Solutions Exam 150603

Here we only give a brief explanation of the solution. Your solution should in general be more elaborated than these ones.

1. The property to prove is $P(n)$: For all w, of $S \Rightarrow^n w$ then $\#_a(w) = \#_b(w) \leqslant 2^{n-1}$.
   We will use course of value induction on the length of the derivation (number of steps) of $S \Rightarrow^n w$.

   Base case: $S \Rightarrow w$ in one step ($n = 1$), hence the rule applied should have been $S \to ab$ or $S \to ba$.

   Then $w = ab$ or $w = ba$ and hence $\#_a(w) = \#_b(w) = 1 \leqslant 2^{1-1} = 1$.

   Step case: IH: if $S \Rightarrow^m w'$ in $0 < m \leqslant n$ steps then $\#_a(w') = \#_b(w') \leqslant 2^{m-1}$.

   Let $S \Rightarrow^{n+1} w$ with $n > 0$.

   Since $n > 0$ then w is derived in at least 2 steps and then the first rule applied should have been $S \to SS$. The derivation will then look like $S \Rightarrow SS \Rightarrow^n w$.

   So we have that $w = w_1 w_2$ with $S \Rightarrow^i w_1$ and $S \Rightarrow^j w_2$, $0 < i, j < n$ and $i + j = n$.

   By IH we should then have that $\#_a(w_1) = \#_b(w_1) \leqslant 2^i$ and $\#_a(w_2) = \#_b(w_2) \leqslant 2^j$.

   We have then that $\#_a(w) = \#_a(w_1) + \#_a(w_2) = \#_b(w_1) + \#_b(w_2) = \#_b(w)$.

   In addition $\#_a(w) = \#_b(w) = \#_a(w_1) + \#_a(w_2) \leqslant 2^i + 2^j = 2^{i+j} = 2^n \leqslant 2^{n+1}$.

2.

	0	1
$\to q_0$	q_1	q
$^*q_1$	q_1	q_2
q_2	q_3	q_2
$^*q_3$	q_1	q
q	q	q

3. (a)

	a	b	c
$\to q_0q_2$	q_1q_2	q_3q_4	$-$
q_1q_2	q_2	$q_1q_3q_4$	q_4
q_2	q_2	q_3q_4	$-$
$^*q_1q_3q_4$	$-$	q_1	q_3q_4
q_1	$-$	q_1	q_4
$^*q_4$	$-$	$-$	$-$
$^*q_3q_4$	$-$	$-$	q_3q_4

   (b) $aaa^*bc^* + ab(\epsilon + bb^*c + cc^*) + ac + bc^*$

4. Each expression is valid 1.5pts.

   $E_1 = (b + c)^*(ac^*b(b + c)^*)^*$: In E_1 any number of b's or c's could occur; as soon as an a comes then it could be followed by some c's but eventually a b must occur, after which any number of b's or c's could follow. This last part could be repeated many times.

   A simpler expression could be $(b + c + ac^*b)^*$.

   $E_2 = (a + c)^*(bc^*a(a + c)^*)^*$: The explanation of E_2 is similar to that of E_1.

   A simpler expression could be $(a + c + bc^*a)^*$.

$E = c^*$: If an a occurs in E it must be followed by a b which in turn should be followed by another a and in turned followed by another b and then the would should be infinite. So no a can occur in E. For similar reasons no b can occur in E, so only c's can occur in E.

5. First we run the algorithm that identifies equivalent states.

	q_0	q_1	q_2	q_3	q_4	q_5
q_6	X	X	X	X	X	
q_5	X	X	X	X	X	
q_4	X	X	X			
q_3	X	X	X			
q_2	X	X				
q_1	X					

After the base case in the algorithm we can distinguish (q_0, q_6), (q_1, q_6), (q_2, q_6), (q_3, q_6), (q_4, q_6) and (q_0, q_5), (q_1, q_5), (q_2, q_65), (q_3, q_5), (q_4, q_5).

After this, a can distinguish (q_0, q_4), (q_1, q_4), (q_2, q_4), (q_0, q_3), (q_1, q_3), (q_2, q_3), (q_0, q_2), (q_0, q_1), and (q_1, q_2), in that order, since it takes them to already distinguishable pairs.

The resulting automaton is:

	a	b
$\rightarrow q_0$	q_3q_4	q_5q_6
q_1	q_0	q_5q_6
q_2	q_1	q_3q_4
q_3q_4	q_5q_6	q_3q_4
${}^*q_5q_6$	q_2	q_5q_6

6. (a) See slide 8 lecture 8.

   (b) Let us assume the language is regular. Then the PL must apply.
   Let n be the constant given by the lemma. Let us consider the word $0^n1^{n+1}2^{n+2}$ which satisfies the conditions in the lemma.
   We know now that the word can be divided in 3 parts xyz and that $|xy| \leqslant n$ so xy should contain only 0's.
   Since $y \neq \epsilon$ then y will contain at least one 0. Hence if we pump y $k > 1$ times we will have more 0's than 1's and then the resulting word will not belong to the language.
   This contradicts the PL so we deduce that the language is not regular.

7. (a) First of all observe that for $n < m < 2n$ to be a valid expression then $n > 1$.

$$S \rightarrow 0S11 \mid 0A1 \qquad A \rightarrow 0A1 \mid 011$$

   (b) When constructing a word we first use the rule $S \rightarrow 0S11$ $i \geqslant 0$ times. This will put i 0's and $2i$ 1's in the word.
   Then we must use the rule $S \rightarrow 0A1$ and thereafter we use the rule $A \rightarrow 0A1$ $j \geqslant 0$ times. This will put $j + 1$ 0's and 1's in the word.
   Finally we finished with 011 adding a 0 and 2 1's in the word.
   So in total we put $n = i+j+2$ 0's and $m = 2i+j+3$ 1's. It is clear that $i+j+2 < 2i+j+3 < 2i + 2j + 4$.

   (c) No, the grammar is not ambiguous.
   There is only 1 recursive production from each of the variables, so the only way this grammar would be ambiguous is if there are 2 possible divisions of n into $i + j + 2$ and $i' + j' + 2$ so that the nr of 1's would be the same, that is $2i + j + 3 = 2i' + j' + 3$.
   But if $i+j = i'+j'$ and $2i+j = 2i'+j'$ hold then $j = i'+j'-i$ and then $2i-i+i'+j' = 2i'+j'$ must hold, which means $i + i' = 2i'$ so $i = i'$ and hence $j = j'$.

(d) The rightmost/leftmost derivation is $S \Rightarrow 0S11 \Rightarrow 00A111 \Rightarrow 000A1111 \Rightarrow 0000111111$.

(e)

$$\begin{array}{ll} S \to ZF \mid ZB & Z \to 0 \\ A \to ZB \mid ZT & O \to 1 \\ F \to ST & T \to OO \\ B \to AO & \end{array}$$

8. (a) i. (2pts) Nullable variables: A, B, H, G.

$$\begin{array}{ll} S \to 0AS1 \mid 0SB1 \mid 0S1 \mid C & \\ A \to 0A \mid 0 & B \to 1B \mid 1 \\ C \to 2C \mid D & D \to 3 \mid 3D \mid 3E \\ E \to 4E \mid F & F \to 4F \mid E \\ G \to 5G \mid 5 \mid H & H \to 5H \mid 5 \end{array}$$

ii. (2pts) Unit productions: $S \to C, C \to D, E \to F, F \to E, G \to H$.

$$\begin{array}{ll} S \to 0AS1 \mid 0SB1 \mid 0S1 \mid 2C \mid 3 \mid 3D \mid 3E & \\ A \to 0A \mid 0 & B \to 1B \mid 1 \\ C \to 2C \mid 3 \mid 3D \mid 3E & D \to 3 \mid 3D \mid 3E \\ E \to 4E \mid 4F & F \to 4F \mid 4E \\ G \to 5G \mid 5 \mid 5H & H \to 5H \mid 5 \end{array}$$

iii. (1.5pts) Non-generating symbols: E, F

$$\begin{array}{ll} S \to 0AS1 \mid 0SB1 \mid 0S1 \mid 2C \mid 3 \mid 3D & \\ A \to 0A \mid 0 & B \to 1B \mid 1 \\ C \to 2C \mid 3 \mid 3D & D \to 3 \mid 3D \\ G \to 5G \mid 5 \mid 5H & H \to 5H \mid 5 \end{array}$$

iv. (1.5pts) Non-reachable symbols: G, H

$$\begin{array}{ll} S \to 0AS1 \mid 0SB1 \mid 0S1 \mid 2C \mid 3 \mid 3D & \\ A \to 0A \mid 0 & B \to 1B \mid 1 \\ C \to 2C \mid 3 \mid 3D & D \to 3 \mid 3D \end{array}$$

(b) A generates 0^+, B generates 1^+, D generates 3^+ and C generates 2^+3^+.
S generates $0^+2^+3^+1^+ + 2^+3^+$.

9.

$$\begin{array}{|cccccc} \{S,A,B\} & & & & & \\ \{A,B\} & \{S,A,B\} & & & & \\ \{A,B\} & \{S,A\} & \{A,B\} & & & \\ \{A,B\} & \{S,A\} & \{B\} & \{S,B\} & & \\ \{B\} & \{S,A\} & \{B\} & \emptyset & \{S,A\} & \\ \{B\} & \{A\} & \{B\} & \{A\} & \{A\} & \{B\} \\ \hline b & a & b & a & a & b \end{array}$$

Since S belongs to the upper set then the string is generated by the grammar.

10. (a) Observe that the word 1 should belong to the language.
Let $\Sigma = \{0, 1\}$ and $a \in \{0, 1\}$.
Let $M = (\{q_0, \ldots, q_4, q_f\}, \Sigma, \delta, q_0, \Box, \{q_f\})$, and δ is as follows:

$\delta(q_0, 1) = (q_1, 1, \mathsf{R})$	The word starts with 1;
$\delta(q_1, \Box) = (q_f, \Box, \mathsf{R})$	The word is only 1 and should be accepted;
$\delta(q_1, a) = (q_2, a, \mathsf{R})$	We have read an even nr of symbols;
$\delta(q_2, a) = (q_3, a, \mathsf{R})$	We have read an odd nr of symbols;
$\delta(q_3, a) = (q_2, a, \mathsf{R})$	If we read another 0 or 1 we have now read an even nr of symbols;
$\delta(q_3, \Box,) = (q_4, \Box, \mathsf{L})$	If the word finished we need to check that the last symbols was a 1;
$\delta(q_4, 1) = (q_f, 1, \mathsf{R})$	If the last symbol was a 1 we accept.

(b) Yes, it will reach q_f when the word needs to be accepted and it will halt if the word should not be accepted.