CIS 511 Homework 4

Stephen Phillips, Dagaen Golomb

February 23, 2015

Problem 1

We want to show that finding if a Turing Machine has a *useless state* is Turing Decidable. We formulate this as a language:

$$L = \{\langle M, q\rangle \mid q \in Q(M),\ \forall s \in \Sigma^* M(q)\, M \text{ does not enter } q\}$$

Now we reduce this to A_{TM} to show that it is undecidable. First as usual, we suppose toward contradiction that there was a decider of this language, which we will denote D. We build a Turing Machine in the following manner:

function N($\langle M, x\rangle$)
  Build the description for the following machine
  function A(y)
    Ignore input y
    Simulate $M(x)$, and output what it outputs
  end function
  Simulate $D(\langle A, q_A\rangle)$, where q_A is the accept state of D, and output what it outputs
end function

If $M(x)$ halts and accepts, then we eventually reach the accept state of A, by construction. Also by construction we never reach the accept state if $M(x)$ loops or if it rejects. Since D is a decider, and building the machine description takes finite time, N always halts. Therefore N accepts if and only if D accepts, which in turn accepts if and only if $\langle M, x\rangle \in A_{TM}$. Therefore, N decides A_{TM}, a contradiction.

Problem 2

Note that the set of decidable languages is countable. If a language is decidable, there must exist a Turing Machine which decides it. We proved in class the set of TMs is countable (via enumerating descriptions of all combinations of bits starting at length 1 and increasing onward). The set of TM's that decides subset languages in $\{1\}^*$ is a subset of all TM's and is therefore countable. Thus, the set of decidable languages in $\{1\}^*$ is countable.

Now, using diagonalization, we can prove that the set of all languages of subsets of $\{1\}^*$ is uncountable. This will show that there must be some language that is a subset of $\{1\}^*$ that is not decidable since the set of TM's deciding languages of this form is countable.

Picture the set of languages that are a subset of $\{1\}^*$ as a bit string with 0 if they include a particular string in $\{1\}^*$ and 0 otherwise. Suppose this set of languages is countable. Then we could list all such languages as so:

ϵ	1	11	111	...
0	0	0	0	
1	0	0	0	
0	1	0	0	
1	1	0	0	
$\vdots$				$\ddots$

Now create the languages where if the i^{th} entry in the i^{th} row is one, that string is not in the language. Conversely, if it is a zero it is in the language. Now, we have created a language that is clearly composed of substrings of $\{1\}^*$, so it should be in the table. However, this language conflicts with every table entry, by construction, by at least one string. But this table is supposed to list all languages of the given form. This is a contradiction. Therefore, the set of languages that are a subset of $\{1\}^*$ is uncountable.

Thus, we have shown that there must exist a language that is a subset of $\{1\}^*$ that cannot be decided.

Problem 3

We want to show that the intersection of two context free languages is undecidable using A_{TM}.

Problem 4

Problem 5

Show that the language

$$ISO = \{\langle G, H\rangle\} \mid G \text{ and } H \text{ are isomorphic graphs}$$

is in NP.

To do this we need to show a verifier V for ISO. As one might expect, the certificate for a member of the language $\langle G, H\rangle$ is the isomorphism $\phi : V_G \to V_H, v \to \phi(v)$ between them. The size of such an isomorphism would be about $2n$ where n is the number of nodes in G and H, so it polynomial in the size of G and H. We also need to check that the verifier V using this runs in polynomial time. A simple algorithm to use this is the following:

function V($\langle G, H\rangle$,ϕ)
  If the number of nodes or edges in G differ from the number of nodes or edges in H, reject
  for Nodes v in G **do**
    If the number of edges (v, u) in G differ from the number of edges $(\phi(v), u')$ in H, reject
    for Edges (v, u) in G **do**
      If $(\phi(v), \phi(u))$ is not in H, reject
    end for
  end for
  Accept
end function

By assering that the size of the graphs are the same, that each node v has the same degree as $\phi(v)$ and maps to the corresponding vertices, we have shown that the isomorphism is correct. If there is no isomorphism then at least one of these tests will fail. Therefore $V(\langle G, H\rangle, y)$ will accept for some input y if and only if there is an isomorphism between G and H

Problem 6

Problem 7