George Washington University
Department of Computer Science

CS160-10
Theory of Computer Translators
Spring 2008

Solution to Exam 1
2/1/2008

1. (10 pts) Consider the following Grammar G:

S → Xy | YZ
X → (X) | x
Y → yYz | ϵ
Z → zXz | Y

(a) Construct FIRST and FOLLOW sets of each non-terminal symbol.

	First	Follow
S	$(, x, y, z, \epsilon$	\$
X	$(, x$	$y, z,)$
Y	y, ϵ	$y, z,$ \$
Z	y, z, ϵ	\$

(b) Fill in the $LL(1)$ parsing table. Do not modify the grammar. If there are conflicts, just write both entries in the table.

	Input symbol					
	(	)	x	y	z	\$
S	$S \to Xy$		$S \to Xy$	$S \to YZ$	$S \to YZ$	$S \to YZ$
X	$X \to (x)$		$X \to x$			
Y				$Y \to yYz, Y \to \epsilon$	$Y \to \epsilon$	$Y \to \epsilon$
Z				$Z \to y$	$Z \to zXz$	$Z \to Y$

2. (10 pts.) Consider the following grammar G.

S → aS | Ab
A → XYZ | ϵ
X → cS | ϵ
Y → dS | ϵ
Z → eS

(a) Give a left-most derivation of the string *aebb*.

$S \Rightarrow aS \Rightarrow aAb \Rightarrow aXYZb \Rightarrow aYZb \Rightarrow aZb \Rightarrow aeSb \Rightarrow aeAbb \Rightarrow aebb$

(b) Explain why it is that if we add the production $X \rightarrow bS$, the grammar is no longer $LL(1)$. Do NOT try to construct a $LL(1)$ parsing table for the new grammar.

Note: We would then have $b \in First(A)$ making the table entry $M[A, b]$ filled with "$A \rightarrow XYZ$". As $A \stackrel{*}{\Rightarrow} \epsilon$ and $b \in Follow(A)$, the table entry $M[A, b]$ will also have to be filled with "$A \rightarrow \epsilon$". Therefore, the new grammar cannot be LL(1).

3. (5 pts) Consider a grammar G which has two production rules: A $\rightarrow \alpha \mid \beta$. Suppose α can derive the empty string and β can derive a string bβ' such that b is in FOLLOW(A). Can G be an LL(1) grammar? Explain clearly.

Note: Suppose $\beta \stackrel{*}{\Rightarrow} \epsilon$ and $\alpha \stackrel{*}{\Rightarrow} a\alpha'$, where $a \in FOLLOW(A)$. Also, assume that $\gamma_2 \stackrel{*}{\Rightarrow} a\gamma_2'$. We then have two possibilities: (i) $S \stackrel{*}{\Rightarrow} \gamma_1 A\gamma_2 \stackrel{*}{\Rightarrow} \gamma_1\alpha\gamma_2 \stackrel{*}{\Rightarrow} \gamma_1 a\alpha'\gamma_2$, and (ii) $S \stackrel{*}{\Rightarrow} \gamma_1 A\gamma_2 \stackrel{*}{\Rightarrow} \gamma_1\beta\gamma_2 \stackrel{*}{\Rightarrow} \gamma_1\gamma_2 \stackrel{*}{\Rightarrow} \gamma_1 a\gamma_2'$. Hence, after taking care of the input tokens corresponding to γ_1, the parser cannot make a clear choice between the two productions $A \rightarrow \alpha$ and $A \rightarrow \beta$.