1 Designing DFAs

1.1 General Method

Typical Problem

Problem
Given a language L, design a DFA M that accepts L, i.e., $\mathbf{L}(M) = L$.

Methodology

- Imagine yourself in the place of the machine, reading symbols of the input, and trying to determine if it should be accepted.
- Remember at any point you have only seen a part of the input, and you don't know when it ends.
- *Figure out what to keep in memory.* It cannot be all the symbols seen so far: it must fit into a finite number of bits.

1.2 Examples

Strings containing 0

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that contain at least one 0.

Solution
What do you need to remember? Whether you have seen a 0 so far or not!

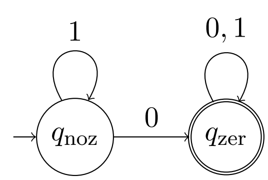

Figure 1: Automaton accepting strings with at least one 0.

Even length strings

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that have an even length.

Solution
What do you need to remember? Whether you have seen an odd or an even number of symbols.

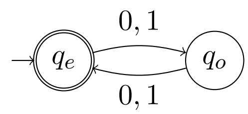

Figure 2: Automaton accepting strings of even length.

Pattern Recognition

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that have 001 as a substring, where u is a substring of w if there are w_1 and w_2 such that $w = w_1 u w_2$.

Solution
What do you need to remember? Whether you

- haven't seen any symbols of the pattern
- have just seen 0
- have just seen 00
- have seen the entire pattern 001

Pattern Recognition Automaton

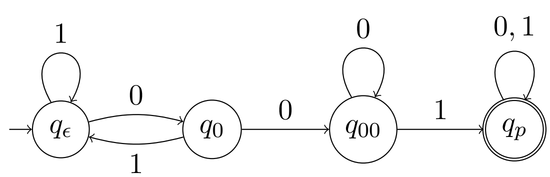

Figure 3: Automaton accepting strings having 001 as substring.

`grep` **Problem**

Problem
Given text T and string s, does s appear in T?

Naïve Solution

$$\overbrace{T_1\, T_2\, T_3\, \ldots T_n}^{=s?}\; T_{n+1}\; \ldots T_t$$

Running time $= O(nt)$, where $|T| = t$ and $|s| = n$.

`grep` Problem

Smarter Solution

Solution

- Build DFA M for $L = \{w \mid \text{there are } u, v \text{ } s.t. \text{ } w = usv\}$
- Run M on text T

Time = time to build M + $O(t)$!

Questions

- Is L regular no matter what s is?
- If yes, can M be built "efficiently"?

Knuth-Morris-Pratt (1977): Yes to both the above questions.

Multiples

Problem

Design an automaton that accepts all strings w over $\{0, 1\}$ such that w is the binary representation of a number that is a multiple of 5.

Solution

What must be remembered? The remainder when divided by 5.

How do you compute remainders?

- If w is the number n then $w0$ is $2n$ and $w1$ is $2n + 1$.
- $(a.b + c) \bmod 5 = (a.(b \bmod 5) + c) \bmod 5$
- *e.g.* $1011 = 11$ (decimal) $\equiv 1 \bmod 5$ $10110 = 22$ (decimal) $\equiv 2 \bmod 5$ $10111 = 23$ (decimal) $\equiv 3 \bmod 5$

Automaton for recognizing Multiples

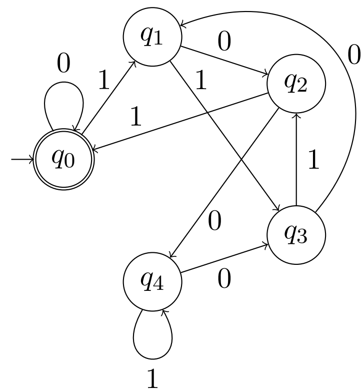

Figure 4: Automaton recognizing binary numbers that are multiples of 5.

A One k-positions from end

Problem
Design an automaton for the language $L_k = \{w \mid k\text{th character from end of } w \text{ is } 1\}$

Solution
What do you need to remember? The last k characters seen so far!

Formally, $M_k = (Q, \{0,1\}, \delta, q_0, F)$

- States $= Q = \{\langle w \rangle \mid w \in \{0,1\}^* \text{ and } |w| \leq k\}$
- $\delta(\langle w \rangle, b) = \begin{cases} \langle wb \rangle & \text{if } |w| < k \\ \langle w_2 w_3 \dots w_k b \rangle & \text{if } w = w_1 w_2 \dots w_k \end{cases}$
- $q_0 = \langle \epsilon \rangle$
- $F = \{\langle 1 w_2 w_3 \dots w_k \rangle \mid w_i \in \{0,1\}\}$

1.3 Lower Bounds

Lower Bound on DFA size

Proposition 1. *Any DFA recognizing L_k has at least 2^k states.*

Proof. Let M, with initial state q_0, recognize L_k and assume (for contradiction) that M has $< 2^k$ states.

- Number of strings of length $k = 2^k$
- There must be two distinct string w_0 and w_1 of length k such that for some state q, $q_0 \xrightarrow{w_0}_M q$ and $q_0 \xrightarrow{w_1}_M q$.

Let i be the first position where w_0 and w_1 differ. Without loss of generality assume that w_0 has 0 in the *ith* position and w_1 has 1.

$$\begin{aligned} w_0 0^{i-1} &= \ \ldots\ \overbrace{0\ \ldots\ 0^{i-1}}^{k} \\ w_1 0^{i-1} &= \underbrace{\ldots}_{i-1}\ 1 \underbrace{\ldots}_{k-i} 0^{i-1} \end{aligned}$$

$w_0 0^{i-1} \notin L_k$ and $w_1 0^{i-1} \in L_k$. Thus, M cannot accept both $w_0 0^{i-1}$ and $w_1 0^{i-1}$.

So far, $w_0 0^{i-1} \notin L_n$, $w_1 0^{i-1} \in L_n$, $q_0 \xrightarrow{w_0}_M q$, and $q_0 \xrightarrow{w_1}_M q$.

$$\begin{aligned} q_0 \xrightarrow{w_0 0^{i-1}}_M q_1 \quad &\text{iff} \quad q \xrightarrow{0^{i-1}}_M q_1 \\ &\text{iff} \quad q_0 \xrightarrow{w_1 0^{i-1}}_M q_1 \end{aligned}$$

Thus, M accepts or rejects both $w_0 0^{i-1}$ and $w_1 0^{i-1}$. Contradiction! □

---

2 Inductive Proofs for DFAs

2.1 Induction Proofs

Induction Principle

- Infinite sequence of statements $S_0, S_1, \ldots$
- *Goal:* Prove $\forall i.\ S_i$ is true
- Prove S_0 is true *[Base Case]*
- For an arbitrary i, assuming S_j is true for all $j < i$ *[Induction Hypothesis]*, establishes S_i to be true *[Induction Step]*.
- Conclude $\forall i.\ S_i$ is true.

Why does induction work?

- Assume S_0 is true (Base case holds), and for any i, assuming S_j is true for all $j < i$, we can conclude S_i is true (Induction step holds).

- Suppose (for contradiction) S_i does not hold for some i.
- Let k be the smallest i such that S_i does not hold. Existence of such a smallest k is a consequence of a property of natural numbers that any non-empty set of natural numbers has a smallest element in it *(Well-ordering principle)*.
- That means for all $j < k$, S_j holds.
- Then by the induction step, S_k holds! Contradiction, establishing that S_i holds for all i.

2.2 Properties about DFAs

Deterministic Behavior

Proposition 2. *For a DFA $M = (Q, \Sigma, \delta, q_0, F)$, and any $q \in Q$, and $w \in \Sigma^*$, $|\hat{\delta}_M(q, w)| = 1$.*

Proof. Proof is by induction on $|w|$. Thus, S_i is taken to be

For every $q \in Q$, and $w \in \Sigma^i$, $|\hat{\delta}_M(q, w)| = 1$.

Base Case: We need to prove the case when $w \in \Sigma^0$. Thus, $w = \epsilon$. By definition $\xrightarrow{w}_M$, $q \xrightarrow{w}_M q'$ if and only $q' = q$. Thus, $|\hat{\delta}_M(q, w)| = |\{q\}| = 1$.

Ind. Hyp.: Suppose for every $q \in Q$, and $w \in \Sigma^*$ such that $|w| < i$, $|\hat{\delta}_M(q, w)| = 1$.

Ind. Step: Consider (without loss of generality) $w = a_1 a_2 \cdots a_i$, such that $a_i \in \Sigma$. Take $u = a_1 \cdots a_{i-1}$

$$\begin{array}{lll} q \xrightarrow{w}_M q' & \text{iff there are } r_0, r_1, \ldots, r_i \text{ such that } r_0 = q,\ r_i = q',\ \text{and } \delta(r_j, a_{j+1}) = r_{j+1} \\ & \text{iff there is } r_{i-1} \text{ such that } q \xrightarrow{u}_M r_{i-1} \text{ and } \delta(r_{i-1}, a_i) = q' \end{array}$$

Now, by induction hypothesis, since $|\hat{\delta}_M(q, u)| = 1$, there is a unique r_{i-1} such that $q \xrightarrow{u}_M r_{i-1}$. Also, since from any state r_{i-1} on symbol a_i the next state is uniquely determined, $|\hat{\delta}_M(q, w)| = 1$.

□

DFA Computation

Proposition 3. *Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA. For any $q_1, q_2 \in Q$, $u, v \in Sigma^*$, $q_1 \xrightarrow{uv}_M q_2$ iff there is $q \in Q$ such that $q_1 \xrightarrow{u}_M q$ and $q \xrightarrow{v}_M q_2$.*

Proof. Let $u = a_1 a_2 \ldots a_i$ and $v = a_{i+1} \cdots a_{i+k}$. Observe that,

$$\begin{array}{ll} q_1 \xrightarrow{uv}_M q_2 & \text{iff there are } r_0, r_1, \ldots, r_{i+k} \text{ such that } r_0 = q_1,\ r_{i+k} = q_2,\ \text{and } \delta(r_j, a_{j+1}) = r_{j+1} \\ & \text{iff there is } r_i\ (= q \text{ of the proposition}) \text{ such that } q_1 \xrightarrow{u}_M r_i \text{ and } r_i \xrightarrow{v}_M q_2 \end{array}$$

□

Conventions in Inductive Proofs

"We will prove by induction on $|v|$" is a short-hand for "We will prove the proposition by induction. Take S_i to be statement of the proposition restricted to strings v where $|v| = i$."

2.3 Proving Correctness of DFA Constructions

Proving Correctness of DFAs

Problem
Show that DFA M recognizes language L.

That is, we need to show that for all w, $w \in \mathbf{L}(M)$ iff $w \in L$. This is often carried out by induction on $|w|$.

Example I

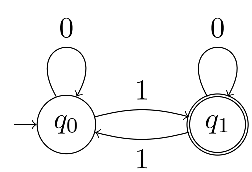

Figure 5: Transition Diagram of M_1

Proposition 4. $\mathbf{L}(M_1) = \{w \in \{0,1\}^* \mid w$ *has an odd number of 1s*$\}$

Proof. We will prove this by induction on $|w|$. That is, let S_i be

$$\text{For all } w \in \{0,1\}^i.\ M_1 \text{ accepts } w \text{ iff } w \text{ has an odd number of 1s}$$

Observe that M_1 accepts w iff $q_0 \xrightarrow{w}_{M_1} q_1$. So we could rewrite S_i as

$$\text{For all } w \in \{0,1\}^i.\ q_0 \xrightarrow{w}_{M_1} q_1 \text{ iff } w \text{ has an odd number of 1s}$$

Base Case: When $w = \epsilon$, w has an even number of 1s. Further, $q_0 \xrightarrow{\epsilon}_{M_1} q_0$, and so M_1 does not accept w.

Ind. Hyp.: Assume that for all w of length $< n$, $q_0 \xrightarrow{w}_{M_1} q_1$ iff w has an odd number of 1s.

Ind. Step: Consider w of length n; without loss of generality, w is either $0u$ or $1u$ for some string u of length $i - 1$.

If $w = 0u$ then, w has an odd number of 1s iff u has an odd number of 1s, iff (by ind. hyp.) $q_0 \xrightarrow{u}_{M_1} q_1$ iff $q_0 \xrightarrow{w=0u}_{M_1} q_1$ (since $\delta(q_0, 0) = q_0$).

On the other hand, if $w = 1u$ then, w has an odd number of 1s iff u has an even number of 1s. Now $q_0 \xrightarrow{w=1u}_{M_1} q_1$ iff $q_1 \xrightarrow{u}_{M_1} q_1$. Does M_1 accept u that has an even number of 0s from state q_1? Unfortunately, we cannot use the induction hypothesis in this case, as the hypothesis does not say anything about what strings u are accepted when the automaton is started from state q_1; it only gives the behavior on strings when M_1 is started in the initial state q_0. We need to strengthen the hypothesis to make the proof work!! The strengthening will explicitly tell us the behavior of the machine on strings when starting from states other than the initial state.

New (correct) induction proof: Let S_i be

$$\begin{aligned} \forall w \in \{0,1\}^i. \quad & q_0 \xrightarrow{w}_{M_1} q_1 \text{ iff } w \text{ has an odd number of 1s} \\ & \text{and } q_1 \xrightarrow{w}_{M_1} q_1 \text{ iff } w \text{ has an even number of 1s} \end{aligned}$$

We will prove this sequence of statements by induction.

Base Case: When $w = \epsilon$, w has an even number of 1s. Further, $q_0 \xrightarrow{\epsilon}_{M_1} q_0$ and $q_1 \xrightarrow{w}_{M_1} q_1$, and so M_1 does not accept w from state q_0, but accepts w from state q_1. This establishes the base case.

Ind. Hyp.: Assume that for all w of length $< n$, $q_0 \xrightarrow{w}_{M_1} q_1$ iff w has an odd number of 1s and $q_1 \xrightarrow{w}_{M_1} q_1$ iff w has an even number of 1s.

Ind. Step: Consider w of length n; without loss of generality, w is either $0u$ or $1u$ for some string u of length $i - 1$.

If $w = 0u$ then, w has an odd number of 1s iff u has an odd number of 1s, iff (by ind. hyp.) $q_0 \xrightarrow{u}_{M_1} q_1$ iff $q_0 \xrightarrow{w=0u}_{M_1} q_1$ (since $\delta(q_0, 0) = q_0$). And w has an even number of 1s iff u has an even number of 1s iff (by ind. hyp.) $q_1 \xrightarrow{u}_{M_1} q_1$ iff $q_1 \xrightarrow{w=0u}_{M_1} q_1$ (since $\delta(q_1, 0) = q_1$).

On the other hand, if $w = 1u$ then $q_0 \xrightarrow{w=1u}_{M_1} q_1$ iff $q_1 \xrightarrow{u}_{M_1} q_1$ (since $\delta(q_0, 1) = q_1$) iff (by ind. hyp.) u has an even number of 1s iff $w = 1u$ has an odd number of 1s. Similarly, $q_1 \xrightarrow{w=1u}_{M_1} q_1$ iff $q_0 \xrightarrow{u}_{M_1} q_1$ (since $\delta(q_1, 1) = q_0$) iff (by ind. hyp.) u has an odd number of 1s iff w has an even number of 1s.

$\square$

Remark

The above induction proof can be made to work *without* strengthening if in the first induction proof step, we considered $w = ua$, for $a \in \{0, 1\}$, instead of $w = au$ as we did. However, the fact that the induction proof works without strengthening here is a very special case, and does not hold in general for DFAs.

Example II

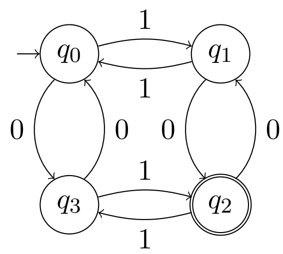

Figure 6: Transition Diagram of M_2

Proposition 5. $\mathbf{L}(M_2) = \{w \in \{0,1\}^* \mid w \textit{ has an odd number of 1s and odd number of 0s}\}$

Proof. We will once again prove the proposition by induction on $|w|$. The straightforward proof would suggest that we take S_i to be

For any $w \in \{0,1\}^i$. M_2 accepts w iff w has an odd number of 1s and 0s

Since M_2 accepts w iff $q_0 \xrightarrow{w}_{M_2} q_2$, we could rewrite the condition as "$q_0 \xrightarrow{w}_{M_2} q_2$ iff w has an odd number of 1s and 0s". The induction proof will unfortunately not go through! To see this, consider the induction step, when $w = 0u$. Now, $q_0 \xrightarrow{w}_{M_2} q$ iff $q_3 \xrightarrow{u}_{M_2} q$, because M_2 goes to state q_3 (from q_0) on reading 0. Since w and u have the same parity for the number of 1s, but opposite parity for the number of 0s, w must be accepted (i.e., reach state q_2) iff u is accepted from q_3 when u has an odd number of 1s and even number of 0s. But is that the case? The induction hypothesis says nothing about strings accepted from state q_3, and so the induction step cannot be established.

This is typical of many induction proofs. Again, we must *strengthen* the proposition in order to construct a proof. The proposition must not only characterize the strings that are accepted from the initial state q_0, but also those that are accepted from states q_1, q_2, and q_3.

We will show by induction on w that

(a) $q_0 \xrightarrow{w}_{M_2} q_2$ iff w has an odd number of 0s and odd number of 1s,

(b) $q_1 \xrightarrow{w}_{M_2} q_2$ iff w has odd number of 0s and even number of 1s,

(c) $q_2 \xrightarrow{w}_{M_2} q_2$ iff w has an even number of 0s and even number of 1s, and

(d) $q_3 \xrightarrow{w}_{M_2} q_2$ iff w has even number of 0s and odd number of 1s.

Thus in the our new induction proof, statement S_i says that conditions (a),(b),(c), and (d) hold for all strings of length i.

Base Case: When $|w| = 0$, $w = \epsilon$. Observe that w has an even number of 0s and 1s, and $q \xrightarrow{\epsilon}_{M_2} q$ for any state q. String ϵ is only accepted from state q_2, and thus statements (a),(b),(c), and (d) hold in the base case.

Ind. Hyp.: Suppose (a),(b),(c),(d) all hold for any string w of length $< n$.

Ind. Step: Consider w of length n. Without loss of generality, w is of the form au, where $a \in \{0,1\}$ and $u \in \{0,1\}^{n-1}$.

- *Case* $q = q_0,\ a = 0$: $q_0 \xrightarrow{0u}_{M_2} q_2$ iff $q_3 \xrightarrow{u}_{M_2} q_2$ iff u has even number of 0s and odd number of 1s (by ind. hyp. (d)) iff w has odd number of 0s and odd number of 1s.
- *Case* $q = q_0,\ a = 1$: $q_0 \xrightarrow{1u}_{M_2} q_2$ iff $q_1 \xrightarrow{u}_{M_2} q_2$ iff u has odd number of 0s and even number of 1s (by ind. hyp. (b)) iff w has odd number of 0s and odd number of 1s
- *Case* $q = q_1,\ a = 0$: $q_1 \xrightarrow{0u}_{M_2} q_2$ iff $q_2 \xrightarrow{u}_{M_2} q_2$ iff u has even number of 0s and even number of 1s (by ind. hyp. (c)) iff w has odd number of 0s and even number of 1s
- ...And so on for the other cases of $q = q_1$ and $a = 1$, $q = q_2$ and $a = 0$, $q = q_2$ and $a = 1$, $q = q_3$ and $a = 0$, and finally $q = q_3$ and $a = 1$. $\square$

Proving Correctness of a DFA

Proof Template
Given a DFA M having n states $\{q_0, q_1, \ldots q_{n-1}\}$ with initial state q_0, and final states F, to prove that $L(M) = L$, we do the following.

1. Come up with languages $L_0, L_1, \ldots L_{n-1}$ such that $L_0 = L$
2. Prove by induction on $|w|$, $\hat{\delta}_M(q_i, w) \cap F \neq \emptyset$ if and only if $w \in L_i$
