Recommended problems
10/18/2005
CSCI 2670

4.6 Assume B is countable and that $f{:}N \rightarrow B$ is a functional correspondence. Let $f_i(j)$ denote the i^{th} symbol in $f(j)$. Consider the string s whose i^{th} element differs from $f_i(i)$ – i.e. if $f_i(i) = 0$, then the i^{th} symbol in s is 1 and vice versa. Then s cannot be in the image of f since it differs from every string in the image of f by at least one symbol. Furthermore, s is in B since it is an infinite sequence over $\{0,1\}$. Therefore, no functional correspondence can exist between N and B – i.e., B is uncountable.

4.7 We begin by lexicographically arranging the elements of T according to the sum of the elements $\{(1,1,1),(1,1,2),(1,2,1),(2,1,1),(1,1,3),(1,2,2),(1,3,1),(2,1,2),(2,2,1),(3,1,1),\ldots\}$. For each natural number i, let $f(i)$ be the i^{th} element in the list. Since the list never repeats, f is clearly one-to-one. Also, for every element t in T, the sum of the components of t is finite. Therefore, there is some natural number n such that $f(n)=t$. Thus, f is onto. This shows that there is a functional correspondence from N to T – i.e., T is countable.

4.2 $\text{USELESS}_{\text{PDA}}$ = {<P> | P is a PDA with at least one useless state}.
2
First let's consider the language E_{PDA} = {<P> | P is a PDA with L(P)=∅}. The following TM decides E_{PDA}

M = "On input <P>

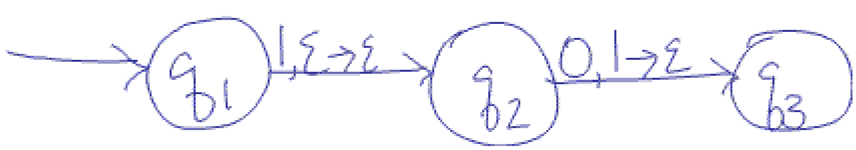
1. Convert P to a CFG G
2. Submit <G> to the TM that decides E_{CFG}
3. If <G> is accepted, then **accept**
4. If <G> is rejected, the **reject**"

Notice that if P has exactly one accept state and L(P) = ∅, then the accept state is a useless state. We can use this to determine if a specific state q_i of P is useless – let P_i be the PDA found by modifying P so that q_i is its only accept state and submit <P_i> to the TM M described above. To determine if P has *any* useless states, do this for each state in P. More specifically, the following Turing machine decides $\text{USELESS}_{\text{PDA}}$

N = "On input <P>

1. Let n = the number of states in P
2. For each i = 1, 2, … n
   a. Let P_i be the PDA found by modifying P so that its only accept state is the i^{th} state of P
   b. Submit <P_i> to M (the TM that decides E_{PDA})
   c. If M accepts, then **accept** (the i^{th} is useless)
3. If M didn't accept any <P_i>, then **reject**"

Note that is is ***not*** sufficient to just see if there is some sequence of transitions from the start state to every other state. The stack contents must also be considered. For example, consider the following PDA.



In this PDA, q_3 is useless even though there is a transition to it. This is because there is no way to push a 1 onto the stack and q_3 can only be reached by popping a 1 off of the stack.

4.25 The intersection of a context-free language and a regular language is context free. Also, the language $\{w \in \{0,1\}^* \mid w \text{ has more 1's than 0's}\}$ is generated by the following context-free grammar, G.

$S \rightarrow T1T$
$T \rightarrow T0T1T \mid T1T0T \mid 1T \mid \varepsilon$

Consider the following Turing machine

M = "On input <D>, where D is a DFA
1. Construct a CFG H such that $L(H) = L(D) \cap L(G)$
2. Submit <H> to the decider for E_{CFG}
3. If it rejects, then **accept**
4. Else, **reject**."

M is clearly a decider since every step of M halts. Also, M accepts iff the intersection of L(D) and L(G) is non-empty – i.e., iff D accepts some string with more 1's than 0's.

4.26 Consider the following Turing machine

M = "On input <G,x>, where G is a CFG
1. Let R be the following RE $R = \Sigma^* x \, \Sigma^*$
2. Create the CFG H such that $L(H) = L(G) \cap L(R)$
2. Submit <G > to the decider for E_{CFG}
3. If it accepts, then **reject**
4. Else, **accept**."

M is clearly a decider since each step halts. Also, M rejects <G,x> iff G generates some string in $\Sigma^* x \, \Sigma^*$ -- i.e., some string with x as a substring. Therefore C_{CFG} is decidable.

The following TM also decides C_{CFG}

T = "on input <G,x>, where G is a CFG
1. For i = 0 to len(x)
2.   Let y = x[i]:x[len(x)-1]
3.   For j = 0 : len(y)
4.   Submit <G,y[0]:y[j-1]> to the decider for A_{CFG}
5.   If it accepts, **accept.**
6.   Next j
7. Next i
8. **reject**