CS 138: MID-QUARTER EXAMINATION 1

Department of Computer Science
University of California, Santa Barbara
Closed-Book, 75 minutes
Fall 2004

INSTRUCTIONS

• *Before* you answer any questions, print your name and perm number.

• Read each question carefully. Make sure that you clearly understand each question before answering it.

• Put your answer to each question on its own page.

• You may wish to work out an answer on scratch paper before writing it on your answer page; answers that are difficult to read may lose points for that reason.

• You may not leave the room during the examination, even to go to the bathroom.

• You may not use any personal devices, such as calculators, PDAs, or cell phones.
1. (5 points) Let

\[
\begin{align*}
    f(n) &= 2f(n-1) + 1, \text{ for } n > 0, \\
    f(0) &= 0.
\end{align*}
\]

Is \( f(n) = 2^n - 1 \), for \( n \geq 0 \)? Prove your answer.

**Answer**

Yes, it is.

**Proof** by induction on \( n \):

**Basis** \( n = 0 \): \( f(0) = 0 = 2^0 - 1 \).

**Induction hypothesis**: \( f(n) = 2^n - 1 \).

**Induction step**: We show that \( f(n + 1) = 2^{n+1} - 1 \):

\[
\begin{align*}
    f(n + 1) &= 2f(n) + 1 \\
    &= 2(2^n - 1) + 1 \\
    &= 2^{n+1} - 1.
\end{align*}
\]
2. (10 points) If the only difference between language $L$ and $L^*$ is the word $\lambda$, is the only difference between $LL$ and $L^*$ the word $\lambda$? Explain your answer.

**Answer**

No, $\lambda \in L^*$, regardless of what is in $L$. Since the only difference between language $L$ and $L^*$ is the word $\lambda$, $\lambda \notin L$. For a counterexample, let $L = \{a^n : n > 0\}$. Then, $L^* = \{a^n : n \geq 0\}$. Since $LL = \{w_1w_2 : w_1, w_2 \in L\}$ and $\lambda \notin L$, $\lambda \notin LL$.

In general, let $P$ be the set of words in $L$ that cannot be factored (i.e., they cannot be written as the concatenation of 2 or more words in $L$). $P \subseteq L \subseteq L^*$ but $P \cap LL = \emptyset$. 


3. (15 points) Give a transition graph for a DFA that is equivalent to the finite accepter below.

![Diagram of an NFA with states q0, q1, and q2, transitions labeled with λ, 0, 1, and 0, 1, 1.]

**Answer**

There are many DFAs that are equivalent to the given NFA.

- The NFA accepts \( \{0, 1\}^* \). To see this, note that, initially, the NFA can get to state \( q_1 \), a final state (accepting \( \lambda \)). Once in that state, regardless of what string of 0s and 1s follows, it can stay in that state. Thus, it accepts \( \{0, 1\}^* \). This is reflected in the DFA below.

![Diagram of a DFA with a single state acceptor accepting \( \{0, 1\}^* \).]

- Another equivalent DFA follows.
Yet another follows.
4. Let language $L = \{ w \in \{0, 1\}^* : w$ begins and ends with the same symbol, and the second symbol in $w$ is the same as its second-to-last symbol $\}$. For example, 01101110 $\in L$ and 101010 $\notin L$.

(a) (15 points) Give a regular expression that denotes $L$ or explain why this cannot be done.

**Answer**

A regular expression for this is

$$(00(0 + 1)^*00) + (01(0 + 1)^*10) + (10(0 + 1)^*01) + (11(0 + 1)^*11)$$

(b) (15 points) Give an NFA (transition graph) that accepts $L$ or explain why this cannot be done.

**Answer**

A transition graph for this is given below.
5. (10 points) Give a transition graph that accepts the language denoted by \((a*b)^*a^*\) or explain why this cannot be done.

**Answer**

A transition graph for this is given below.

Let \(\Sigma = \{a, b\}\). Any \(w \in \Sigma^*\) has \(n\) \(b\) symbols, for some \(n \geq 0\). Let \(w\) have exactly \(n\) \(b\) symbols. Then, \(w\) is contained in the language denoted by

\[
\underbrace{(a*b) \cdots (a*b)}_{n} a^*.
\]

Thus, the language denoted by \((a*b)^*a^*\) has every string in \(\Sigma^*\). Since it cannot have any strings that are not in \(\Sigma^*\),

\[(a*b)^*a^* = \Sigma^*.
\]

Therefore, the following transition graph also is correct:
6. (30 points) Let $L_1$ and $L_2$ be regular languages. Show that $L_1 \cap L_2$ is a regular language. (Hint: DeMorgan’s law)

**Answer**

**Claim 1:** If $L$ is a regular language, $\overline{L}$ is regular.

**Proof:** Assume $L$ is regular, and let DFA $M$ accept it. Create $\overline{M}$ by copying $M$, except that $\overline{M}$’s final states are exactly those states that are not final in $M$. Clearly, $\overline{M}$ accepts exactly those words that $M$ does not: $L(\overline{M}) = \overline{L}$.

**Claim 2:** If $L_A$ and $L_B$ are regular, then $L_A \cup L_B$ is regular.

**Proof:** Assume $L_A$ and $L_B$ are regular, and let them be denoted by regular expressions $r_A$ and $r_B$, respectively. Then $L_A \cup L_B$ is denoted by $r_A + r_B$.

Invoking claims 1 and 2,

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

is regular.