Computability and Logic Homework 3

Due: Thursday, September 22, 2005

Each of problems 1 through 3 is worth 10 points. Problem 4 is worth 20 points.

1. Homorphisms A *homomorphism* of strings is a function $f : \Sigma^* \to \Sigma^*$ such that $f(\Lambda) = \Lambda$ and for any $u, v \in \Sigma^*$, $f(uv) = f(u)f(v)$. For any language $L \subseteq \Sigma^*$, $f[L]$ denotes the image of the set L under f, that is:

$$f[L] = \{f(w) \mid w \in L\}$$

and $f^{-1}[L]$ denotes the inverse image of L under f, that is:

$$f^{-1}[L] = \{w \mid f(w) \in L\}.$$

a. Prove **by induction** that if L is regular, then $f[L]$ is regular. (The case for star is the most interesting.)

b. Sketch a proof of the same fact using finite automata. (Hint: at least one of the automata featured in the proof should probably be nondeterministic.)

c. Argue, however you like, that if L is regular then $f^{-1}[L]$ is regular. Your proof need not be formal.

2. Divisibility and Regularity Let $\Sigma = \{0, \ldots, 9\}$. For each string $w \in \Sigma^*$, define $[\![w]\!]$ to be the integer "denoted" by w, as follows: $[\![\Lambda]\!] = 0$, and for any string w and digit i, $[\![wi]\!] = 10[\![w]\!] + i$.

a. Prove that each of the following languages is regular:

1. $\{w \mid [\![w]\!]$ *is even*$\}$
2. $\{w \mid [\![w]\!]$ *is a multiple of* $4\}$
3. $\{w \mid [\![w]\!]$ *is a multiple of* $3\}$

b. Prove or disprove: For any $n > 1$, the language $D_n = \{w \mid [\![w]\!]$ *is a multiple of* $n\}$ is regular.

3. Indistinguishability

a. What are the equivalence classes of $\equiv_L$ (the indistinguishability relation defined in class) for the language $L = \{a^n b^n \mid n \geq 1\}$?

b. What about for $L = \{w \in \{a, b\}^* \mid w$ *has the same number of* a*'s as* b*'s*$\}$?

c. Which, if either, of these languages is/are regular?

4. Subset Construction and State Minimization Construct a minimum-state DFA to accept the language $aa(a + b)^* + (a + b)^*bb$ by taking the following three steps:

1. Construct an NFA that accepts the language.
2. Convert that NFA to an equivalent DFA.
3. Perform the state minimization algorithm on that DFA.

Show the automaton you construct in each step.