Formal Languages and Automata Theory
Homework Set #3

William Shoaff

Summer 2018 (May 27, 2018)

Definitions: The language of a DFA M is

$$L(M) = \{w : \hat{\delta}^*(q_0, w) \in \mathbb{F}\}$$

That is, a string w is in the language of M if there is a path from the start state q_0 to a final state in $\mathbb{F}$ found by repeatedly following state transitions using the transition function δ. For NFA's the transition function maps (state, character) pairs to subsets of states. ϵ-transitions are also allowed.

The language of an NFA N is

$$L(N) = \{w : \hat{\delta}^*(q_0, w) \cap \mathbb{F} \neq \emptyset\}$$

That is, some path labeled w from the start state to a final state exists.
Theorem: The class of languages recognized by DFA's and NFA's are equivalent.

Proof. 1. Every DFA is also an NFA so

$$\{L(M) : M \text{ is a DFA}\} \subseteq \{L(N) : N \text{ is a NFA}\}$$

2. The Rabin–Scott subset construction algorithm establishes the other inclusion (states become subsets of states).

□

Regular Expressions: The empty set $\emptyset$, empty string ϵ, and each character $c \in \Sigma$ is a regular expression. If r_1 and r_2 are regular expressions, then the following are regular expressions:

$$r_1 + r_2 = r_1 | r_2 = r_1 \cup r_2 \qquad \text{(precedence 0)}$$
$$r_1 \cdot r_2 \qquad \text{(precedence 1)}$$
$$r_1^* \qquad \text{(precedence 2)}$$
$$(r_1) \qquad \text{(precedence 3)}$$

Problems:

1. (2 pts) Give an NFA with ϵ-transition that recognizes binary numbers: An optional $+$ or $-$ sign; a string of bits; a binary point; and another string of bits, where at least one of the bit strings is non-empty.

2. (2 pts) Use the Rabin–Scott algorithm to convert the NFA of problem (1) into a DFA.

3. (2 pts) Let A and B be finite automata. Show how to construct a finite automata that accepts

(a) The union of their languages $L(A) \cup L(B) = L(A) + L(B)$

(b) The intersection of their languages $L(A) \cap L(B)$

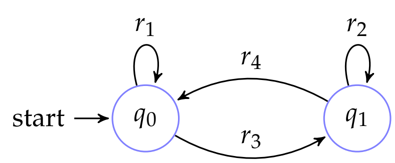
(c) The concatenation of their languages $L(A) \cdot L(B)$

(d) The Kleene closure $L(A)^*$ of $L(A)$.

4. (2 pts) A general transition graph (GTG) is a transition graph with edges labeled by regular expressions. It is complete (on n states) when all n^2 edges are present.

(a) Prove that a complete, directed, generalized transition graph with self-loops on n nodes has n^2 edges.

(b) What regular expression is equivalent to the two-state finite automata below? How does your answer depend on choice of final state?

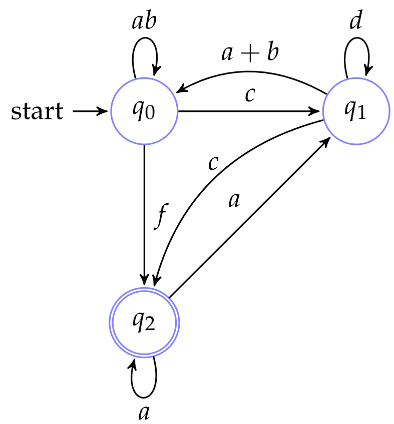


5. (2 pts) Consider 3-state the incomplete graph below.

(a) Complete the generalized transition graph (GTG) by adding regular-expression labeled edges as needed.

(b) Show how to remove a state q_1, converting the complete GTG to a 2-state machine.

(c) Finally, find a regular expression, from the 2-state machine, that describes the language of the machine.



Total Points: 10