↳ Operations for REG can be applied to NFAs:

↳ Corresponding equations

$$L(A_1 . A_2) = L(A_1) . L(A_2) \quad \text{etc.}$$

important when we consider relationship with logics.

<u>First goal here:</u>

$L \in REG$ iff there is NFA A with $L = L(A)$.

<u>Theorem:</u>

Let A_1, A_2 NFAs.

Then

there is an NFA $A_1 . A_2$ with $L(A_1 . A_2) = L(A_1) . L(A_2)$

there is an NFA $A_1 \cup A_2$ with $L(A_1 \cup A_2) = L(A_1) \cup L(A_2)$.

<u>Proof:</u>

Exercise

<u>Theorem:</u>

Let A an NFA. There is an NFA A^* with $L(A^*) = L(A)^*$.

<u>Construction:</u>

Let $A = (Q, q_0, \to, Q_F)$

Then

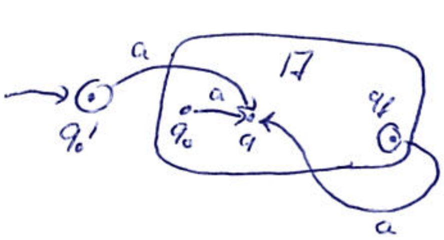
$$A^* := (Q \cup \{q_0'\}, q_0', \to \cup \to', Q_F \cup \{q_0'\})$$

where

$q_0' \xrightarrow{a}' q$ if $q_0 \xrightarrow{a} q$

$q_f \xrightarrow{a}' q$ —— " ——

<u>Illustration:</u>



· initial state not reachable (again)

· loop back from final states.

Theorem:

If $L \in REG$, then there is an NFA A with $L = L(A)$.

Show reverse:

Let A be an NFA, then $L(A)$ is regular.

Approach:

↳ Represent automaton with n states by (recursive) system of n equations.

↳ Solve this system.

↳ Relies on Arden's lemma

Lemma (Arden '60):

Let $U, V \subseteq \Sigma^*$ with $\varepsilon \notin U$.

Consider another language $L \subseteq \Sigma^*$.

Then

$$L = U.L \cup V \quad \text{iff} \quad L = U^*.V.$$

Remark:

„=>" Such an equation has a unique solution!

Proof:

„=>" Let $L \subseteq \Sigma^*$ so that $L = U.L \cup V$.

Claim: $L = U^*.V$

„⊇" Show that $U^*.V = \left(\bigcup_{i \in \mathbb{N}} U^i\right).V = \bigcup_{i \in \mathbb{N}} U^i.V \subseteq L$

Prove by induction that $U^n.V \subseteq L$ f.a. $n \in \mathbb{N}$.

IA: $U^0.V = \varepsilon.V = V \subseteq U.L \cup V \subseteq L$.

IS: Assume $U^n.V \subseteq L$.

Then

$$\mathcal{U}^{n+1}.V = \mathcal{U}.(\mathcal{U}^n.V)$$
$$\overset{(IH)}{\subseteq} \mathcal{U}.L$$
$$\subseteq \mathcal{U}.L \cup V \subseteq L.$$

Since $\mathcal{U}^n.V \subseteq L$ f.a. $n \in \mathbb{N}$,
we have $\bigcup_{i \in \mathbb{N}} \mathcal{U}^i.V \subseteq L$ □

„$\subseteq$" Assume $L \not\subseteq \mathcal{U}^*.V$
Then there is a shortest word $w \in L$ with
$$w \notin \mathcal{U}^*.V.$$
Since
$$L = \mathcal{U}.L \cup V,$$
we have
$$w \in \mathcal{U}.L \text{ or } w \in V.$$
We cannot have $w \in V$, otherwise $w \in \mathcal{U}^*.V$ ↯.
So
$$w \in \mathcal{U}.L, \text{ which means}$$
$$w = u.w' \text{ with } u \neq \varepsilon \text{ as } \varepsilon \notin \mathcal{U}.$$
Since
w is the shortest word in L with $w \notin \mathcal{U}^*V$, and
since w' is shorter,
we get
$$w' \in \mathcal{U}^*.V.$$
So
$$w = u.w' \in \mathcal{U}^*.V \text{ ↯}$$
Inclusion holds.

□

- Apply Arden's lemma as a tool to construct regular representation for an automaton.

<u>Theorem:</u>

- If L is recognised by an NFA,
then L is regular.

<u>Proof:</u>

Let $L = L(A)$ with $A = (Q, q_0, \rightarrow, Q_F)$.
W.l.o.g., $Q = \{q_0, \dots, q_{n-1}\}$
Define for every state q_i the language

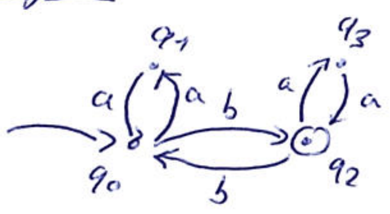
X_i = "words that have an accepting run from q_i in A."
$= L(Q, q_i, \rightarrow, Q_F)$.

These languages satisfy

$$X_i = \bigcup_{a \in \Sigma} \bigcup_{q_i \xrightarrow{a} q_j} a.X_j \cup \begin{cases} \varepsilon & \text{if } q_i \in Q_F \\ \emptyset & \text{otherwise} \end{cases}$$

By
↳ mutual insertion of these equations
↳ applications of Arden's lemma
construct expressions that prove the languages X_i regular.
By
$L = X_0$, also L is regular. □

<u>Example:</u>



Corresponding system of equations:

$$X_0 = a.X_1 \cup b.X_2 \quad (1)$$
$$X_1 = a.X_0 \quad (2)$$
$$X_2 = b.X_0 \cup a.X_3 \cup \varepsilon \quad (3)$$
$$X_3 = a.X_2 \quad (4)$$

Insert (4) into (3) and (2) into (1):

$$X_0 = a.a.X_0 \cup b.X_2$$

$$X_2 = b.X_0 \cup a.a.X_2 \cup \varepsilon$$

By Arden's lemma:

$$X_2 = (a.a)^*.(b.X_0 \cup \varepsilon)$$

Insert into X_0:

$$\begin{aligned} X_0 &= a.a\,X_0 \cup b.(aa)^*.(b.X_0 \cup \varepsilon) \\ &= (a.a \cup b\,(aa)^*.b).X_0 \cup b(aa)^* \\ &= (a.a \cup b.(aa)^*.b)^*.\,b\,(aa)^* \end{aligned}$$

1.1.3 Deterministic finite automata.

Definition (DFA):

An NFA $A = (Q, q_0, \rightarrow, Q_F)$ is called deterministic,
if for all $q \in Q$ and all $a \in \Sigma$
there is precisely one $q' \in Q$ with $q \xrightarrow{a} q'$.

↳ Sometimes convenient in applications.

↳ Question:

For every non-deterministic automaton A,
(equivalently, for every regular language)
is there a deterministic automaton A' with $L(A) = L(A')$?

Yes, powerset construction.

Theorem (Rabin & Scott '59)

For every NFA A with n states,
there is a DFA A' with $L(A) = L(A')$
and A' has at most 2^n states.

Construction:

Let $A = (Q, q_0, \rightarrow, Q_F)$.
Define
$$A' := (\mathbb{P}(Q), \{q_0\}, \rightarrow', Q_F')$$
with
$$Q_1 \xrightarrow{a}' Q_2 \quad \text{where} \quad Q_2 = \{q_2 \mid q_1 \xrightarrow{a} q_2 \text{ with } q_1 \in Q_1\}$$
and
$$Q_F' := \{Q' \subseteq Q \mid Q' \cap Q_F \neq \emptyset\}$$
// sets that contain a final state

Note that A' deterministic

- For every action a, there is a goal state (may be the empty set, $\emptyset \in \mathbb{P}(Q)$).
- Goal state uniquely defined.

Consequence of this construction:

Closure of regular languages under complementation.

Note:

For an NFA A, not easy to find $\overline{L(A)}$

$L(A)$ = Words w so that there is run of A on w that is accepting

For complement:

$\overline{L(A)}$ = Words w so that all runs of A on w do not accept

In general, this $\forall$-quantification cannot be translated into $\exists$-quantification.
↳ For DFAs this works:

$\overline{L(A)}$ = Words w so that all runs of A on w do not accept
= Words w so that there is a run of A on w that does not accept.

Theorem:
Consider DFA A. Then there is DFA $\overline{A}$ with $L(\overline{A}) = \overline{L(A)}$.
Construction:
"swap final states"
Let $A = (Q, q_0, \rightarrow, Q_F)$.
Then
$\overline{A} := (Q, q_0, \rightarrow, Q \setminus Q_F)$.

To sum up.
Let $L = L(A)$ for an NFA with n states.
Then there are DFAs for L and $\overline{L}$ with at most 2^n states.
↳ Bound is optimal
=> There are languages L_n recognised by NFA with $n+1$ states
that are not recognised by DFA with less than 2^n states.
↳ If you only consider states reachable from q_0,
you can often do with less than 2^n.

1.2 Decidability and Complexity

Problems:

Given NFA A.

↳ Emptiness : $L(A) = \emptyset$?

↳ Universality : $L(A) = \Sigma^*$?

↳ Word problem: Given also $w \in \Sigma^*$. Is $w \in L(A)$?

Emptiness most important problem

↳ remaining problems can be reduced to it.

Theorem:

Emptiness for NFA A with n states can be solved in time $O(n^2)$.

Proof:

Breadth or depth-search for a final state.

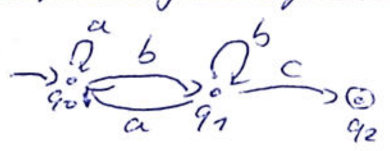
↳ Linear in number of edges, $O(n^2)$

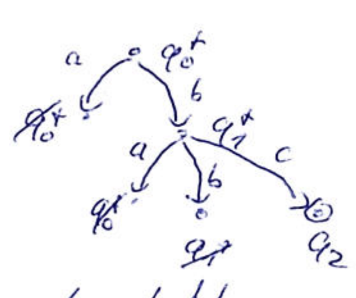
↳ If final state found, return yes.

⇒ accept word along this path

Otherwise no. □

Example for emptiness:





✓ Return "not empty"

Further problems also decidable:

↳ Intersection : $L(A_1) \cap L(A_2) = \emptyset$?

↳ Equivalence : $L(A_1) = L(A_2)$?

↳ Inclusion : $L(A_1) \subseteq L(A_2)$?