Formal Languages Final Study guide

Liam Dillingham

May 2, 2019

1 Definitions

1.1 Strings

- Σ: alphabet. An alphabet is a *finite* set of symbols (not including ϵ)
- Σ^k: strings from the alphabet Σ of length k
- Σ^*: The set of all strings over an alphabet (including Σ^0 i.e. ϵ).
- Σ^+: set of non-empty strings
- A language L is a set of strings from the alphabet Σ^* such that $L \subseteq \Sigma^*$

1.2 Finite Automata

1.2.1 Deterministic Finite Automata

A *DFA*, labeled as A, is defined as $A = (Q, \Sigma, \delta, q_0, F)$, such that:

1. Q: a finite set of *states*
2. Σ: a finite set of *input symbols*
3. $\delta(q, a)$: a transition function with arguments as q: the current state, and a: the current input symbol, where $\delta : Q \times \Sigma \to Q$
4. q_0, or the starting state in Q
5. F: The set of final or accepting states such that $F \subseteq Q$.

Extended Transition Function $\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$
The *extended transition function* precisely describes what happens when we start in any state and follow any sequence of inputs i.e. defines δ for whole words instead of symbols

Language of DFA if A is a DFA, then $L(A) = \{w \mid w \in \Sigma^* \text{ and } \hat{\delta}(q_0, w) \in F\}$

1.2.2 Nondeterministic Finite Automata

The only difference between a *DFA* and *NFA* is that for an *NFA*, δ maps to a set of states. that is, $\delta : Q \times \Sigma \to 2^Q$ i.e. $\mathcal{P}(Q)$

Extended Transition Function

$$\text{basis: } \hat{\delta}(q, \epsilon) = q. \text{ induction: } \hat{\delta}(q, w) = \hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)$$

1.2.3 ϵ-Nondeterministic Finite Automata

For ϵ-*NFA*, we explicitly define transitions for ϵ, i.e. $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$

Extended Transition Function

- **ECLOSE(q)**: All the states that q can reach using only ϵ
- **ECLOSE(S)**: $\bigcup_{r \in S}$ **ECLOSE(r)**, where S is a set of states

For the precise definition, we have:

$$\text{basis: } \hat{\delta}(q, \epsilon) = \textbf{ECLOSE}(q). \quad \hat{\delta}(q, w) = \hat{\delta}(q, xa) = \textbf{ECLOSE}\left(\bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a) \right)$$

The language described by an ϵ-*NFA*, A, is defined as: $A = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$.

1.3 Regular Expressions

1.3.1 Operators

Union: if $L = \{001, 10, 11\}$ and $M = \{$

1.4 Properties of Regular Languages

1.4.1 The Pumping Lemma

The pumping lemma for regular languages Let L be a regular language. Then there exists a constant n (which depends on L) such that for every string w in L such that $|w| \geq n$, we can break w into three strings, $w = xyz$ such that:

1. $y \neq \epsilon$
2. $|xy| \leq n$
3. For all $k \geq 0$, the string xy^kz is also in L