Cheatsheet - Formal Languages & Regular Expressions

Fabio Lama – fabio.lama@pm.me

NOTE Make sure to check the *Automata Theory* cheatsheet, too.

1. Regular Operations

Let L_1 and L_2 be languages. The following operations are regular operations:

- Union: $L_1 \cup L_2 = \{x \mid x \in L_1 \quad \text{or} \quad x \in L_2\}$
- Concatenation: $L_1 \circ L_2 = \{xy \mid x \in L_1 \quad \text{and} \quad y \in L_2\}$
- Star: $L_1^* = \{x_1 x_2 ... x_m \mid m \geq 0, \text{each} \quad x_1 \in L_1\}$

For example:

$$A = \{a, b, c\}$$
$$B = \{x, y, z\}$$
$$A \cup B = \{a, b, c, x, y, z\}$$
$$A \circ B = \{ax, ay, az, bx, by, bz, cx, cy, cz\}$$
$$A^* = \{\varepsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, ...\}$$

Where ε means "none".

1.1. Properties

1.1.1. Union

$$A \cup B = B \cup A \quad (\text{commutative})$$
$$(A \cup B) \cup C = A \cup (B \cup C) \quad (\text{associative})$$
$$A \cup \emptyset = A$$
$$A \cup A = A \quad (\text{idempotent})$$

1.1.2. Concatenation

$$(A \circ B) \circ C = A \circ (B \circ C) \quad (\text{associative})$$
$$A \circ \varepsilon = \varepsilon \circ A = A$$
$$A \circ \emptyset = \emptyset$$

Note that concatenation is not commutative.

1.1.3. Concatenation and Union

$$(A \cup B) \circ C = (A \circ C) \cup (B \circ C)$$
$$A \circ (B \cup C) = (A \circ B) \cup (A \circ C)$$

1.1.4. Kleene Star

$$\emptyset^* = \{\varepsilon\}$$
$$\varepsilon^* = \varepsilon$$
$$(A^*)^* = A^*$$
$$A^* \circ A^* = A^*$$
$$(A \cup B)^* = (A^* B^*)^*$$

2. Regular Expressions

2.1. Atomic Expressions

The empty language, $\emptyset$, is a regular expression, which is the empty regular language. Any letter $a \in \Sigma$ is a regular expression and its language is $\{a\}$. Empty string, ε, is a regular expression representing the regular language $\{\varepsilon\}$.

- *Concatenation*: if R_1 and R_2 are regular expressions, so is $R_1 \circ R_2$.
- *Union*: if R_1 and R_2 are regular expressions, so is $R_1 \cup R_2$.
- *Kleene star*: if R is a regular expression, so is R^*.

2.1.1. Examples

What is the language of ab^*?

$$\{a, ab, abb, abbb, \ldots\}$$

What is the language of $ab^* \cup b^*$?

$$\{a, ab, abb, abbb, \ldots\} \cup \{\varepsilon, b, bb, bbb, \ldots\} = \{\varepsilon, a, b, ab, bb, abb, bbb, \ldots\}$$

What is the language of $ab^+ \cup b^+b$?

$$\{ab, abb, abbb, \ldots\} \cup \{bb, bbb, bbbb, \ldots\} = ab^* \cup b^* / \{a, \varepsilon, b\} \quad (\text{excluding} \quad a, \varepsilon, b)$$

What is the language of $\Sigma^* a$?

$$\{a, aa, ba, aaa, aba, baa, bba, \ldots\}$$

What is the language of $\Sigma^* a \Sigma^*$?

$$\{a, aa, ab, ba, aaa, aab, aba, abb, baa, bab, bba, \ldots\}$$

2.2. Regular & Non-Regular Languages

- A language is **regular** if it can be accepted by a **finite automata**.
- A language is **regular** is it can be accepted by a **regular expression**.
- Every finite language is regular.

Examples of a **non-regular** language:

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$
$$L = \{xx \mid x \in \{a, b\}^*\}$$
$$L = \left\{a^{n!} \mid n \in \mathbb{N}\right\}$$

2.3. Pumping Lemma

The Pumping Lemma proves whether a language is regular or not. If L is a regular language, then there is a number p (the pumping length) where, if s **is any string in** L **of length at least** p, then s may be divided into three pieces, $s = xyz$, satisfying the following conditions:

- $\forall i \geq 0, xy^i z \in L$
- $|y| > 0$
- $|xy| \leq p$

This means that if the language is finite, it is regular, we choose p to be the number of states in the finite automata representing L and if $|s| \geq p$, s must have a repeated state (Pigeonhole Principle).

For example, let's prove $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular:

- Assuming L is regular. Let p be the pumping length.
- Let $s = a^p b^p$, $|s| > p$
- Pumping Lemma: $s = xyz$
- For any i, $xy^i z \in L$. Let us try $i = 2$
- Cases:
  - 1.) y is only a's. $xyyz$ will have more a's than b's
  - 2.) y is only b's. $xyyz$ will have more b's than a's
  - 3.) y has a's and b's. $xyyz$ will have a's and b's jumbled up.
- Respectively, L is **not** regular.

Last updated 2023-01-02 20:27:47 UTC