CS525 Winter 2012 \ Class Assignment #3, 2/8/2012

Aylin Caliskan

Nicholas Milkovits

Ariel Stolerman

4.10)

Let $INF_{PDA} = \{\langle M\rangle \mid M \text{ is a PDA and } L(M) \text{ is an infinite language}\}$. Following is a proof that INF_{PDA} is decidable:

A context-free language is infinite if there exists a cycle within its derivation rules. For PDAs, we can construct a CFG corresponding to any given PDA and test it. Therefore we can construct a Turing machine N that given the input M does as follows:

- Check if the input M is a valid encoding of a PDA. If not, *reject*.
- Create G a CFG that is equivalent to M, i.e. $L(M) = L(G)$, and convert G to Chomsky normal form.
- Look for a cycle in the grammar's rules in **BFS** (in order to avoid infinite loops) such that at any iteration on the cycle the generated string is pumped (i.e. the cycles describes a derivation of the form $R \overset{*}{\rightarrow} aRb$ where $|ab| > 0$ and a, b are terminals).
- If found such cycle, *accept*. Otherwise, *reject*.

4.12)

Let $A = \{\langle R, S\rangle \mid R \text{ and } S \text{ are regular expressions}, L(R) \subseteq L(S)\}$. Following is a proof that A is decidable. We will show a Turing machine M that decides A:

- Check that R, S are proper regular expressions, otherwise *reject*.
- Construct a NFA A' from the regular expression R (such that $L(A) = L(R)$), and then a DFA A from A'.
- Construct a NFA B' from the regular expression S (such that $L(B) = L(S)$), and then a DFA B from B'.
- Construct a DFA C that recognizes $L(A) \cap \overline{L(B)}$.
- Simulate the TM from the book that decides E_{DFA} on C. If it accepts, *accept*. Otherwise, *reject*.

Note that if $L(R)$ is not fully contained within $L(S)$ then $\exists w \in L(R) \wedge w \notin L(S) \Rightarrow w \in L(R) \cap \overline{L(S)}$. Furthermore, the construction of the NFAs and DFAs can be done using a Turing machine, and the intersection of regular languages is a regular language, so we can construct a DFA for it. Thus if the intersection above is discovered to be empty, $L(A)$ must be fully contained in $L(B)$, and so $L(R)$ is fully contained in $L(S)$.

4.22)

Let $L = \{\langle M \rangle | \ M\ is\ a\ PDA\ that\ has\ a\ useless\ state\}$. Following is a proof that L is decidable. We will construct a Turing machine M that decides L as follows:

- Check that M is a proper PDA, otherwise *reject*.
- For any state q in M:
  - Mark q as the only accepting state, and denote that PDA as M'.
  - Use the Turing machine that decides E_{PDA} on M'. If it accepts – *accept*.
- If got to this point, *reject*.

Clearly if marking any state q as the only accepting state, and the language recognized by that variant is empty, then there exists a useless state in the input PDA M.