Formal Languages Assigment 5

Rachel Hwang

November 14, 2013

- Exercise 0.1 *Design a PDA to accept each of the following languages.*
  1. $L = \{a^i b^j c^k : i = jorj = k\}$

  2. $L =$ the language of all strings with twice as many 0's as 1's

- Exercise 0.2 *Show that if P is a PDA, then there is a PDA $P2$ with only two stack symbols, such that $L(P2) = L(P)$. Hint: Binary-code the stack alphabet of P.*

  Let $P = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0\}$. We want to construct $P2 = \{Q', \Sigma, 0, 1, \delta', q_0', Z_0'\}$.
  Q' will contain Q, our Σ is the same, and we select a stack alphabet of only two symbols.

  To define δ', let us first define some homomorphism $f : \Gamma \to 0, 1*$ encoding, where every symbol in Γ gets a unique encoding. For example, we may number the states, and have the $f(q)$ simply be the binary representation of q's number (state $1 \to 0^{k-1}1$, etc.). The encoding will be $k = (log_2 n)$ rounded-up symbols long. Note that Z_0' must be in 0,1.

  Now, for every state $q \in Q$, letter $a \in \Sigma$, and symbol $X \in \Gamma$, for every transition $\delta(q, a, X) = (p, \gamma)$, assuming $f(X) = X_1 \dots X_k$ where $X_i \in 0, 1$, we add to Q', $k - 1$ states. We are essentially adding one state and transitions between them to pop each X_i off the stack, popping off all the X_i which encode X one digit at a time, and then finally pushing γ onto the stack in a final transition to p. For all $1 \le i \le k-2$, we add the transitions

  $$\delta(q, a, X_1) = (q_1, \epsilon)$$
  $$\delta(q_i, \epsilon, X_{i+1}) = (q_{i+1}, \epsilon)$$
  $$\delta(q_{k-1}, \epsilon, X_k) = (p, f(X))$$

  We also add the start state q_0', and the transition $\delta(q_0', \epsilon, Z_0') = (q_0, f(Z_0))$.

  We have now created a PDA $P2$ which has a stack alphabet of two characters and which manipulates the stack in a way equivalent to P.

- Exercise 0.3 *Use the CFL pumping lemma to show each of these languages not to be context-free.*

  1. $\{a^i b^j c^k : i < j < k\}$

     For some number n, let us pick a string in L, $z = a^m b^{m+1} c^{m+2}$ where m is larger than n and $z = uvwxy$, $|vwz| \le n$ and $vx \ne \epsilon$. By the pumping lemma, we can pick any i and uv^iwx^iy will be in L.

     We don't know where in the string v and w are, but it must be one of several cases
     1) v and x are entirely in the a section of z
     2) v and x are entirely in the b section of z
     3) v and x are entirely in the c section of z
     4) v and x are partially in the a section of z, and partially in the b section
     5) v and x are partially in the b section of z, and partially in the c section

     Since $|vwz| \le n < m$, we know that v and x cannot span all three sections.

     In case 1), we can pick $i = m + 1$, adding at least one a. So the number of a's is no longer less than b's and $z' \notin L$.
     In cases 2) and 4), we can pick $i = m + 1$, adding at least one b. So the number of b's is no longer less than c's and $z' \notin L$.
     In case 3), we can pick $i = 0$, subtracting at least one c. So number of b's is no longer less than c's and $z' \notin L$.
     In case 5), we can pick $i = 0$, subtracting at least one b, so number of a's is no longer less than b's and $z' \notin L$.

     Now that we shown that for all cases, we can pick an i such that $z' \notin L$, by the pumping lemma, L is not a context free language.

2. $L = \{a^n b^n c^i : i \leq n < k\}$

   For some number n, let us pick a string in L, $z = a^{k-1}b^{k-1}c^{k-1}$ where $z = uvwxy$, $|vwz| \leq n$ and $vx \neq \epsilon$. By the pumping lemma, we can pick any p and uv^pwx^py will be in L.

   We don't know where in the string v and w are, but it must be one of several cases
   1) v and x are entirely in the a section of z
   2) v and x are entirely in the b section of z
   3) v and x are entirely in the c section of z
   4) v and x are partially in the a section of z, and partially in the b section
   5) v and x are partially in the b section of z, and partially in the c section

   Since $|vwz| \leq n < m$, we know that v and x cannot span all three sections.

   In case 1), we can pick $p = m + 1$, adding at least one a. So the number of a's is now greater than k and $z' \notin L$.
   In case 2), we can pick $p = m + 1$, adding at least one b. So the number of b's is now greater than k and $z' \notin L$.
   In case 3), we can pick $p = m + 1$, adding at least one c. So the number of c's is now greater than k and $z' \notin L$.

   In cases 4), we can pick $p = m + 1$, which will add at least a and one b. So number of a's and number of b's are both no longer less than k's and $z' \notin L$.
   In cases 5), we can pick $p = m + 1$, which will add at least b and one c. So number of b's and number of c's are both no longer less than k's and $z' \notin L$.

   Now that we shown that for all cases, we can pick an i (the same i!) such that $z' \notin L$, by the pumping lemma, L is not a context free language.

3. $L = \{0^p : p\ is\ prime\}$

   For some number n, let us pick a string in L, $z = 0^m$ where $m > n$ and is prime, and $z = uvwxy$, $|vwz| \leq n$ and $vx \neq \epsilon$. By the pumping lemma, we can pick any i and uv^iwx^iy will be in L.

   If we pick $i = m+1$, we get $z' = uv^{m+1}wx^{m+1}y$. We know that the length of z' is $m+(|v|\cdot m)+(|x|\cdot m) = m + m \cdot (|v| + |x|)$. Since we are adding a multiple of our prime m to the length of z', the length of z' is not prime. Therefore, by the pumping lemma, L is not a context free language.

- Exercise 0.4 *Give an algorithm to decide the following: Given a CFG G and one of its variables A, is there any sentential form in which A is the first symbol. Note: Remember that it is possible for A to appear first in the middle of some sentential form but then for all the symbols to its left to derive ϵ.*

  Given $G = \{V, T, P, S\}$, we can build a directed graph $G = \{V, E\}$ where the set of vertices is the set of variables in G. Any two edges v and v' are connected, that is $(v, v') \in E$, iff v appears in the leftmost position of a production rule $\Rightarrow$ v' appears in the leftmost position of a production rule.

  For all production rules in G, $X \to \beta_1 \ldots \beta_k$ where $\beta_i \in \{V \cup R \cup \epsilon\}$, we perform the following steps:

  1. If $\beta_1 \in T$, then that production does not have a variable at its leftmost. Otherwise, add (X, β_1) to E.
  2. For each β_i, add (X, β_i) to E if
     a) $\beta_i \notin T$
     b) for all β_j where $j < i$, $\beta_j \notin T$ and β_j is nullable.

  Once we have have finished building our graph using the above method, we can perform a graph search to see if S and A are connected. If they are, this means that $S \Rightarrow^* A\beta_1 \ldots \beta_k$, there is some setential form with A in the left-most position.