COMS 3261 Handout 3A: Finite Automata Review

Angel (Leyi) Cui and Jeannie Ren

Fall 2022

1 Basic

1.1 Definitions

- Alphabet: A finite set of characters/symbols, denoted Σ.
  - Ex: $\Sigma_1 = \{0, 1\}$, $\Sigma_2 = \{a, b, c\}$, $\Sigma_3 = \{0, 1, ..., y, z, +, =, (,)\}$
- String: A sequence of characters/symbols concatenated together.
  - Ex: $w_1 = 1010, w_2 = hello, w_3 = (1 + 1) - 5, w_4 = \varepsilon$
  - Length of a string: The number of characters/symbols in a string, denoted $|w|$.
    - Ex: $|w_1| = 4, |w_2| = 5, |w_3| = 7, |w_4| = 0$
  - Concatenation of Strings: $xy = x \cdot y$.
    - $|xy| = |x \cdot y| = |x| + |y|$
    - Ex: $1010 \cdot 01 = 101001, 1010 \cdot \varepsilon = 1010$
- Σ^k: All strings over Σ of length k.
  - Ex: $\{0, 1\}^0 = \{\varepsilon\}$, $\{0, 1\}^0 \cup \{0, 1\}^1 \cup \{0, 1\}^2 = \{\varepsilon, 0, 1, 00, 10, 01, 11\}$
- Σ^*: The set of all possible strings made from characters/symbols in the alphabet Σ.

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^k = \{\epsilon\} \cup \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

  - Ex: For Σ= $\{0, 1\}$, $\Sigma^* = \{\varepsilon, 0, 1, 11, 01, 10, 11, 000, 001, \dots\}$
- Language: A set of strings over an alphabet, namely a subset of Σ^*.
  - Ex: $L_1 = \{w|w$ has an odd number of 0's and an odd number of 1's$\}$
    - $\{01, 10, 0001, 000111, 111000, 010101, 01010111, ...\}$
  - Ex: $L_2 = \{w|w$ is a word in the CST textbook$\}$
    - {computer, science, theory, ...}

1.2 Language Operations

- Concatenation: $L_1 \cdot L_2 = \{xy | x \in L_1 \text{ and } y \in L_2\}$, the set of all possible strings that can be formed from the concatenation of one string in L_1 and one string in L_2.
  - Ex: $L_1 = \{w \in \{0,1\}^* | w$ consists of only 0's and $|w|$ is even$\}$, $L_2 = \{w \in \{0,1\}^* | w$ consists of only 1's and $|w|$ is odd$\}$
    - $L_1 \cdot L_2 = \{w | w$ consists of an even numbers of 0's followed by an odd number of 1's$\}$
  - Ex: $L_3 = \{a, ab\}$, $L_4 = \{\varepsilon, b, bbb\}$
    - $L_3 \cdot L_4 = \{a, ab, abbb, abb, abbbb\}$
- Star: $L^* = \{w | w = \varepsilon \text{ or } w = w_1 w_2 \dots w_n \text{ where } w_1, w_2, ..., w_n \in L\}$, the concatenation of zero or more strings in L.
  - Ex: If $L = \{$good, bad$\}$, then $L^* = \{\varepsilon$, good, bad, goodbad, goodgood, badgood, badbad, ...$\}$
- Complement: $\overline{L} = L^c = \{w \in \Sigma^* | w \notin L\}$, every possible string over the alphabet Σ that is not already in L.
  - Ex: If $L_1 = \emptyset$ then $\overline{L_1} = \Sigma^*$.
- Union: $L_1 \cup L_2 = \{x | x \in L_1 \text{ or } x \in L_2\}$
  - $L_1 = \{ax | x \in \{a,b\}^*\}$
  - $L_2 = \{xa | x \in \{a,b\}^*\}$
  - $L_1 \cup L_2 = \{ax \text{ or } xa | x \in \{a,b\}^*\}$
- Intersection: $L_1 \cap L_2 = \{x | x \in L_1 \text{ and } x \in L_2\}$
  - Extending from the previous example: $L_1 \cap L_2 = \{axa | x \in \{a,b\}^*\}$

2 Finite Automata

2.1 Deterministic Finite Automata (DFA)

- $D = (Q, \Sigma, \delta, q_0, F)$
  - Q: finite set of states
  - Σ (Alphabet): (finite) alphabet
  - δ (Transition Function): $Q \times \Sigma \to Q$. Every state, symbol pair goes to exactly one state.
  - q_0: start state
  - F: set of accepting states
- Computation: Read each character in the input string w one at a time (left to right), starting at q_0, and following transitions specified by δ. If after reading all of the input, the computation ends in an accepting state, we say that w is accepted by the DFA. Otherwise, we say that w is rejected by the DFA.
- Language recognized by a DFA: the set of all strings that are accepted by a DFA, $L(D)$

- For L to be the language recognized by D, every string in L must be accepted by D and every string not in L must be rejected by D.
- A language L is **regular** if there is a DFA D that $L(D) = L$.
  * Every finite language is regular, but not every regular language is finite.
  * The empty language $\emptyset$ is a regular language.
  * If L_1 and L_2 are regular languages, then $\overline{L_1}$, $L_1 \cup L_2$, $L_1 \cdot L_2$, $L_1 \cap L_2$, L_1^* are all regular languages.
  * Equivalently: The class of regular languages is closed under the complement, union, concatenation, intersection, and star operations.

2.2 Nondeterministic Finite Automata (NFA)

- $N = (Q, \Sigma, \delta, q_0, F)$
  - Q: finite set of states
  - Σ (Alphabet): (finite) alphabet
  - δ (Transition Function): $Q \times (\Sigma \cup \{\varepsilon\}) \to P(Q)$. Every state, symbol pair goes to a subset of states (namely in the powerset of Q).
  - q_0: start state
  - F: set of accepting states
- Computation: Read each character in the input string w one at a time (left to right), starting at q_0, and following transitions specified by δ, with an option to take an epsilon transition at any time one is available (pursue all options in parallel). After reading all of the input, if at least one of the computation paths ends in an accepting state, we say that w is accepted by the NFA. Otherwise, if all possible comptuation paths end in a non-accepting state, we say that w is rejected by the NFA.
- Language recognized by a NFA: the set of all strings that are accepted by an NFA, $L(N)$
  - Every string in L must be accepted by N (ie. there exists some accepting computation), and every string not in L must be rejected by N (ie. rejected by all possible computation paths).
  - A language L is **regular if and only if** there is an NFA N that $L(N) = L$.
- Equivalence with DFAs: Using the subset construction, we can transform any NFA into an equivalent DFA, which means that NFAs and DFAs recognize the same set of languages.
  - **Note:** A language can be recognized by multiple DFA/NFA, but a DFA/NFA can only recognize a single language.