

COMS 3261 - Computer Science Theory

Lecture 3, Part 2: Regular Expressions

Idea: We can use the regular operations ($\cup, \circ, *$) to build up expressions describing languages.

Example: $(0 \cup 1)0^*$

$\{0, 1\}$ (hidden concatenation)
 $\{0\}$ the language $\{0\}$
 $\{1\}$ the language $\{1\}$
 $\{0^*\}$ zero or more concatenated 0's.
 $(\Sigma = \{0, 1\})$

Read this: 'either 0 or 1' followed by some number of zeroes.
This language contains: 00, 10, 0, 1, 100000, 0000...

not: 101

Def. We say that a regular expression evaluates to the language of strings it describes.

$$(0 \cup 1)0^* = \{w \mid w \text{ is a string starting with } 0 \text{ or } 1, \text{ followed by some number of } 0\text{'s}\}$$

Example: $(0 \cup 1)^*$?

$$= \{ \text{language of all binary strings} \}$$

Def. (Regular Expressions, formally.)

(Idea: We'll define inductively - because regular expressions can be long and complicated, we'll recursively describe how to build one.)

We say that R is a regular expression if:

- $R = a$, for some $a \in \Sigma$. ($\{a\}$)
- $R = \epsilon$ ($\{\epsilon\}$)

$$- R = \emptyset \quad (\{ \})$$

• $- R = R_1 \cup R_2$, where R_1 and R_2 are regular expressions.

• $- R = R_1 \circ R_2$

$- R = R_i^*$

$$R = (\emptyset \cup 1) \emptyset^*$$

$$\begin{array}{c} \swarrow \quad \searrow \\ R_1: (\emptyset \cup 1) \quad R_2: \emptyset^* \end{array}$$

$$\begin{array}{c} \swarrow \quad \searrow \\ R_4: \emptyset \quad R_5: 1 \quad \bigg| \quad (R_3: \emptyset)^* \end{array}$$

Some shorthand for regular expressions:

$- \Sigma :=$ 'any symbol in the alphabet Σ '

(Example: if $\Sigma = \{0, 1\}$, then $\Sigma := (\emptyset \cup 1)$)

$- R^+ :=$ 'one or more copies of R , concatenated'

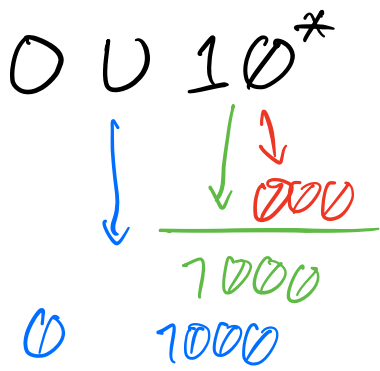
$$R^+ = RR^*$$

$-$ For $k \in \mathbb{N}$, we write $R^k :=$ 'k copies of R , concatenated together.'

(Example: $(\emptyset 1)^3 = \emptyset 1 \emptyset 1 \emptyset 1$.)

Order of operations:

- 1) Kleene star
- 2) Concatenation
- 3) Union.



When in doubt, use parentheses!

(Lots of) Examples.

$$0^* 1 0^* = \{w \mid w \text{ contains a single } 1\}$$

$$1^* (01^*)^* = \{w \mid w \text{ consists of } \begin{array}{l} \text{zero or more } 1\text{'s,} \\ \text{followed by zero or more substrings consisting} \\ \text{of one zero and one or more } 1\text{'s.} \end{array}\}$$

$$= \{w \mid \text{every } 0 \text{ in } w \text{ is followed by at least one } 1.\}$$

$$(\Sigma \Sigma \Sigma)^* = \{w \mid \frac{\text{length of } w}{|w|} \text{ is a multiple of } 3\}$$

$$\underline{0 \Sigma^* 0} \cup \underline{1 \Sigma^* 1} \cup \underline{0} \cup \underline{1}$$

$$= \{w \mid w \text{ starts and ends with a } 1, \text{ or starts and ends with a } 0.\}$$

$$(0 \cup \epsilon)(1 \cup \epsilon) = \{01, 1, 0, \epsilon\}$$

$$\underbrace{1^*}_{\text{pink}} \emptyset := \emptyset$$

$$\emptyset^* := \{\epsilon\}$$

Let $\Sigma = \{-, ., 0, 1, \dots, 9\}$, $D := \{0, \dots, 9\}$

$$(\epsilon \cup -)(D^+ \cup D^+ \cdot D^+) =$$

numbers like 103, -92.9, 48.666, 0.7, ...

Next: Converting regular expressions to NFAs.