

CSE 311: Foundations of Computing I

Assignment #9

November 27, 2013

Due: Friday December 6, 2013

Problems

1. Use the method given in class to design a linear-time algorithm to determine all occurrences of the string 10100101000 in strings over the alphabet $\{0,1\}$.
2. Prove that the set of all binary strings with fewer 1's than 0's is not recognized by any DFA.
3. Consider the candy machine problem from HW#8. Suppose that the specification was such that the user could put an amount $\$X$ into the machine and press "G" for gumball or "M" for M&M's, and the machine would then return $\$(X-0.10)$ change or $\$(X-0.15)$ change, along with the specified candy. For instance, if the user put $\$4.75$ in the machine and hit "M" they would get $\$4.60$ in change. Prove that this functionality cannot be implemented with a finite state machine.
4. A *rational quadratic polynomial* is a function of the form $f(x) = px^2 + qx + r$ where $p \neq 0$, q , and r are rational numbers. Show that the set of all rational quadratic polynomials is countable.
5. Let B be the set of all infinite sequences composed of symbols from $\{0,1,2\}$. Show that B is uncountable using a proof by diagonalization.
6. Show that the following problem is undecidable using the fact that the Halting Problem is undecidable: Given the code $\langle P \rangle$ of a Java program P and an input x , determine whether or not P prints out x and then halts.
[This problem is optional and will not be graded]

EXTRA CREDIT #1:

Show that the following problem is undecidable: Given the code of two Java programs $\langle P \rangle$ and $\langle Q \rangle$, determine whether or not there is any input on which P and Q print the same output.

EXTRA CREDIT #2:

In this question you are to design a finite state controller for the traffic lights at an intersection of a busy two-way East-West street and a small two-way street that heads South of the E-W street but does not cross it: There is a left-turn lane for the westbound traffic on the E-W street to turn Southbound and there is a sensor L that detects traffic waiting in this left-turn lane. There is also a sensor N to detect waiting Northbound traffic on the small street. We use input notation 0 to denote that neither sensor detects a car, B to denote that both sensors detect cars, and L and N to denote that only one of the sensors is activated.

There are 4 traffic lights: Eastbound, Westbound, Northbound, and Left-turn that each cycles from Green to Yellow to Red back to Green in response to input signals. (We think of these outputs as EG, EY, ER, etc.) The length of time that a Yellow occurs is governed by a timer signal T that ends it. (We won't worry about how T is activated.)

Under normal circumstances, the Eastbound and Westbound lights are Green and the Northbound and Left-turn lights are Red. If there is waiting Northbound traffic, then all other lights must turn to Yellow and then to Red before the Northbound light can turn Green. Northbound traffic is rare enough that it takes precedence over all other traffic. If there is waiting Left-turn traffic, then the Eastbound and Northbound lights must turn to Yellow and then to Red before that traffic can turn left but Westbound traffic is unaffected. Left-turn traffic takes precedence over everything but Northbound traffic. So that the lights don't switch back instantly, after the Northbound light turns Green it must stay Green so long as there is still waiting traffic and for at least two timer signals before it turns Yellow. (There is no such requirement for the length of Green at any other traffic light.)