Regular Expressions I

- Example

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

- This is a simplification of

$$(\{0\} \cup \{1\}) \circ \{0\}^*$$

- Regular expressions are practically useful
  Example: finding lines in a file containing "a" or "b"
  `egrep '(a|b)' file`

Regular Expressions II

- Formally this means we consider the language

$$\Sigma^* a \Sigma^* \cup \Sigma^* b \Sigma^*$$

  Σ^*: all strings over Σ
- Example

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

  all strings by 0 and 1
- Example:

$$(0\Sigma^*) \cup (\Sigma^* 1)$$

  all strings start with 0 or end with 1

Formal definition of regular expression I

- R is a regular expression if it is one of the following expressions
    1. a, where $a \in \Sigma$
    2. ϵ $(\epsilon \notin \Sigma)$
    3. $\emptyset$
    4. $R_1 \cup R_2$, where R_1, R_2 are regular expressions
    5. $R_1 \circ R_2$, where R_1, R_2 are regular expressions
    6. R_1^*, where R_1 is a regular expression
- $\emptyset$ and ϵ

  ϵ: empty string

Formal definition of regular expression II

$\emptyset$: empty language (language without any string)

$$(0 \cup \epsilon)1^* = 01^* \cup 1^*$$
$$(0 \cup \emptyset)1^* = 01^*$$
$$\emptyset 1^* = 1^* \emptyset = \emptyset$$

Concatenating 1^* with nothing $\Rightarrow$ nothing

- We have an inductive definition

  An expression is constructed from smaller strings

Examples I

- 0^*10^*: strings with exactly one 1
- $(\Sigma\Sigma)^*$: strings with even length
- Assume $\Sigma = \{0, 1\}$

$$0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1$$

  Strings that start and end with the same symbol
- $\emptyset^* = \{\epsilon\}$
- $R \cup \emptyset = R$
- $R \circ \epsilon = R$

Floating number in language I

$$(+ \cup - \cup \epsilon)(DD^* \cup DD^*.D^* \cup D^*.DD^*),$$

where

$$D = \{0, \ldots, 9\}$$

- 72, 2.1, 7., -.01

$$72 \in DD^*$$
$$2.1 \in DD^*.D^*$$
$$7. \in DD^*.D^*$$
$$.01 \in D^*.DD^*$$

Floating number in language II

- Why not $D^*.D^*$

  . is not allowed

Equivalence with finite automata I

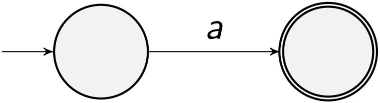
- They have equivalent descriptive power
- language regular $\Leftrightarrow$ described by regular expression

Lemma 1.55 I

- Language by a regular expression
  $\Rightarrow$ regular (described by an automaton)
- The proof is by induction. We go through all cases in the definition
- $R = a \in \Sigma$

  Can such a language be recognized by an NFA?

  This language has only one string and can be recognized by



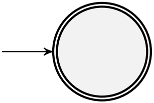
Lemma 1.55 II

Note that this is an NFA.

Formal definition:

$$N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$$
$$\delta(q_1, a) = \{q_2\}$$
$$\delta(r, b) = \emptyset, r \neq q_1 \text{ or } b \neq a$$

- $R = \epsilon$

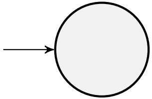


Lemma 1.55 III

Formal definition

$$N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$$
$$\delta(q_1, a) = \emptyset, \forall a$$

- $R = \emptyset$



Lemma 1.55 IV

Formal definition

$$N = (\{q\}, \Sigma, \delta, q, \emptyset)$$
$$\delta(r, a) = \emptyset, \forall r, a$$

Note: earlier we only say $F \subset Q$, so F can be $\emptyset$

- For the other three situations

$$R = R_1 \cup R_2$$
$$R = R_1 \circ R_2$$
$$R = R_1^*$$

we use the proof in NFAs

Lemma 1.55 V

- We will see details by an example