MATH 2301 25 Sept 2020

* Regular expressions & finite automata.

We have:

- An alphabet Σ, e.g. $\Sigma = \{0,1\}$ or $\Sigma = \{a, b, \dots, z\}$ (finite set of symbols, called letters of Σ) (or words)
- Strings are finite ordered lists of elements of Σ, written without spaces or punctuation.

  E.g. if $\Sigma = \{0,1\}$ then 10, 00, 1110, 0, ε, 1 all strings.

- The empty or null string ε is the unique string with no elements, or length 0.
- Σ^* = set of all strings, including ε

  If $\Sigma = \emptyset$ then $\Sigma^* = \{\varepsilon\}$, otherwise Σ^* is infinite.

- A language L is any subset of Σ^*, including $\emptyset \subseteq \Sigma^*$ and $\Sigma^* \subseteq \Sigma^*$
- The length of $w \in \Sigma^*$ is the number of letters in w.

* Basic constructions

- Concatenation (on strings): If $v = a_1 a_2 \cdots a_k$ & $w = b_1 b_2 \cdots b_\ell$ are both strings ($a_i, b_j \in \Sigma$), then their concatenation vw is the string $a_1 a_2 \cdots a_k b_1 b_2 \cdots b_\ell$.
- Concatenation (on languages): Let L_1, L_2 be languages; ie $L_1, L_2 \subseteq \Sigma^*$.

Then $L_1 \circ L_2$ is the <u>concatenated</u> language:

$L_1 \circ L_2 = \{ xy \mid x \in L_1, y \in L_2 \}$.

<u>E.g.</u> $L_1 = \{0, 1\}$, $L_2 = \{101, 111\}$

$L_1 \circ L_2 = \{0101, 0111, 1101, 1111\} \neq L_2 \circ L_1$

<u>E.g.</u> $L_1 = \{\varepsilon, 0, 1\}$, $L_2 = \{101, 111\}$

$L_1 \circ L_2 = \{0101, 0111, 1101, 1111, 101, 111\}$

<u>E.g.</u> $L_1 = \{0, 1\}$, $L_2 = \phi$

$L_1 \circ L_2 = \phi$.

$L_1 = \{\varepsilon, 0\}$, $L_2 = \{\varepsilon, 1\}$

$L_1 \circ L_2 = \{\varepsilon, 0, 1, 01\}$.

* <u>Note</u>: $L = \phi$ is the empty language containing <u>no</u> strings. $L = \{\varepsilon\}$ is not the empty language, but only contains the empty string ε.

$\varepsilon \neq \phi$.

- <u>Union</u>: If $L_1, L_2 \subseteq \Sigma^*$, then $L_1 \cup L_2$ is the set union. So, $L_1 \cup L_2 = \{ x \mid x \in L_1 \text{ or } x \in L_2 \}$.

<u>E.g.</u> $L_1 = \{0, 1\}$, $L_2 = \{101, 111\}$

$L_1 \cup L_2 = \{0, 1, 101, 111\}$

<u>E.g.</u> $L_1 = \{0, 1\}$, $L_2 = \phi$ then $L_1 \cup L_2 = \{0, 1\}$.

- <u>Star</u>: Let $L \subseteq \Sigma^*$. Then L^* is the following:

$L^* = \{ x_1 x_2 \cdots x_k \mid k \geq 0, \text{ and } x_i \in L \text{ for each } i \}$

<u>E.g.</u> $L = \{0\}$

$L^* = \{\varepsilon, 0, 00, 000, \ldots, 0^k, \ldots\}$ ($0^k = 000\cdots0$ (k times))

<u>E.g.</u>: $L = \phi$ then $L^* = \{\varepsilon\}$

<u>E.g.</u>: $L = \{0, 11\}$, $L^* = \{\varepsilon, 0, 11, 011, 00, 1111, 11000, 011000111, \ldots\}$

$L = \{\varepsilon\}$; $L^* = \{\varepsilon\}$.

- <u>Lexicographic order</u> (dictionary order)

Suppose that we have ordered the elements of Σ.

Given $L \subseteq \Sigma^*$, the lexicographic ordering on L is such that

① shorter words $\leq$ longer words (first check)

② For any length k, order the strings of length k in dictionary order: (second check)

if $v = a_1 a_2 \cdots a_k$, $w = b_1 b_2 \cdots b_k$, then

$v < w$ if : either $a_1 < b_1$ or

$a_1 = b_1$ & $a_2 < b_2$, or

$a_1 = b_1$, $a_2 = b_2$ and $a_3 < b_3$, ... etc.

Pattern-matching with <u>regular expressions</u>

<u>Definition</u> : A valid regular expression is either:

① ϕ , or

② ε , or

③ a for $a \in \Sigma$, or

④ rs for any valid regular expressions r, s

⑤ $r|s$ for any valid regex r, s

⑥ r^* for any valid regex r.

<u>E.g.</u> $\Sigma = \{0,1\}$: $((0|1)^*)^* \mid 111 \mid \varepsilon \mid 10$

is a valid regex.

* <u>Note</u>: (|) is associative, so I omit parentheses.

<u>E.g.</u> $00^* 11^* (01 | 10 | \varepsilon | 1111) = 0(0^*)1(1^*)(01|10|\varepsilon|1111)$

* Order of operations:

$*$ has priority over concatenation

concat. has priority over $|$

Rule: All words of Σ^* are valid regex.

Prop: Every regular expression r corresponds to a language; we call it $L(r)$.

	Regular expression	$\longleftrightarrow$	Languages
①	ϕ	$\longleftrightarrow$	$L(\phi) = \phi$
②	ε	$\longleftrightarrow$	$L(\varepsilon) = \{\varepsilon\}$
③	a for $a \in \Sigma$	$\longleftrightarrow$	$L(a) = \{a\}$
④	rs	$\longleftrightarrow$	$L(rs) = L(r) \circ L(s)$
⑤	$r \mid s$	$\longleftrightarrow$	$L(r \mid s) = L(r) \cup L(s)$
⑥	r^*	$\longleftrightarrow$	$L(r^*) = L(r)^*$.

* **Remark/definition**: We say that a string $w \in \Sigma^*$ matches a regular expression r if $w \in L(r)$.

E.g. $(01)^* = r$; some words that match r:

01, 0101, 01010101, ε

Eg. $1(000 \mid 111)^* 0 = r$; some words that match r:

10000, 11110, 10, 11110000

E.g.: $1^*(10 \mid 01)^* = r$

ε, $1111\,1001011001$