Lesson 4: Regular expressions

Theme: Regular expressions as alternative description of regular languages.

In the following we fix an alphabet Σ. Regular expressions (over Σ) are expressions built inductively as follows.

- ∅ is a regular expression.
- a is a regular expression, for every symbol a ∈ Σ.
- If e₁, e₂ are regular expressions, then so are \((e₁ \cdot e₂)\) and \((e₁ \cup e₂)\).
- If e is a regular expression, then so is \((e)\)\(^*\).

A regular expression \(e\) over Σ defines a language, denoted by \(L(e)\), over the same alphabet as follows.

- If \(e = \emptyset\), then \(L(e) = \emptyset\).
- If \(e = a\), where \(a \in \Sigma\), then \(L(e) = \{a\}\).
- If \(e = (e₁ \cdot e₂)\), where \(e₁\) and \(e₂\) are regular expressions, then \(L(e) = L(e₁) \cdot L(e₂)\).
- If \(e = (e₁ \cup e₂)\), where \(e₁\) and \(e₂\) are regular expressions, then \(L(e) = L(e₁) \cup L(e₂)\).
- If \(e = (e₁)\)\(^*\), where \(e₁\) is a regular expression, then \(L(e) = L(e₁)\)\(^*\).

Usually, we omit writing \(\cdot\) in \((e₁ \cdot e₂)\), and instead, we simply write \((e₁e₂)\). Also, when there is no ambiguity, we will omit writing the brackets and simply write \(e₁e₂\) and \(e₁\)\(^*\), instead of \((e₁ \cdot e₂)\) and \((e₁\)\(^*\).

The following theorem states that the class of languages defined by regular expressions is exactly the class of regular languages.

**Theorem 4.1** Regular expressions define precisely the class of regular languages. That is, for every regular expression \(e\) over Σ, \(L(e)\) is a regular language, and vice versa, for every DFA \(A\), there is a regular expression \(e\) such that \(L(e) = L(A)\).

Combining what we have learnt so far, we obtain three different, but equivalent, characterisations of regular languages, as stated below.

**Corollary 4.2** Let \(L\) be a language. Then, the following are equivalent.

- \(L\) is accepted by a DFA.
- \(L\) is accepted by an NFA.
- \(L\) is defined by a regular expression.

**Remark 4.3** The term regular expressions are commonly abbreviated as *regex*. In most literatures and websites, the term “regex” are used more often than “regular expression.” Due to its widespread applications, many modern programming languages now include libraries for regex. The following are some of them.

- Java: [https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html](https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html)
- Python: [https://docs.python.org/2/library/re.html](https://docs.python.org/2/library/re.html)