

## MATH 2301

- \* Regular expressions (and pattern-matching)

Let  $\Sigma$  be an alphabet

\*\* Defn: A regular expression (or regex)  $r$  is a string in the letters of  $\Sigma$ , together with the symbols "|", "\*", and " $\phi$ ", satisfying one of the following:

- 1)  $r = \phi$
- 2)  $r = \epsilon$
- 3)  $r = a$  for some  $a \in \Sigma$
- 4)  $r = r_1 r_2$  for regexes  $r_1$  and  $r_2$  *on concatenation*
- 5)  $r = r_1 | r_2$  for regexes  $r_1$  and  $r_2$  *on "or"*
- 6)  $r = r_1^*$  for a regex  $r_1$ . *on star*

Additionally, we can use parentheses "(" & ")" to signify grouping.

(Just like in an algebraic expression.)

\*\* We assume that  $\Sigma$  does not contain \*, |,  $\phi$ ,  $\epsilon$ , (, ).

\*\* Order of operations

Brackets are subexpressions, so they come first.

Apply \* first, then concatenation, and then or.

Concatenation & | are associative, so we don't need to bracket multiple concatenations or multiple "or's

## \*\* Examples

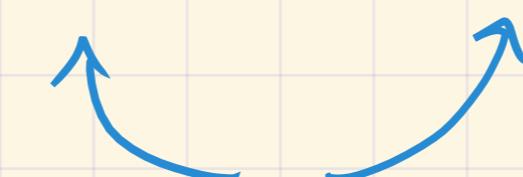
$$\Sigma = \{0, 1\}$$

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

$$r = \phi^*, \quad r = \varepsilon^*$$

$$r = 0^*$$

$$r = 01, \quad r = \varepsilon 1 \quad r = 1$$



different, but they'll have the same meaning

$$r = (01 | \phi^* | 110)^* 010 (01 | \varepsilon)$$

How to parse?

Mentally, break it up as:  $r = r_1 r_2 r_3$

$$r_1 = (\dots)^*$$

$$01 | \phi^* | 110$$

an or of 3  
subexpressions

(keep going)

$r_2 = 010 = \text{concatenation of } 0, 1, 0$

$r_3 = 01 | \varepsilon = \text{or of } 0, 1, \varepsilon$

## \*\* Matching

A word  $w \in \Sigma^*$  is said to match a regex  $r$  if one or more of the following hold.

- 1)  $r = \epsilon$  and  $w = \epsilon$
- 2)  $r = a$  and  $w = a$ , for some  $a \in \Sigma$
- 3)  $r = r_1 r_2$  and  $w$  can be written as  $w = xy$ , where  $x$  and  $y$  are words, and  $x$  matches  $r_1$  and  $y$  matches  $r_2$ .
- 4)  $r = r_1 | r_2$  and  $w$  either matches  $r_1$ , or matches  $r_2$ , or matches both.
- 5)  $r = r_1^*$  and either  $w = \epsilon$ , or  $w = x_1 x_2 \dots x_k$ , where each  $x_i$  is a word, and each  $x_i$  matches  $r_1$ .

## \*\* Examples . Let $\Sigma = \{0, 1\}$

- 1)  $r = 0$  :  $w = 0$  is the only string that matches
- 2)  $r = 1$  :  $w = 1$  is the only string that matches
- 3)  $r = \epsilon$  :  $w = \epsilon$  is the only string that matches.
- 2)  $r = \phi$  : no strings match this.
- 3)  $r = 01$  :  $w = 01$  is the only string that matches
- 4)  $r = \phi 1$  : no strings match this

5)  $r = 0 \mid 1$  :  $\omega = 0, \omega = 1$  match.

6)  $r = 1^*$  :  $\omega = \epsilon, \omega = 1, \omega = 11, \omega = 11111$  etc  
match.

7)  $r = (00 \mid 11)^*$

Match:  $\omega = 00, \omega = 11, \omega = 000000, \omega = 111111$

$\omega = \epsilon, \omega = \underline{\underline{00}}\underline{\underline{11}}\underline{\underline{11}}\underline{\underline{00}}\underline{\underline{11}}$



each of these  
match  $(00 \mid 11)$

(and many others...)

8)  $r = 0(1 \mid 0)^*1$

$\omega$  starts with a 0 and ends with a 1.

these are exactly all the words that match.

$$\omega = 0 \epsilon 1 = 01$$

↑  
matches  $(01)^*$

## **\*\* The language of a regex**

Let  $r$  be a regex. The language of  $r$ , denoted  $L(r)$  is the set of all words that match  $r$ .

### **\*\*\* Example**

$$r = 0(1|0)^*1$$

$L(r) = \{ w \in \Sigma^* \mid w \text{ begins with a } 0 \text{ and ends with a } 1 \}$

**\*\* Question** : Given some  $L \subseteq \Sigma^*$ , is there a regular expression  $r$  such that  $L(r) = L$  ?

### **\*\*\* Example**

$L \subseteq \Sigma^*$ ,  $L = \{ w \mid w \text{ begins with a } 0 \text{ or ends with a } 1 \}$ .

(Homework ... finish next time)