Regular languages

Peter Mayr

Computability Theory, January 22, 2021

Regular expressions

Definition

The set of **regular expresssions** over Σ is defined inductively by

- $\emptyset, \epsilon$ are regular;
- a is regular for every $a \in \Sigma$;
- if r_1, r_2 are regular, then also $r_1 r_2$ and $r_1 + r_2$;
- if r is regular, then also r^*.

Example

$(a + (b(c^*)))$ is regular over $\Sigma = \{a, b, c\}$, usually denoted $a + bc^*$.

Note

- Regular expressions are just strings of symbols.
- Parenthesis are used when necessary for parsing.
- **Convention:** * binds stronger than $\cdot$, $\cdot$ stronger than $+$.

Semantics

Definition

The **language** $L(r) \subseteq \Sigma^*$ **of a regular expression** r is defined inductively by

- $L(\emptyset) := \emptyset$, $L(\epsilon) := \{\epsilon\}$
- $L(\mathrm{a}) := \{a\}$ for $a \in \Sigma$
- $L(r_1 r_2) := \{w_1 w_2 \ : \ w_1 \in L(r_1), w_2 \in L(r_2)\}$ concatenation
  $L(r_1 + r_2) := L(r_1) \cup L(r_2)$ union
- $L(r^*) := L(r)^* := \underbrace{L(r)^0}_{\{\epsilon\}} \cup \underbrace{L(r)^1}_{L(r)} \cup L(r)^2 \cup \ldots$ **Kleene star**

Example

- $L(a + bc^*) = \{a\} \cup \{bc^n \ : \ n \in \mathbb{N}\}$
- regular expression for words ending in 01: $(0+1)^*01$
- regular expression for words in which 0 and 1 alternate: $(1+\epsilon)(01)^*(0+\epsilon)$

Regular languages and automata

Definition

$L \subseteq \Sigma^*$ is **regular** if $L = L(r)$ for some regular expression r over Σ.

Theorem

$L \subseteq \Sigma^*$ is regular iff $L = L(M)$ for some DFA M with input alphabet Σ.

Proof

$\Rightarrow$: Given a regular expression r, it suffices to build an ϵ-NFA M with $L(M) = L(r)$ by induction on r:

- $r = \emptyset$ 

 $r = \epsilon$ 

- $r =$ a for $a \in \Sigma$

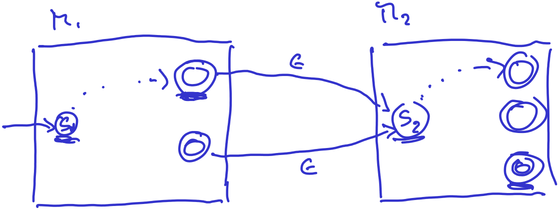
Proof $\Rightarrow$: Closure under concatenation

Let $r = r_1 r_2$. Assume ϵ-NFAs M_1, M_2 accept $L(r_1), L(r_2)$, resp. Compose M_1 and M_2 into a new ϵ-NFA M for r with

- states $Q_1 \cup Q_2$ (wlog Q_1, Q_2 are disjoint)
- the starting state s_1 of M_1
- the accepting states F_2 of M_2
- $\Delta = \Delta_1 \cup \Delta_2 \cup \{\epsilon\text{-transitions from } F_1 \text{ to } s_2\}$.

Then $L(M) = L(r_1 r_2)$.

Note: The only path from s_1 to F_2 is via an ϵ-transition from F_1 to s_2.



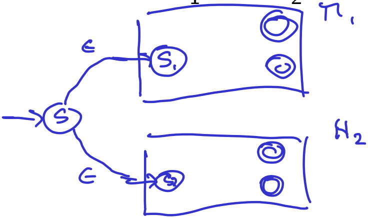
Proof $\Rightarrow$: Closure under union

Let $r = r_1 + r_2$. Assume ϵ-NFAs M_1, M_2 accept $L(r_1), L(r_2)$, resp.
Compose M_1 and M_2 **in parallel** into a new ϵ-NFA M with

- states $\{s\} \cup Q_1 \cup Q_2$ (disjoint union)
- a new starting state s
- accepting states $F_1 \cup F_2$
- $\Delta = \Delta_1 \cup \Delta_2 \cup \{\epsilon$-transitions from s to s_1 and to $s_2\}$.

Then $L(M) = L(r_1 + r_2)$.

Note: The only path from s to either F_1 or F_2 is via an ϵ-transition from s to s_1 or to s_2.



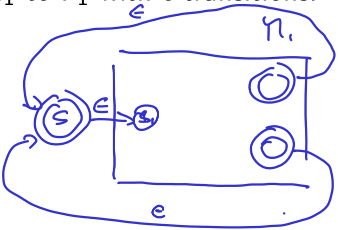
Proof $\Rightarrow$: Closure under *

Let $r = r_1^*$. Assume ϵ-NFA M_1 accepts $L(r_1)$.
Loop M_1 to get a new ϵ-NFA M with

- states $\{s\} \cup Q_1$ (disjoint union)
- a new starting state s
- new accepting states $\{s\}$ $F = F_1 \cup \{s\}$
- $\Delta = \Delta_1 \cup \{\epsilon\text{-transitions from } s \text{ to } s_1 \text{ and from } F_1 \text{ to } s_2\}$.

Then $L(M) = L(r_1^*)$.

Note: The only path from s to s is via ϵ or via concatenations of paths from s_1 to F_1 with ϵ-transitions.



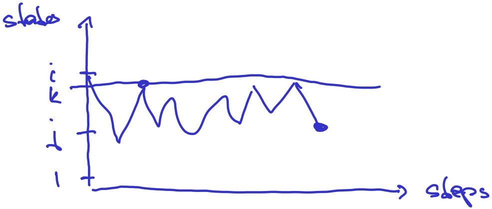
This completes the proof that every regular language is accepted by some ϵ-NFA (hence a DFA).

Proof $\Leftarrow$

Given a DFA M find a regular expression r such that $L(M) = L(r)$.

Assume M has states $\{1, \ldots, n\}$. For $i, j, k \leq n$ define

$$R^k_{ij} := \left\{ w \in \Sigma^* \ : \ \begin{array}{l} \delta^*(i, w) = j \text{ and all strictly intermediate states} \\ \text{on the path labelled by } w \text{ are } \leq k \end{array} \right\}$$



Claim $(\star)$**:** $R_{ij}^k = L(r_{ij}^k)$ for some regular r_{ij}^k.

Proof by induction on k:
Basis $k = 0$**:** No intermediate states on the path from i to j. Let

$$A := \{a \in \Sigma \ : \ \delta(i, a) = j\}$$

- For $i \neq j$, let

$$r_{ij}^0 := \begin{cases} \emptyset & \text{if } A = \emptyset \\ a_1 + \cdots + a_\ell & \text{if } A = \{a_1, \ldots, a_\ell\}, \ell \geq 1 \end{cases}$$

- For $i = j$, let

$$r_{ij}^0 := \begin{cases} \epsilon & \text{if } A = \emptyset \\ \epsilon + a_1 + \cdots + a_\ell & \text{if } A = \{a_1, \ldots, a_\ell\}, \ell \geq 1 \end{cases}$$

Induction step: Let $w \in R_{ij}^k, k \geq 1$.

- If k is not an intermediate point on the path described by w, then $w \in R_{ij}^{k-1}$.
- If w's path goes to k at least once, then

$$w \in R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$$

.

Hence $R_{ij}^k = R_{ij}^{k-1} \cup R_{ik}^{k-1}(R_{kk}^{k-1})^* R_{kj}^{k-1}$.
By induction assumption, R_{ij}^k is the language of the regular expression

$$r_{ij}^{k-1} + r_{ik}^{k-1}(r_{kk}^{k-1})^* r_{kj}^{k-1}$$

and Claim $(\star)$ is proved.

Let 1 the start state and F the final states of M.
For $k = n$, Claim $(\star)$ yields regular $r := \sum_{f \in F} r_{1f}^n$ such that $L(M) = L(r)$. $\square$