2 Finite automata

Finite automata are the simplest computational models for computers with an extremely limited amount of memory.

2.1 Examples of finite automata

Reading: Sipser 1.1 (pp. 31-34)
Example: Controller for an automatic door (a sliding door with a front pad and a rear pad). Two states (closed and open) and four possible input conditions (front, rear, neither, and both).
Example: A farmer (F) with a cabbage (C), a dog (D), and a goat (G) wants to cross a river. There is a small boat which can only carry the farmer plus one of the three things. At any time, the cabbage and the goat cannot be left alone neither can the dog and the goat. How can the farmer cross the river? A finite automaton with start state FCDG-$\emptyset$ and accept state $\emptyset$-FCDG can solve the puzzle.
Example: Use of automata theory in software applications includes: study of the behavior of digital circuits, lexical analyzer in compilers, text pattern matching, and verification of finite-state systems.

2.2 Deterministic finite automata

Reading: Sipser 1.1 (pp. 35-44)

- DFA $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an alphabet, $q_0 \in Q$ is a start state, $F \subseteq Q$ is a set of accept/final states, and $\delta : Q \times \Sigma \to Q$ is a transition function, where $\delta(q,a) = p$ is the state that M will enter if the current state is q and the current symbol is a.
- State diagrams an transition tables are other representations of DFAs.
- Although rigorously, $\delta(q,a)$ should be defined $\forall q \in Q$ and $\forall a \in \Sigma$, all inaccessible and dead-end states with associated arcs may be removed to simplify the DFA.
- Extending δ to $\hat{\delta} : Q \times \Sigma^* \to Q$: For any $q \in Q$, define $\hat{\delta}(q,\varepsilon) = q$ and $\hat{\delta}(q,w) = \delta(\hat{\delta}(q,x),a)$ if $w = xa$ (or $\hat{\delta}(q,w) = \hat{\delta}(\delta(q,b),y)$ if $w = by$).
- Language of a DFA M (or language recognized by M) is $L(M) = \{w | \hat{\delta}(q_0,w) \in F\}$. A language is called a regular language if some DFA recognizes it.
- DFAs as language recognizers with an input tape, a control unit, and a cursor: Input string $w \in \Sigma^* \to$ DFA $\to$ accept/reject.

Example (Sipser p. 36): A DFA that accepts strings over the alphabet $\{0,1\}$ that have at least one 1 and an even number of 0s after the last 1. (Given a DFA M, what is $L(M)$?)
Example: A DFA that accepts strings over alphabet $\{0,1\}$ that have even numbers of 0s and 1s. (Given a language A, what is the DFA M such that $L(M) = A$?)

2.3 Nondeterministic finite automata

Reading: Sipser 1.2 (pp. 47-54)

- NFA $N = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an alphabet, $q_0 \in Q$ is a start state, $F \subseteq Q$ is a set of accept/final states, and $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ is a transition function, where $\delta(q,a) = P$ is the set of states that N may enter if the current state is q and the current symbol is a. In the case of $\delta(q,\varepsilon) = P$, N ignores the current input symbol and transitions from the current state q to any state in P.
- How does an NFA accept/reject a string? The meaning of nondeterminism.
- ε-closure: For any $P \subseteq Q$, $E(P)$ is the set of all states reachable from any state in P via zero or more ε-transitions.
- Extending δ to $\hat{\delta} : Q \times \Sigma^* \to 2^Q$: For any $q \in Q$, define $\hat{\delta}(q,\varepsilon) = E(\{q\})$ and $\hat{\delta}(q,w) = E(\cup_{i=1}^{k} \delta(p_i,a))$ if $w = xa$ and $\hat{\delta}(q,x) = \{p_1, \ldots, p_k\}$.

- Language of an NFA N (or language recognized by N) is $L(N) = \{w|\hat{\delta}(q_0,w) \cap F \neq \emptyset\}$.
- NFA as a language recognizer: NFA accepts a string if there is a path from the start state to some accept state, and it rejects a string if there is no path from the start state to any accept state.

Example: An NFA that accepts decimal numbers (a number that may have $+$ or $-$ preceeding it, but must have a decimal point).

2.4 Equivalence of DFAs and NFAs

Reading: Sipser 1.2 (pp.54-58)
The subset construction method: Given NFA $N = (Q,\Sigma,\delta,q_0,F)$. Construct a DFA $M = (Q',\Sigma,\delta',q_0',F')$ as follows such that $L(M) = L(N)$.

- Q' is the power set of Q, i.e., Q' contains all the subsets of Q. Note that if $|Q| = n$ then $|Q'| = 2^n$. This is just the worst case. Since many states in M are inaccessible or dead-end states and thus may be thrown away, so in practice, $|Q'|$ may be much less than 2^n.
- $q_0' = E(\{q_0\})$.
- $F' = \{R \in Q' | R \cap F \neq \emptyset\}$.
- For each $R \in Q'$ and each $a \in \Sigma$, $\delta'(R,a) = E(\cup_{p \in R} \delta(p,a))$.

After a DFA is constructed with the above method, all inaccessible and dead-end states with associated arcs may be removed to simplify the DFA.
Theorem: The equivalence of DFAs, NFAs, and regular languages.
Example: A DFA that accepts decimal numbers (converted from the previous NFA).
Example (Sipser p. 51): A four-state NFA and an eight-state DFA that recognize the language consisting of strings over $\{0,1\}$ with a 1 in the third position from the end.
Example: A bad case for the subset construction: $|Q_N| = n+1$ and $|Q_D| = 2^n$.

2.5 Closure under regular operations

Reading: Sisper 1.1 (pp. 44-47) and 1.2 (pp. 58-63)

- Closure under union: If A and B are regular, so is $A \cup B$.
- Closure under concatenation: If A and B are regular, so is AB.
- Closure under star: If A is regular, so is A^*.
- Closure under complementation: If A is regular, so is $\overline{A}$ (which is $\Sigma^* - A$).
- Closure under intersection: If A and B are regular, so is $A \cap B$.
- Closure under difference: If A and B are regular, so is $A - B$.
- Closure under reverse: If A is regular, so is A^R (where $A^R = \{w^R | w \in A\}$ and w^R is the reversed w).
- Closure under homomorphism: If A is regular, so is $h(A)$ (where $h(A) = \{h(w)|w \in A\}$ and $h : \Sigma \to (\Sigma')^*$ is a homomorphism).
- Closure under inverse homomorphism: If A is regular, so is $h^{-1}(A)$ (where $h^{-1}(A) = \{w|h(w) \in A\}$).

Example: Prove that $A = \{w \in \{a,b\}^* | w$ is of odd length and contains an even number of a's$\}$ is regular. (A is the intersection of two regular languages.)