4. Minimization

Goal: • Use NFAs / DFAs as a (finite) data structure for languages (which are typically infinite objects).
• Smaller DFAs are welcome as the computation time depends on the size of the input

Approach: • Understand the requirement on the automaton imposed by the language (without looking at how the language is represented).
• Turn this requirement into an automaton.

4.1 Theorem of Myhill & Nerode

Goal: Derive an exact condition, sufficient and necessary (characterization), for regularity of $L \subseteq \Sigma^*$.

Definition:
The Nerode-right congruence $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ induced by $L \subseteq \Sigma^*$ is defined by

$u \equiv_L v$: if $\forall w \in \Sigma^*: uw \in L$ iff $vw \in L$.

Idea:
$u \equiv_L v$ iff u and v behave the same for all concatenations wrt. membership in L:
Either $uw \in L$ and $vw \in L$
or $uw \notin L$ and $vw \notin L$.

Definition of a congruence $\equiv_L \subseteq \Sigma^* \times \Sigma^*$:
• Equivalence relation: Reflexiv, transitiv, symmetric.
• Compatibel with the operations (here concatenation): $u \equiv_L v \Rightarrow ux \equiv_L vx$ for all $x \in \Sigma^*$.

Note: If $u \in L$ then $[u]_{\equiv_L} \subseteq L$.

Why? Append ε.

Example:

Consider $L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$.

We have

- $a \not\equiv_{L_1} aab$, since $a\underline{ab} \in L_1$ but $aab\underline{ab} \notin L_1$.
- $aab \equiv_{L_1} aaabbb$, both expecting one b.

In general:

$[a^k]_{\equiv_{L_1}} = \{a^k\}$. These are already infinitely many classes.

$[a^{k+1}b]_{\equiv_{L_1}} = \{a^{l+1}b^{l+1-k} \mid l \geq k\}$

// Everything that yields $a^{k+1}b$ if we remove as many as left as we remove bs right.

$[b]_{\equiv_{L_1}}$ = the rest.

Note: · L_1 has infinitely many $\equiv_{L_1}$-equivalence classes.

· The number of equivalence classes is called the index of an equivalence relation.

Theorem (Myhill & Nerode '57/'58):

$L \subseteq \Sigma^*$ is regular iff $\equiv_L$ has finite index.

Remark: L_1 is not regular, we need infinitely many classes.

Proof:

$\Rightarrow$: If $L \subseteq \Sigma^*$ is regular, we have $L = L(A)$

$\Leftarrow$:

for a DFA $A = (\Sigma, Q, q_0, \to, Q_F)$.

Consider the relation $\equiv_A \subseteq \Sigma^* \times \Sigma^*$ induced by A:

$u \equiv_A v$, if $\exists q \in Q: q_0 \xrightarrow{u} q$ and $q_0 \xrightarrow{v} q$.

Claim: • $\equiv_A$ is an equivalence on Σ^*

• $\equiv_A$ refines $\equiv_L$, meaning $\equiv_A \subseteq \equiv_L$.

Indeed, let $u \equiv_A v$.

To establish $u \equiv_L v$, we have to show that

for all $w \in \Sigma^*$: $uw \in L$ iff $vw \in L$.

To this end, consider a word $w \in \Sigma^*$:

$$uw \in L \iff \exists q \in Q \; \exists q_f \in Q_F : q_0 \xrightarrow{u} q \xrightarrow{w} q_f.$$

$$\overset{u \equiv_A v,\ A \text{ DFA}}{\iff} \exists q \in Q \; \exists q_f \in Q_F : q_0 \xrightarrow{v} q \xrightarrow{w} q_f.$$

$$\iff vw \in L.$$

Hence, $u \equiv_L v$.

This shows

$$\text{index } \equiv_L \;\leq\; \text{index } \equiv_A \;\leq\; |Q|.$$

"$\subseteq$" By construction of the <u>equivalence class automaton</u> (of L).

<u>Idea</u>: Store in the state which class has been reached.

Let $u_1, \dots, u_k \in \Sigma^*$ be representatives of the k equivalence classes of $\equiv_L$.

For every $w \in \Sigma^*$ there is precisely one u_i with $[w]_{\equiv_L} = [u_i]_{\equiv_L}$.

Why?

Because

$$\Sigma^* = [u_1]_{\equiv_L} \cup \dots \cup [u_k]_{\equiv_L}$$

with $[u_i]_{\equiv_L} \cap [u_j]_{\equiv_L} = \emptyset$ for all $i \neq j$.

Construct the DFA

$$A_L := (\Sigma, Q_L, q_{0L}, \to_L, Q_{F_L})$$

with
- $Q_L := \{[u_1]_{\equiv_L}, \dots, [u_k]_{\equiv_L}\}$
- $q_{0L} := [\varepsilon]_{\equiv_L}$
- $[u_i]_{\equiv_L} \xrightarrow{a}_L [u_i a]_{\equiv_L}$ for all $1 \leq i \leq k$ and all $a \in \Sigma$.
- $Q_{F_L} := \{[u_j]_{\equiv_L} \mid u_j \in L\}$.

Claim: $[\varepsilon]_{\equiv_L} \xrightarrow{w}_L [w]_{\equiv_L}$.

With this, we get

$$\boxed{L(A_L) = L.}$$

Proof: Let $w \in \Sigma^*$:

$$w \in L(A_L)$$

$$\Leftrightarrow \exists\, [u_j]_{\equiv_L} \in Q_{F_L} : [\varepsilon]_{\equiv_L} \xrightarrow{w} [u_j]_{\equiv_L}$$

$$\overset{(*)}{\Longleftrightarrow} \exists\, u_j \in L : [u_j]_{\equiv_L} = [w]_{\equiv_L}$$

$$\overset{(**)}{\Longleftrightarrow} w \in L.$$

(*) „⇒" By $[\varepsilon]_{\equiv_L} \xrightarrow{w}_L [w]_{\equiv_L}$ and determinism.

„⇐" By $[\varepsilon]_{\equiv_L} \xrightarrow{w}_L [w]_{\equiv_L}$ and $[u_j]_{\equiv_L} = [w]_{\equiv_L}$

(**) Since $u_i \equiv_L w$, we have $u_j \in L$ iff $w \in L$.

Example (Suffix detection):

$L_{01} = \{w01 \mid w \in \{0,1,2\}^*\}$ over $\Sigma = \{0,1,2\}$.

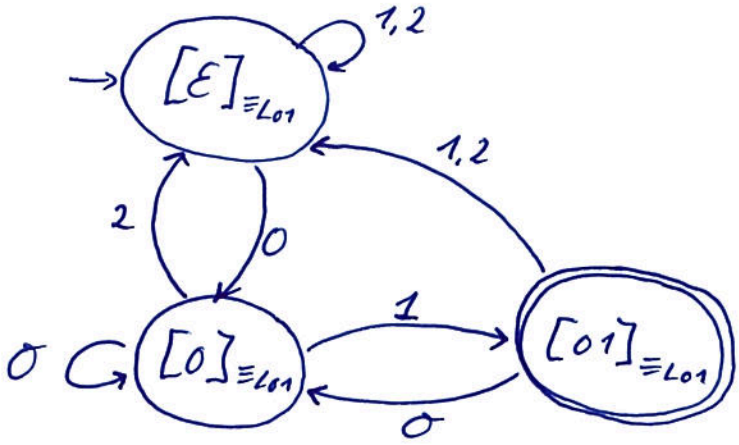
$\equiv_{L_{01}}$ has index 3:

$[0]_{\equiv_{L_{01}}}$ = all words ending in 0

$[01]_{\equiv_{L_{01}}}$ = all words ending in 01

$[\varepsilon]_{\equiv_{L_{01}}}$ = the rest.

$\mathcal{A}_{L_{01}}$:



Note: One can also construct the equivalence class automaton for L_1 but it will not have finitely many states.

4.2 Deterministic Minimal Automata

Goal: Show that
- $\mathcal{A}_L$ is a minimal deterministic automaton for L
- $\mathcal{A}_L$ is unique up to isomorphism.

Corollary (of Myhill & Nerode's Theorem):
Let $L \subseteq \Sigma^*$ be regular and with $\equiv_L$ of index $k \in \mathbb{N}$.
Every DFA accepting L has $\geq k$ states.
$\mathcal{A}_L$ reaches this minimal number.

Proof: $\text{index}_{\equiv_L} \leq |Q|$.
Construction of $\mathcal{A}_L$. □

Note: NFAs can be (a lot) more compact than DFAs
(in the sense that they have < $\text{index}_{\equiv_L}$ states).

- Consider the family of languages $(L_{\text{nondet}}(n))_{n \in \mathbb{N}}$ with

$L_{\text{nondet}}(n)$:= All words over 0,1 with 1 at the nth position from the right.

- For $L_{\text{nondet}}(n)$, an NFA can be found that has $n+1$ states.
- No DFA accepting $L_{\text{nondet}}(n)$ has less than 2^n states.

Why? Check the Nerode classes.

Intuitively:

- The DFA cannot know whether a 1 that has just been read is the appropriate one.
It thus has to remember n-digits, which yields 2^n states.
// This is an informal argument, the proof needs Myhill & Nerode.
- The NFA can guess the right 1:

Claim:
The minimal automaton is unique up to isomorphism.

Definition:

Consider $A_1 = (\Sigma, Q_1, q_{01}, \to_1, Q_{F1})$ and $A_2 = (\Sigma, Q_2, q_{02}, \to_2, Q_{F2})$.

They are called isomorphic, if there is a bijection

$$\beta : Q_1 \to Q_2$$

with $\beta(q_{01}) = q_{02}$

$\beta(Q_{F1}) = Q_{F2}$

$\forall q, q' \in Q_1 \; \forall a \in \Sigma$: $q \xrightarrow{a}_1 q'$ iff $\beta(q) \xrightarrow{a}_2 \beta(q')$.

Function β is called an isomorphism from A_1 to A_2.

(Note that isomorphism defines an equivalence on NFAs.)

Intuitively: NFAs are isomorphic if they coincide up to the names of states.

Theorem (Isomorphism theorem for DFAs):

Let $L \subseteq \Sigma^*$ be regular with index $\equiv_L$ being $k \in \mathbb{N}$.

Then every DFA A that accepts L and has k states is isomorphic to A_L.

Proof: We construct an isomorphism from A_L to A.

Let $A_L = (\Sigma, Q_L, q_{0L}, \to_L, Q_{FL})$ with $Q_L = \{[u_1]_{\equiv_L}, \dots, [u_k]_{\equiv_L}\}$.

Let $A = (\Sigma, Q, q_0, \to, Q_F)$ with $|Q| = k$.

Define function $\beta : Q_L \to Q$ by

$\beta([u_i]_{\equiv_L}) =$ the $q \in Q$ satisfying $q_0 \xrightarrow{u_i} q$.

(This is indeed a function since A is deterministic.)

Claim: β is an isomorphism from A_L to A.

Checking this is homework. □

4.3 Constructing the Deterministic Minimal Automaton

Goal: Given an arbitrary DFA, construct out of it the (up to isomorphism) unique minimal DFA.

Idea: Apply two steps:
- ↳ Eliminate unreachable states
- ↳ Collapse equivalent states.

Definition:
Let $A = (\Sigma, Q, q_0, \to, Q_F)$.
A state $q \in Q$ is reachable, if there is $w \in \Sigma^*$ with $q_0 \xrightarrow{w} q$

Note:
- If $q \in Q$ is reachable, it is reachable by w with $|w| \leq |Q|$.
- See fixed point for emptiness.

From now on:
Assume all unreachable states have been removed.

Definition:
Let $A = (\Sigma, Q, q_0, \to, Q_F)$.
Two states $q_1, q_2 \in Q$ are equivalent, $q_1 \sim q_2$,
if for all $w \in \Sigma^*$:

$$q_1 \xrightarrow{w} q_1' \in Q_F \quad \text{iff} \quad q_2 \xrightarrow{w} q_2' \in Q_F.$$

There is a close correspondence between $\sim$ and $\equiv_L$:

Lemma: Let $L = L(A)$ with $A = (\Sigma, Q, q_0, \to, Q_F)$.
Consider $q_0 \xrightarrow{u} q_1$ and $q_0 \xrightarrow{v} q_2$.
We have

$$u \equiv_L v \quad \text{iff} \quad q_1 \sim q_2.$$

Consequence of the lemma (and our understanding of the minimal automaton):

- If two states are reachable and equivalent, they should be collapsed to obtain the minimal DFA.

Phrased differently, equivalence $\sim$ identifies those states that have to collapse in the minimal automaton.

Why? These states represent the same $\equiv_L$-class.

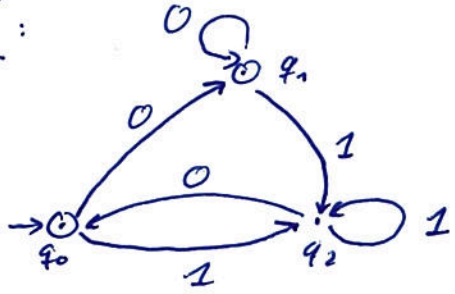
- Hence, we will construct from A an automaton where the states are the $\sim$-equivalence classes of the reachable states.

The transition relation is defined as expected:

$$[q]_\sim \xrightarrow{a} [q']_\sim, \text{ if } q \xrightarrow{a} q'.$$

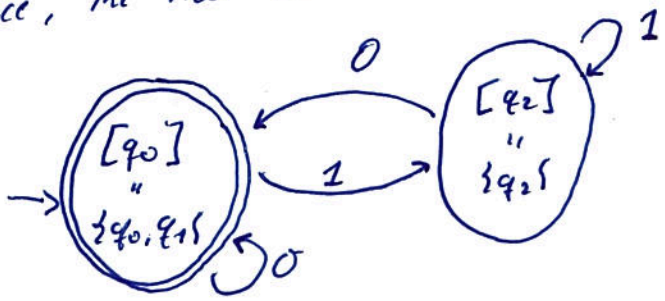
Why is this well-defined?

Example:



We have $q_0 \sim q_1$, but $q_2 \not\sim q_0$.

Hence, the new automaton is



Todo: Check whether two states are $\sim$-equivalent.

Idea: Do a fixed point that determines the inequivalent states.

Works in $O(|Q|^2)$.

Table filling algorithm:

↳ Maintain a table of pairs of states $\{q, q'\}$ with $q \neq q'$.

↳ Mark all states that have been identified as being different.

↳ Collapse all equivalent states.

If we have states $Q = \{q_0, \dots, q_{n-1}\}$,
we can do with a triangular table
of size

$$\frac{1}{2} n \cdot (n-1).$$

Example:

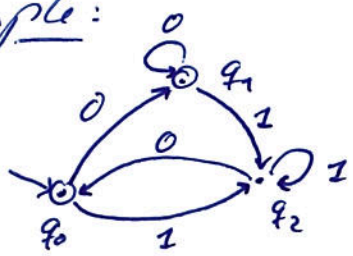
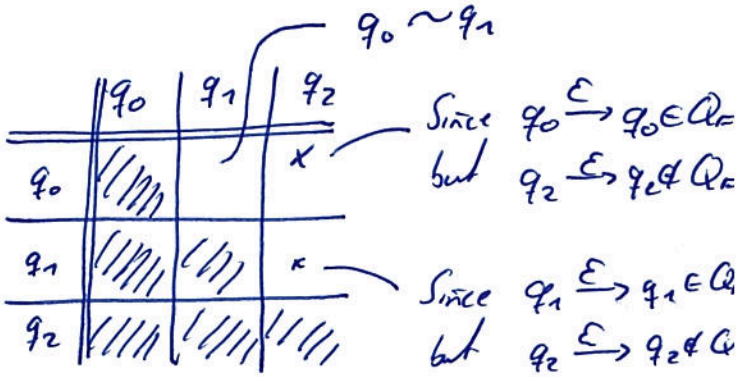


Table:



Pseudocode:

Initially, the table is unmarked;
Mark all $\{q, q'\}$ with $q \in Q_F$ and $q' \notin Q_F$ or vice versa;
while $\exists$ unmarked pair $\{q_1, q_1'\}$ and a letter $a \in \Sigma$
with $q_1 \xrightarrow{a} q_2$ and $q_1' \xrightarrow{a} q_2'$
where $\{q_2, q_2'\}$ is marked do
Mark $\{q_1, q_1'\}$;
od
collapse maximal sets of unmarked states;

Note:

- All this does not work for NFAs.
- Minimizing NFAs is an active research topic.
- One option is bisimulation, see program analysis lecture.