Computability 2012
Professor Sheryl Shulman
Formal Languages
Assignment 5
Owen Meyer

1.) A CFG over $\{a,b\}$ that generates a language consisting of all the strings with twice as many a's as b's.

$$L = (bba \mid bab \mid abb)^*$$

$$G: S \to B \mid \lambda$$
$$B \to bbA \mid bAb \mid Abb$$
$$A \to aB \mid a$$

Let $u = \overset{n}{\underset{G}{\Rightarrow}} u$; allowing fo sentential form, any string u derivable from G has the property $2n_a(u) = n_b(u)$ (P)

where $n_a(u)$ = number of a's in u (+1 if there is an A)

$n_b(u)$ = number of b's in u

show P by induction on the length of the derivation of u.

Base case:

$S \to \lambda$, $S \to B$ where $n_a(u) = 0$, $n_b(u) = 0$ ✓

For the inductive case, Let:

$$\overset{n}{\underset{G}{\Rightarrow}} u \underset{G}{\Rightarrow} w$$

---

assume $2n_a(u) = n_b(u)$; show $2n_a(w) = n_b(w)$

Let x_1 = some string of 0 or more terminals $\{a,b\}$

x_2 " "

$$n_a(w) = n_a(u) + 1$$
$$2n_a(w) = 2(n_a(u) + 1)$$
$$= 2n_a(u) + 2$$
$$= n_b(u) + 2$$
$$= n_b(w)$$

$$2n_a(w) = n_b(w)$$ ✓

	x_1Bx_2	$n_a(w)$	$n_b(w)$	x_1Ax_2	$n_a(w)$	$n_b(w)$
$A \to aB$		NC	NC	x_1aBx_2	$n_a(u)$	$n_b(u)$
$A \to a$		NC	NC	x_1ax_2	$n_a(u)$	$n_b(u)$
$B \to bbA$	x_1bbAx_2	$n_a(u)+1$	$n_b(u)+2$		NC	NC
$B \to bAb$	x_1bAbx_2	$n_a(u)+1$	$n_b(u)+2$		NC	NC
$B \to Abb$	x_1Abbx_2	$n_a(u)+1$	$n_b(u)+2$		NC	NC

...cont

cont...

Now, using a derivation schema, show that any string that can be described by L can be derived by G.

A string describable by L has the form:

$a^{n_1}b^{m_1}a^{n_3}b^{m_3}\ldots a^{n_{k-1}}b^{m_{k-1}}a^{n_k}b^{m_k}a^{n_{k-2}}b^{m_{k-2}}\ldots a^{n_2}b^{m_2}a^{n_0}b^{m_0}$ where $2\sum_{i=0}^{k} n_i = \sum_{i=0}^{k} m_i$

a schema to derive any such string from G:

$S \Rightarrow B$ $\qquad S \to B$

$\overset{k}{\Rightarrow} a^{n_1}b^{m_1}a^{n_3}b^{m_3}\ldots a^{n_{k-1}}b^{m_{k-1}}A\,b^{m_k}a^{n_{k-2}}b^{m_{k-2}}\ldots a^{n_2}b^{m_2}a^{n_0}b^{m_0}$

from k application of $B \to bbA \mid bAb \mid Abb$ followed by $A \to aB$

$\Rightarrow a^{n_1}b^{m_1}a^{n_3}b^{m_3}\ldots a^{n_{k-1}}b^{m_{k-1}}a^{n_k}b^{m_k}a^{n_{k-2}}b^{m_{k-2}}\ldots a^{n_2}b^{m_2}a^{n_0}b^{m_0}$

9.) $\{a,b,c\}$: $\{a^i b^m c^i \mid 0 \leq n+m \leq i\}$

$S \to \lambda \mid aSc \mid B$

$B \to bBc \mid bc \mid C$

$C \to Cc \mid \lambda$

11.) $\{a,b\}$: $\{a^m b^i a^n \mid i = m+n\}$

$S \to BC$

$B \to aBb \mid \lambda$

$C \to bCa \mid \lambda$

37.

$L_1 = \{a^n b^n c^m \mid n,m > 0\}$ $L_2 = \{a^n b^m c^m \mid n,m > 0\}$

$S \rightarrow Sc \mid B$
$B \rightarrow aBb \mid ab$

$S \rightarrow aS \mid B$
$B \rightarrow bBc \mid bc$

$L_1 \cup L_2$

$S \rightarrow Sc \mid B \mid aS \mid C$
$B \rightarrow aBb \mid ab$
$C \rightarrow bCc \mid bc$

$S \Rightarrow Sc$
$\Rightarrow Bc$
$\Rightarrow abc$

$S \Rightarrow aS$
$\Rightarrow aC$
$\Rightarrow abc$

This grammar must be ambiguous because there is no way to make a single characterization of $L_1 \cup L_2$ in set notation. The only way to design a grammar that generates this union is to have a branching point where either L_1 or L_2 can be entered from a common starting point where there is no way back to this point. However once either L_1 or L_2 has been entered from this union grammar, the sets they derive do still intersect so some strings can be derived from either the L_1 side of the grammar or the L_2 side.