Dept. of Computer Science & Engineering
York University

Assignment 3 — Solutions

Total marks: 60.

Out: July 15
Due: July 25 by 18:00 in the dropbox.

Note that:

- The assignment can be handwritten or typed. It **must** be legible.
- You must do this assignment individually.
- Submit this assignment only if you have read and understood the policy on academic honesty on the course web page. If you have questions or concerns, please contact the instructor.
- Use the dropbox near the main office to submit your assignments, or hand them in at the beginning of class (please note the times and day above). No late submissions will be accepted.

1. [15 points] Let $L_1 = \{0^n 1^m \mid 0 < n \leq m < 2n\}$. Let G_1 be the grammar with starting symbol S and the following rules:

   Rule 1: $S \to 0S11$
   Rule 2: $S \to T$
   Rule 3: $T \to 0T1$
   Rule 4: $T \to 01$

   The goal of this question is to prove that G_1 generates the language L_1.

   a) For each n and m satisfying $0 < n \leq m < 2n$, describe a leftmost derivation of $0^n 1^m$ using the grammar G_1. (That is, say how many times to apply each rule and in what order.)

   Solution:
   First we appply Rule 1 $(m - n)$ times; this produces $0^{m-n}S1^{2(m-n)}$.
   Secondly, we apply Rule 2 once; this produces $0^{m-n}T1^{2(m-n)}$.
   Thirdly, we apply Rule 3 $(2n - m - 1)$ times; this produces $0^{n-1}T1^{m-1}$.
   Fourthly (and finally), we apply Rule 4 once; this produces $0^n 1^m$.

b) Give a formal proof that every string that can be generated from T is of the form 0^k1^k for some positive integer k.

Solution:
By strong induction on the length of strings generated from T.
The statement of the induction hypothesis (IH) is that if a string is generated from T, then it must be of the form 0^k1^k for some positive integer k.
Base case: For $|s| = 2$ (T does not generate any shorter strings). Then $s = 01$, and this is of the form 0^k1^k for $k = 1$.
Induction step. Assume that the IH holds for all strings generated by T of length n or less. Consider a string s of length $n+1$ produced by T. Then s must have been obtained using Rule 3 and thus $s = 0w1$ where w is a string of length $n-1$ generated by T. By the IH, w is of the form 0^k1^k for some positive integer k. Thus s is of the form 0^i1^i for some positive integer $i = k+1$. Therefore every string that can be generated from T is of the form 0^k1^k for some positive integer k.

c) Give a formal proof that every string generated by G_1 is in L_1.

Solution:
By strong induction on the length of strings generated by G_1.
The statement of the induction hypothesis (IH) is that if a string is generated by G_1, then it must be in L_1.
Base case: For $|s| = 2$ (G_1 does not generate any shorter strings). Then $s = 01$, and this is in L_1.
Induction step. Assume that the IH holds for all strings generated by G_1 of length n or less. Consider a string s of length $n+1$ produced by G_1.
If s was obtained by Rule 2, then s is produced by T. Then by part b), s is of the form 0^k1^k for some positive integer k. Thus s is in L_1.
If s was obtained by Rule 1, then $s = 0w11$ where w is a string of length $n-2$ generated by G_1. By the IH, w is in L_1 and so is of the form 0^n1^m where $0 < n \leq m < 2n$. Thus s is of the form $0^{n+1}1^{m+2}$ where $0 < n \leq m < 2n$. This means that s is of the form 0^i1^j where $1 < i < j < 2i$ (since $m < 2n$, $m+2 < 2(n+1)$, and thus $j < 2i$). Thus s is in L_1.
There are no other ways of generating strings in G_1. Therefore every string that can be generated by G_1 is in L_1.

2. [10 points] Let $L_2 = \{a^ib^jc^k \mid i,j,k \geq 0, i = j \text{ or } j = k\}$. Describe a pushdown automaton that recognizes L_2 by giving its state diagram.

Solution:
One solution is a PDA that is very similar to the one in Figure 2.17 in Sipser, but where we replace the branch that goes to q_5, q_6 and q_7 by an alternative branch to

accept $a^n b^m c^m$. To do this we delete q_5, q_6 and q_7 and any transitions involving them and add:
a transition $\epsilon, \epsilon \to \epsilon$ from q_1 to a new state q_8,
from q_8 we add a transition $a, \epsilon \to \epsilon$ to q_8 and a transition $\epsilon, \epsilon \to \epsilon$ to a new state q_9,
from q_9 we add a transition $b, \epsilon \to b$ to q_9 and a transition $\epsilon, \epsilon \to \epsilon$ to a new state q_{10},
from q_{10} we add a transition $c, b \to \epsilon$ to q_{10} and a transition $\epsilon, \$ \to \epsilon$ to a new accept state q_{11} (q_{11} has no outgoing transitions).

3. [10 points] Let $L_3 = \{a^n b^m c^n \mid n \geq m\}$. Prove that L_3 is not a context free language.

   Solution:
   By contradiction. Assume that L_3 is a CFL. By the pumping lemma for CFLs, there is a pumping length p where if s is any string in L_3 of length at least p, then $s = uvxyz$ and for all $i \geq 0$, $uv^ixy^iz \in L_3$, $|vy| > 0$, and $|vxy| \leq p$.
   Let $s = a^p b^p c^p$.
   Neither v nor y can cross a and b regions because if one of then did, then when we pump up, we would get as and bs out of order. So we only need to consider the cases where each is in one of the three regions of s:
   If $v = a^i$ and $y = a^j$ and we pump up we will get more as than cs.
   The case $v = c^i$ and $y = c^j$ is similar.
   If $v = b^i$ and $y = b^j$, then if we keep pumping up we will get more bs than as.
   If $v = a^i$ and $y = b^j$, and we pump up we will either get more as than cs (if $v \neq \epsilon$) or eventually get get more bs than cs (if $y \neq \epsilon$), or both.
   The case $v = b^i$ and $y = c^j$ is similar.
   The case $v = a^i$ and $y = c^j$ is ruled out since $|vxy| \leq p$.
   Thus is all cases we get a contradiction, and therefor L_3 is not context free.

4. [15 points] We have seen in class that $L = \{ww \mid w \in \{0,1\}^*\}$ is not a context free language. However the complement of L, $\overline{L} = (\{0,1\}^* \setminus L)$ is a context free language. Give a context free grammar G that generates $\overline{L}$ and explain its design (you need not give a formal proof that it generates $\overline{L}$).

   Solution:
$$\begin{aligned}
&S \to O \mid E \\
&O \to 0 \mid 1 \mid 0O0 \mid 0O1 \mid 1O0 \mid 1O1 \\
&E \to F \mid 0F0 \mid 0F1 \mid 1F0 \mid 1F1 \\
&F \to 01 \mid 0U0 \mid 0U1 \mid 10 \mid 1Z0 \mid 1Z1 \\
&U \to 01 \mid 11 \mid 0U0 \mid 0U1 \mid 1U0 \mid 1U1 \\
&Z \to 00 \mid 10 \mid 0Z0 \mid 0Z1 \mid 1Z0 \mid 1Z1
\end{aligned}$$

   O generates all odd length strings. E generates all even length strings that are not in L, i.e., differ on at least one character in the first and second half of the string. F generates all even length strings that differ on at least the first character in the first and second half of the string. U generates all even length strings such that the second half starts with a 1. Z generates all even length strings such that the second half starts with a 0.

5. [10 points] Let $L_5 = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$. Describe a Turing Machine that decides L_5. While you need not draw a state diagram, you should describe the machine in detail.

   Solution:
   1) If the input is the empty string accept. Otherwise, shift the input rightwards by 1 cell and mark the left end of the tape with $. While doing this, we can check whether the input is of the form $a^i b^j c^k$ for $i, j, k \geq 0$ and reject if it is not. At the end we return to the start of the tape. (Optionally, this can be described in more detail, but the description becomes very tedious.)
   2) Now we make sure that $i \leq j$ and $i \leq k$:
   While there is an a on the tape (scan right to find one), find b afterwards and cross it and find a c afterwards and cross it. If there is no b or no c reject. At the end of each iteration return to the start of the tape.
   3) Now we make sure that $j \leq k$:
   While there is an b on the tape (scan right to find one), find a c afterwards and cross it. If there is no c reject. At the end of each iteration return to the start of the tape.
   4) If we complete all this without rejecting, then we accept.