by the expression

$$1011 \cup 0.$$

The language $\{w : \text{ the first and last symbols of } w \text{ are equal}\}$ is described by the expression

$$0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1.$$

- That’s all.
- Thanks for listening.