Lectures 2 & 3: Formal Languages and Complexity
October 5, 1998
Bud Mishra

1 Finite State Automata

An DFA (Deterministic Finite Automaton) is made of several components

- A set of states
- Rules of transition
  (Rules for going from one state to another)
- Input alphabet
- A start state + a set of accept states

Definition. 1.1 A finite automaton is a 5-tuple (quintuple)

$$(Q, \Sigma, \delta, q_0, F),$$

where

1. Q = Finite set of states
2. Σ = Finite alphabet of input symbols
3. Transition function

$$\begin{aligned} \delta &: Q \times \Sigma \to Q \\ &: (q_1, a) \mapsto q_2. \end{aligned}$$

   Deterministic

4. $q_0 \in Q, =$ Distinguished start state
5. $F \subseteq Q, =$ Set of accept (final) states

1.1 Computation

Let $M = (Q, \Sigma, \delta, q_0, F) =$ DFA. Let $\sigma = a_1 a_2 \ldots a_n =$ a string over the alphabet Σ.

M accepts σ if $\exists$ a sequence of states $r_0, r_1, \ldots, r_n \in Q$ such that

1. $r_0 = q_0$ (M starts in the initial state.)
2. $r_n \in F$ (M finishes in an accept state after reading σ.)
3. $\forall\ i \geq 0\ \delta(r_i, a_{i+1}) = r_{i+1}$. ($M$ moves from state to state according to the transition function δ.)

$\mathcal{L}(M) =$ The language recognized by the machine M; Language of machine M. The set of all strings that machine M accepts.

$$\mathcal{L}(M) = \{\sigma \in \Sigma^* | q_0 \rightarrow_\delta^\sigma q_f, q_f \in F\}$$

2 Regular Languages

Languages accepted by finite automata, DFA's.

Definition. 2.1 Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two languages.

Define the regular operations *union*, *concatenation* and *Kleene's star* operations as follows:

1. Union:
$$\mathcal{L}_1 \cup \mathcal{L}_2 = \{\sigma | \sigma \in \mathcal{L}_1 \vee \sigma \in \mathcal{L}_2\}$$
2. Concatenation:
$$\mathcal{L}_1 \circ \mathcal{L}_2 = \{\sigma\mu | \sigma \in \mathcal{L}_1 \wedge \mu \in \mathcal{L}_2\}$$
3. Kleene's Star
$$\mathcal{L}_1^* = \{\sigma_1 \sigma_2 \cdots \sigma_k | k \geq 0 \wedge \forall i \sigma_i \in \mathcal{L}_1\}$$

2.1 Closure Properties

Theorem. 2.2 *The class of regular languages is closed under the regular operations:* union, concatenation *and* Kleene's star *operations.*

1. $\forall \mathcal{L}_1, \mathcal{L}_2 = Reg\ \ \mathcal{L}_1 \cup \mathcal{L}_2 = Reg.$
2. $\forall \mathcal{L}_1, \mathcal{L}_2 = Reg\ \ \mathcal{L}_1 \circ \mathcal{L}_2 = Reg.$
3. $\forall \mathcal{L}_1 = Reg\ \ \mathcal{L}_1^* = Reg.$

PROOF.

Deferred, see Sipser.

3 Nondeterminism

Definition. 3.1 A nondeterministic finite automaton (NFA) is a 5-tuple (quintuple)

$$(Q, \Sigma, \delta, q_0, F),$$

where

1. $Q =$ Finite set of states
2. $\Sigma =$ Finite alphabet of input symbols
3. Transition function

$$\begin{aligned} \delta\ &:\ Q \times \Sigma_\epsilon \to \mathcal{P}(Q) \\ &:\ (q_1, \epsilon) \mapsto Q' \subseteq Q \\ &:\ (q_1, a) \mapsto Q'' \subseteq Q \end{aligned}$$

   Deterministic

4. $q_0 \in Q, =$ Distinguished start state
5. $F \subseteq Q, =$ Set of accept (final) states

3.1 Nondeterministic Computation

Let $N = (Q, \Sigma, \delta, q_0, F)$ = NFA. Let $\sigma = y_1 y_2 \ldots y_m$ = a string over the extended alphabet $\Sigma \cup \{\epsilon\} =]Sigma_\epsilon$.

N accepts σ if $\exists$ a sequence of states $r_0, r_1, \ldots, r_m \in Q$ such that

1. $r_0 = q_0$ (N starts in the initial state.)
2. $r_m \in F$ (N finishes in an accept state after reading σ.)
3. $\forall\ i \geq 0\ r_{i+1} \in \delta(r_i, a_{i+1})$. ($N$ *chooses* to move from a state to another state according to the transition function δ while staying on the sequence of states r_0, ...)

$\mathcal{L}(N)$ = The language recognized by the machine N; Language of machine N. The set of all strings that machine N accepts.

$$\mathcal{L}(N) = \{\sigma \in \Sigma^* | q_0 \rightarrow^\sigma_\delta Q_f, Q_f \cap F \neq \emptyset\}$$

4 Equivalence

Definition. 4.1 Two machines M_1 and M_2 are said to be *equivalent* if they recognize the same language:

$$\mathcal{L}(M_1) = \mathcal{L}(M_2).$$

Theorem. 4.2 *Every nondeterministic finite automaton has an equivalent deterministic finite automaton.*

PROOF.

Omitted. By standard subset construction, see Sipser.

Corollary. 4.3 *A language is regular iff some nondeterministic finite automaton accepts it.*

5 Closure under Regular Operations

See Sipser.

6 Homomorphisms

Map from $\Sigma_1^* \to \Sigma_2^*$ defined as follows:

$$\begin{aligned} h &: \Sigma_1 \to \Sigma_2^* \\ &: a_i \mapsto \sigma_i \end{aligned}$$

Extend h as follows:

$$\begin{aligned} h &: \Sigma_1^* \to \Sigma_2^* \\ &: \epsilon \mapsto \epsilon \\ &: a_{i_1} a_{i_2} \cdots a_{i_n} \mapsto \sigma_{i_1} \sigma_{i_2} \cdots \sigma_{i_n} \end{aligned}$$

Note that

$$\forall \mu_1, \mu_2 \in \Sigma_1^* \;\; h(\mu_1 \mu_2) = h(\mu_1) h(\mu_2).$$

Homomorphic image of $\mathcal{L}$

$$h(\mathcal{L}) = \{h(\mu) | \mu \in \mathcal{L}\}.$$

Inverse homomorphic image of $\mathcal{L}$

$$h^{-1}(\mathcal{L}) = \{\mu | h(\mu) \in \mathcal{L}\}.$$

Note

$$h^{-1}(\sigma) = \{\mu | h(\mu) = \sigma\}.$$

Theorem. 6.1 *The class of regular languages is closed under homomorphisms and inverse homomorphisms.*

Let

$$h : \Sigma_1^* \to \Sigma_2^*,$$

be a homomorphism.

If $\mathcal{L}_1 \subseteq \Sigma_1^*$ *is regular then so is* $h(\mathcal{L}_1)$*.*

If $\mathcal{L}_2 \subseteq \Sigma_2^*$ *is regular then so is* $h^{-1}(\mathcal{L}_2)$*.*

PROOF.

HOMOMORPHISM

Let $M(\mathcal{L}_1) = (Q, \Sigma_1, \delta, q_0, F)$ be a DFA accepting $\mathcal{L}_1$. Construct a new NFA N as follows:

First define k as

$$k = \max\{|h(a_i)| : a_i \in \Sigma_1\}.$$

Let Q' be defined as

$$Q \cup Q \times \Sigma_2 \cup \cdots \cup Q \times \Sigma_2^k.$$

The NFA is defined as follows:

$$N = (Q', \Sigma_2, \delta', q_0, F),$$

where

$$\begin{aligned} \delta' &: Q' \times \Sigma_{2_\epsilon} \to \mathcal{P}(Q') \\ &: ((q_1, b_1, \ldots, b_j), b_{j+1}) \mapsto (q_1, b_1, \ldots, b_j, b_{j+1}) \\ &: ((q_1, b_1, \ldots, b_l), \epsilon) \mapsto \{\delta(q_1, a_i) | h(a_i) = b_1 \cdots b_l\} \end{aligned}$$

Thus the NFA remembers a *long enough* string to know which homomorphic translation of an input symbol is to be used.

INVERSE HOMOMORPHISM

Let $M(\mathcal{L}_2) = (Q, \Sigma_2, \delta, q_0, F)$ be a DFA accepting $\mathcal{L}_2$. Construct a new DFA M' as follows: M' after reading a_i does what m would do with $h(a_i)$.

The DFA is defined as follows:

$$M' = (Q, \Sigma_1, \delta', q_0, F),$$

where

$$\begin{aligned} \delta' &: Q \times \Sigma_1 \to Q \\ &: (q_1, a) \mapsto \delta(q_1, h(a)). \end{aligned}$$

Thus if $h(a)$ is $b_1 b_2 \cdots b_l$ then

$$\delta(q_1, h(a)) = \delta(\delta(\cdots, \delta(\delta(q_1, b_1), b_2), \cdots), b_l).$$

if $h(a)$ is ϵ then

$$\delta(q_1, h(a)) = q_1.$$

The rest is straightforward.

7 Quotient Operation

Let $\mathcal{L}_1$, $\mathcal{L}_2 \subseteq \Sigma^*, =$ Regular languages.

Quotient of $\mathcal{L}_1$ with respect to $\mathcal{L}_2$ is defined as

$$\mathcal{L}_1/\mathcal{L}_2 = \{\sigma \in \Sigma^* | \exists \mu \in \mathcal{L}_2 \; \sigma\mu \in \mathcal{L}_1\}$$

Theorem. 7.1 *The class of regular languages is closed under the quotient operation.*

PROOF.

Let $\mathcal{L}_1$, $\mathcal{L}_2 \subseteq \Sigma^*, =$ Regular languages.

Let $\Sigma = \{a_1, a_2, \ldots, a_n\}$.

Define alphabets:

$$\begin{aligned} \Delta &= \{b_1, b_2, \ldots, b_n\}. \\ \Delta' &= \{b'_1, b'_2, \ldots, b'_n\}. \\ \hat{\Delta} &= \Delta \cup \Delta' = \{b_1, b_2, \ldots, b_n, b'_1, b'_2, \ldots, b'_n\}. \end{aligned}$$

Define the following homomorphisms:

$$\begin{aligned} h_1 &: \Sigma \to \hat{\Delta}^* \\ &: a_i \mapsto b_i \\ \\ h_2 &: \hat{\Delta} \to \Sigma^* \\ &: b_i \mapsto a_i \\ &: b'_i \mapsto a'_i \\ \\ h_3 &: \hat{\Delta} \to \Sigma^* \\ &: b_i \mapsto \epsilon \\ &: b'_i \mapsto a_i \end{aligned}$$

Note that

$$h_3(\Delta' * h_1(\mathcal{L}_2) \cap h_2^{-1}(\mathcal{L}_1)) = \mathcal{L}_1/\mathcal{L}_2.$$

Since the class of regular languages is closed under homomorphisms, inverse homomorphisms, intersection and concatenation, it is also closed under the quotient operation.

A more constructive proof:

Let $M_1 = (Q, \Sigma, \delta, q_0, F) =$ DFA, accepting $\mathcal{L}_1$; $\mathcal{L}(M_1) = \mathcal{L}_1$. Let $\mathcal{L}_2 =$ arbitrary (e.g., regular) language.

Construct a new DFA as follows: $M' = (Q, \Sigma, \delta, q_0, F')$. The final states of M' are

$$F' = \{q \in Q | \exists \mu \in \mathcal{L}_2 \; \delta(q, \mu) \in F\}.$$

$$\begin{aligned} \delta(q_0, \sigma) \in F' \quad & \text{iff} \exists \mu \in \mathcal{L}_2 \; \delta(q_0, \sigma\mu) \in F \\ & \text{iff} \exists \mu \in \mathcal{L}_2 \; \sigma\mu \in \mathcal{L}_1 \\ & \text{iff} \sigma \in \mathcal{L}_1 / \mathcal{L}_2. \end{aligned}$$

The class of regular languages is closed under the quotient operation with respect to an arbitrary set.