Final , Formal Lang, Logic,

Sherri Shulman
TESC, CS

Mar 9, 2011

1. Construct a Turing machine with input alphabet $\{a, b\}$ to accept the language $\{a^i b^j | i \geq 0, j \geq i\}$ by final state.

2. Alter your solution to the above problem to obtain a Turing machine that accepts by halting.

3. Construct a two-tape Turing machine with input alphabet $\{a, b\}$ that accepts strings with the same number of a's and b's. The computation with input u should take no more than $2 * \text{length}(u) + 3$ transitions.

4. Construct a Turing machine (with no macros) to compute the following number-theoretic functions:

   (a)
   $$even(n) = \begin{cases} 1 & \textit{if } n \textit{ is even} \\ 0 & \textit{otherwise} \end{cases}$$

   (b)
   $$lt(n, m) = \begin{cases} 1 & \textit{if } n < m \\ 0 & \textit{otherwise} \end{cases}$$

   For each machine provide one example showing the action of your machine on a sample input.

5. Design a machine that computes:

   $$gt(n, m) = \begin{cases} 1 & \textit{if } n > m \\ 0 & \textit{otherwise} \end{cases}$$

   You can use macros, including the one that you created in the above problem. For each machine provide one example showing the action of your machine on a sample input.

6. Trace the actions of the machine $MULT$ for computations with input:

   (a) $n = 0, m = 4$
   (b) $n = 1, m = 0$
   (c) $n = 2, m = 2$‘

7. Let F be a Turing machine that computes a total unary number-theoretic function f. Design a machine to compute the function:

$$g(n) = \Sigma_{i=0}^{n} f(i)$$

8. Let G be the context-sensitive grammar:

$$\begin{aligned} G : & S \to SBA|a \\ & BA \to AB \\ & aA \to aaB \\ & B \to b \end{aligned}$$

   (a) Give a derivation of $aabb$
   (b) What is $L(G)$
   (c) Construct a context-free grammar that generates $L(G)$

9. Let L be the language $\{a^i b^{2i} a^i | i > 0\}$.
   (a) Construct a context-sensitive grammar G that generates L
   (b) Give the derivation of $aabbbbaa$ in G.
   (c) Give a design of a TM that accepts this language (note that I say design and not definition: use English to describe the pieces of the TM and what each section is intended to accomplish).

10. Design a two tape TM that determines if two strings u and v over $\{0, 1\}$ are identical. The computation begins with $BuBvB$ on the tape and should require no more thatn $3(length(u) + 1)$ transitions. (The limitation is guidance only: if you have a longer one, don't worry. But consider this lower bound.)

11. Construct a Turing machine that descides whether a string over $\{0, 1\}^*$ is the encoding of a nondeterministic Turing machine. What would be required to change this to a machine that decides whether the input is the representation of a deterministic Turing machine.

12. Translate the following sentences into predicate logic, put them in clausal form and prove the stated conclusion by resolution:
   (a) Suppose all barbers shave everyone who does not shave himself. Moreover, no barber shaves anyone who shaves himself. Conclude that there are no barbers.
   (b) Suppose John likes anyone who doesn't like himeself. Conclude that it is not the case that Jon likes no one who likes himself.

13. Translate the following into predicate logic.

i There exists a dragon.

ii The dragon either sleeps in its cave or hunts in the forest.

iii If the dragon is hungry, it cannot sleep.

iv If the dragon is tired, it cannot hunt.

Answer the following questions:

(a) Whast does the dragon do when it is hungry.

(b) What does the dragon do when it is tired? (Assume that if X cannot do Y then X does not do Y.)

14. Asume we have a unary function symbol s (with the intended meaning "successor of") to define a counter beginning with the constant 0. So 0 corresponds to 0, $s^n(0)$ corresponds to n. Do not try to use built-in arithmetic arithmetic operations supplied with Prolog. Write a program to calculate the length of a list using this counter. Demonstrate your program on the input $[a, b, [b, c]]$.

15. Write a program for the function $flatten$ that strips off all brackets from a list except the outermost. (NOTE! There are solutions to this on the web. If you use a solution, give the resolution, including the backtracking etc. showing how the flatten works.)

16. What is the advantage of modifying this program to *append′*: (this was a homework problem.)

    a1 a([],Y,Y):-!

    a2 a([X—Y],Z, [X—W]):-a(Y,Z,W).

    (a) Consider the situation when we have two given lists x and y and we want to find out the result of appending one to the front of the other, that is a goal of the form $? - a([x, y, z], [u, v], Z)$. Consider also goals of the form $? - a([x, y, z], V, W)$.

    (b) What problems arise in *append′* (in contrast to the same program without the cut) when considering a goal of the form $? - a(X, Y, [x, y, z])$. COnsider what happens when you try to get more than one answer substitution for the variables.

17. Give the type constraints generated for the following program fragment written in the language defined in the Cardelli paper (assume the type environment given on p. 7):

```
let rec FIB =
fun(n)
if ZERO(n) then
     1
else
     if ZERO(PRED(n)) then
          1
     else FIB(PRED(n)) + FIB(PRED(PRED(n)))
          in
          FIB(3)
```

18. Write a 1 page proposal for your project topic.