

* Admin: Final exam on 8 Nov \rightarrow Move on Wattle soon

* Recap: Alphabets Σ (Typically $\{0,1\}$)

Languages $L \subseteq \Sigma^*$

Operations: concatenation, union, intersection,
star

* Star of a language (Kleene star)

L a language on some Σ

$$L^* = \{\epsilon\} \cup \{w_1, w_2, \dots, w_n \mid w_i \in L\}$$

E.g. $L = \{00, 110\}$

$$L^* = \{\epsilon, 00, 110, 0000, 00110, 11000, 110110, \dots\}$$

Note: $\Sigma \subseteq \Sigma^*$, so Σ itself is a language

Σ^* in the ~~above~~ above sense is the same as the Σ^* we discussed earlier.

* Dictionary order. (a total order on Σ^*)

Given an ordering on Σ , for example, $(0,1)$

We can produce a total order on Σ^* , called the dictionary order.

Let $v = a_1 \dots a_m \leftarrow$ length m

$w = b_1 \dots b_n \leftarrow$ length n

If $m < n$ then $v < w$

If $n < m$ then $w < v$

If $m = n$: start by comparing a_1 & b_1

If $a_1 < b_1$, $v < w$

If $b_1 < a_1$, $w < v$

If $a_1 = b_1$, compare a_2 & b_2 and so on.

[dictionary ordering]

Back to * (Star)

$L = \emptyset$, then $L^* = \{\epsilon\}$

$L = \{\epsilon\}$, then $L^* = \{\epsilon\}$

L^* is infinite except the two cases above [check.]

$$L^* = \{\epsilon\} \cup L \cup (L \circ L) \cup (L \circ L \circ L) \cup (L \circ L \circ L \circ L) \cup \dots$$

* Use this stuff for pattern-matching. (regular expr)

Fix alphabet Σ .

A regular expression is a word made up of symbols from Σ as well as certain additional symbols, which follow some recursive rules.

Three basic constructors (base cases) (for regexes)

① $r = a$, for some $a \in \Sigma$

② $r = \epsilon$ (empty string)

③ $r = \phi$ (empty set)

E.g. $\Sigma = \{0, 1\}$, the following are valid regular expressions:

- 0
- 1
- ϵ
- ϕ

REG EX = regular expression

Three inductive / recursive constructors for regexes r :

④ $r = r_1 r_2$, e.g. $r = 00, 01, 1110$
where r_1 and r_2 are valid regexes.

⑤ $r = r_1 \mid r_2$, e.g. $r = 0 \mid 110, r = 101 \mid 11 \mid 1100$
(OR)
where r_1, r_2 are valid regular expressions.

⑥ $r = (r_1)^*$, where r_1 is a valid regular expression
E.g. $(010)^*$

Note: Parentheses are used for grouping.

③

* Order of operations

When parsing regular expressions, order is:

- ① brackets / parentheses first
- ② star next
- ③ concatenation next
- ④ OR last

* Examples: $\Sigma = \{0, 1\}$

An exercise in parsing a complicated regex.

$$r = (01 \mid \phi^* \mid 111)^* 010 (0 \mid 1 \mid \epsilon \mid 00)$$

(Analogy: $(x + 5y + 25z^2)^3 (56x^2 + 12y^2)$)

$$r = \underbrace{(01 \mid \phi^* \mid 111)^*}_{r_1} \underbrace{010}_{r_2} \underbrace{(0 \mid 1 \mid \epsilon \mid 00)}_{r_3} \quad \text{[concatenation constructor]}$$

$$r_1 = (01 \mid \phi^* \mid 111)^* = S^*, \text{ where } S = 01 \mid \phi^* \mid 111 \quad \text{[star constructor]}$$

$$r_2 = 010$$

$$r_3 = (0 \mid 1 \mid \epsilon \mid 00)$$

$$S = \underbrace{01}_{s_1} \mid \underbrace{\phi^*}_{s_2} \mid \underbrace{111}_{s_3} \quad \text{[OR constructor]}$$

$$s_1 = 01 \rightarrow \text{concatenation of } 0 \text{ \& } 1$$

$$s_2 = \phi^*$$

$$s_3 = 111$$

④

⑤

More examples $\Sigma = \{0, 1\}$

$r = 0, r = 1, r = \varepsilon, r = \phi$.

$r = 00, 101, \varepsilon|01|\phi, (\varepsilon|01^*|\phi)^* 11$

etc.