Solutions Exam 180822

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

1. Our property is: $P(n) : \forall w$, if $S \Rightarrow^n w$ then $\exists i, n \leqslant i \leqslant 2n$ such that $w = a^n b^i$.

   We will use mathematical/simple induction on the length of the derivation (number of steps) $S \Rightarrow^n w$.

   Base cases: $S \Rightarrow w$, hence the rule applied should have been $S \to ab$ or $S \to abb$.

   For $S \to ab$ we have that $w = ab$ and hence $i = 1$ which is such that $1 \leqslant 1 \leqslant 2$ as desired.

   For $S \to abb$ we have that $w = abb$ and hence $i = 2$ which is such that $1 \leqslant 2 \leqslant 2$ as desired.

   Step case: Our IH is: $\forall w$, if $S \Rightarrow^n w$ then $\exists i, n \leqslant i \leqslant 2n$ such that $w = a^n b^i$.

   Let $S \Rightarrow^{n+1} w$ with $n > 0$. We need to prove $P(n+1)$, that is, $\forall w$, if $S \Rightarrow^{n+1} w$ then $\exists i, n+1 \leqslant i \leqslant 2n+2$ such that $w = a^{n+1} b^i$.

   Since $n > 0$ then the first rule applied should have been $S \to aSb$ or $S \to aSbb$.

   In the case the first rule was $S \to aSb$ then we have that $S \Rightarrow aSb \Rightarrow^n w$ so $w = aw_1b$ with $S \Rightarrow^n w_1$. Then the IH applies to n and we know that $\exists i, n \leqslant i \leqslant 2n$ such that $w_1 = a^n b^i$. Here $\exists (i+1)$ so $w = aa^n b^i b = a^{n+1} b^{i+1}$. We also have that $n+1 \leqslant i+1 \leqslant 2n+1 \leqslant 2n+2$ as desired.

   In the case the first rule was $S \to aSbb$ then we have that $S \Rightarrow aSbb \Rightarrow^n w$ so $w = aw_1bb$ with $S \Rightarrow^n w_1$. Then the IH applies to n and we know that $\exists i, n \leqslant i \leqslant 2n$ such that $w_1 = a^n b^i$. Here $\exists (i+2)$ so $w = aa^n b^i bb = a^{n+1} b^{i+2}$. We also have that $n+1 \leqslant i+1 \leqslant 2n+1$ and hence $n+1 \leqslant i+2 \leqslant 2n+2$ as desired.

2. We define a DFA:

	0	1	2
$\to q_0$	q_0	q_1	q_2
q_1	q_0	q_1	$-$
$^*q_2$	q_2	q_3	q_0
$^*q_3$	q_2	q_3	$-$

3.

	0	1	2
$\to q_0$	q_0q_2	q_1	q_2
q_1	q_1	q_2q_3	q_3
$^*q_2$	q_2	q_1	q_3q_4
$^*q_0q_2$	q_0q_2	q_1	$q_2q_3q_4$
$^*q_2q_3$	q_2q_4	q_1	q_3q_4
$^*q_3$	q_4	$-$	q_4
q_4	q_3	$-$	q_3
$^*q_2q_4$	q_2q_3	q_1	q_3q_4
$^*q_3q_4$	q_3q_4	$-$	q_3q_4
$^*q_2q_3q_4$	$q_2q_3q_4$	q_1	q_3q_4

4. I will solve equations:

$$\begin{aligned}
&E_0 = 0E_0 + 1E_1 + 2E_2 = 0E_0 + 2E_2 + 11E_3 + 12E_4 = 0E_0 + 2E_2 + (112 + 12)E_4\\
&E_1 = 1E_3 + 2E_4\\
&E_2 = 0E_1 + 1E_2 = 1E_2 + 01E_3 + 02E_4 = 1E_2 + (012 + 02)E_4\\
&E_3 = 2E_4\\
&E_4 = 0E_2 + 1E_3 + \epsilon = 0E_2 + 12E_4 + \epsilon
\end{aligned}$$

So $E_2 = 1^*(012+02)E_4$ and

$$\begin{aligned} E_0 &= 0E_0 + (21^*(012+02)+112+12)E_4 \\ E_4 &= (01^*(012+02)+12)E_4 + \epsilon = (01^*(012+02)+12)^* \end{aligned}$$

Hence

$$E_0 = 0E_0 + (21^*(012+02)+112+12)(01^*(012+02)+12)^* = 0^*(21^*(012+02)+112+12)(01^*(012+02)+12)^*$$

5. (a) See slide 4 lecture 8.
   (b) See slide 7 lecture 8.
   (c) Observe that the words 11 cannot belong to the language since it starts and ends with 11 AND has 11 as substring. On the other hand, just 1 should be allowed. Hence we have

$$0(0+1)^*11(0+1)^*0 + 1(0+01)^*01 + 1$$

6. (a) See slide 17 lecture 9.
   (b) Let us assume our language $\mathcal{L}$ is regular. Hence the PL should apply.
   Let n be the constant given by the PL.
   Let $w = 0^n1^{n+1}2^{n+2}$. We have that $w \in \mathcal{L}$ and that $|w| \geqslant n$.
   Hence $w = xyz$ with $y \neq \epsilon$ and $|xy| \leqslant n$.
   Then y will contain only 0's and will contain at least a 0.
   Let us consider the word $xyyz$, that is, the case for $k = 2$.
   We have that $\#_0(xyyz) \geq n+1$ but $\#_1(xyyz) = \#_1(xyz) = n+1$ since there are no 1's in y.
   This means that $xyyz \notin \mathcal{L}$, which contradicts the Pumping lemma. Hence $\mathcal{L}$ cannot be regular.

7. One could observe that actually this language is simply $a^*b^*c^*$ so the grammar

$$S \to ABC \qquad A \to aA \mid \epsilon \qquad B \to bB \mid \epsilon \qquad C \to cC \mid \epsilon$$

would generate it.

Clearly this grammar generates words $a^ib^jc^k$ where $i, j, k \geqslant 0$, that is, no other restriction on the i, j, k.

We will now show that these words belong to $\mathcal{L}$, that is, that $i \leqslant j+k \vee j \leqslant i+k$ is satisfied.

We procede by contradiction. Let us assume there a word $a^ib^jc^k$ such that $\neg(i \leqslant j+k \vee j \leqslant i+k)$. Hence the word is such that $\neg(i \leqslant j+k) \wedge \neg(j \leqslant i+k) = i \nleqslant j+k \wedge j \nleqslant i+k = i > j+k \wedge j > i+k$. Thus we have that $i+j > i+j+2k$ and this is not possible since $\geqslant 0$.

Another way to show this is the following. Let us assume we have a word $a^ib^jc^k$ such that $i > j+k$. Then $i+k > j+2k \geqslant j+2k \geqslant j$, that is $i+k \geqslant j$, since $k \geqslant 0$.

Now, if you have not realised this, one could try the "hard" way and generate those word satisfying each condition on a separate way. This is done below.

(a)

$$\begin{array}{ll} S \to P \mid TQ & \\ P \to aPc \mid Pc \mid R & Q \to bQc \mid Qc \mid \epsilon \\ R \to aRb \mid Rb \mid \epsilon & T \to aTb \mid aT \mid \epsilon \end{array}$$

(b) P generates the words where $i \leqslant j+k$ and TQ the words where $j \leqslant i+k$.
P makes sure there is a c at the end for every a we add at the beginning. Extra c's at the end are ok. When we are done adding c's then we move to R in order to add a's and b's in the middle of the word.
R makes sure there is a b to the right for every a we add to the left. Extra b's to the right are ok. When we are done we produce ϵ.

Observe that P can generate the empty word, words with no a's, and words with no b's.

Q makes sure there is a c to the right for every b we add to the left. Extra c's to the right are ok. When we are done adding c's we produce ϵ. Observe we can even produce words with no b's via Q.

T makes sure there is an a to the left for every b we add to the right. Extra a's to the left are ok. When we are done adding a's we produce ϵ. Observe we can even produce words with no b's via T.

Observe TQ could also produce the empty word.

(c) The word $aabbc$ satisfies both $i \leqslant j + k$ and $j \leqslant i + k$ so it can be produced both via P and via TQ, hence the existence of two difference parse trees for the word.
(Here you will need to give the parse trees).

(d) i. ϵ belongs to the language of R because $R \to \epsilon$;
ii. b belongs to the language of R because $R \to Rb$ and i);
iii. abb belongs to the language of R because $R \to aRb$ and ii);
iv. abb belongs to the language of P because $P \to R$ and iii);
v. $aabbc$ belongs to the language of P because $P \to aPc$ and iv);
vi. $aabbc$ belongs to the language of S because $S \to P$ and v).

8. (a) Nullable: A, B, D

$$\begin{array}{llll} S \to ab \mid aAb \mid abB \mid aAbB \mid cCd \mid cCdD \mid eEfF & & & \\ A \to a \mid aA \mid B & B \to b \mid bB & C \to c \mid cC & D \to d \mid dD \mid C \\ E \to eE \mid F & F \to fF \mid Ef & G \to eE \mid hH \mid g & H \to fF \mid gG \mid h \end{array}$$

(b) Unit productions: $A \to B, D \to C, E \to F$

$$\begin{array}{llll} S \to ab \mid aAb \mid abB \mid aAbB \mid cCd \mid cCdD \mid eEfF & & & \\ A \to a \mid aA \mid b \mid bB & B \to b \mid bB & C \to c \mid cC & D \to d \mid dD \mid c \mid cC \\ E \to eE \mid fF \mid Ef & F \to fF \mid Ef & G \to eE \mid hH \mid g & H \to fF \mid gG \mid h \end{array}$$

(c) Non-generating: E, F

$$\begin{array}{lll} S \to ab \mid aAb \mid abB \mid aAbB \mid cCd \mid cCdD & & \\ A \to a \mid aA \mid b \mid bB & B \to b \mid bB & C \to c \mid cC \\ D \to d \mid dD \mid c \mid cC & G \to hH \mid g & H \to gG \mid h \end{array}$$

(d) Non-reachable: g, h, G, H

$$\begin{array}{ll} S \to ab \mid aAb \mid abB \mid aAbB \mid cCd \mid cCdD & \\ A \to a \mid aA \mid b \mid bB & B \to b \mid bB \\ C \to c \mid cC & D \to d \mid dD \mid c \mid cC \end{array}$$

(e) See slide 17, lecture 13.

(f)

$$\begin{array}{llll} S \to RT \mid XT \mid RY \mid XY \mid ZV \mid ZW & & & \\ R \to a & T \to b & U \to c & V \to d \\ X \to RA & Y \to TB & Z \to UC & W \to VD \\ A \to a \mid RA \mid b \mid TB & & B \to b \mid TB & \\ C \to c \mid UC & & D \to d \mid VD \mid c \mid UC & \end{array}$$

9.

$\{S, A\}$				
$\{S, C\}$	$\{B\}$			
$\{S\}$	$\{S\}$	$\{C\}$		
$\{S, B\}$	$\emptyset$	$\{S, A\}$	$\{S\}$	
$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{B, C\}$	$\{B\}$
a	a	b	c	b

S belongs to the upper-most set, which means that the word is generated by the grammar since S is the starting symbol of the grammar.

10. (a) We have three cases to consider: $n = 1 \,\&\, m = 2$, $n = 1 \,\&\, m = 3$ and $n = 2 \,\&\, m = 3$. So the regular expression is: $ab(aa + aaa)c + aabaaac$.

    (b) See slide 26 lecture 15.

    (c) Let $M = (\{q_0, ..., q_{10}, q_f\}, \{a, b, c\}, \delta, q_0, \Box, \{q_f\})$, with δ is as follows:

$\delta(q_0, a) = (q_1, a, R)$	at least an a should come;
$\delta(q_1, a) = (q_2, a, R)$	this is the case $n = 2$;
$\delta(q_1, b) = (q_3, b, R)$	this is the case $n = 1$;
$\delta(q_2, b) = (q_4, b, R)$	we need to look for $m = 3$;
$\delta(q_3, a) = (q_5, a, R)$	we read first a of 2 or 3;
$\delta(q_4, a) = (q_6, a, R)$	we read first a of 3;
$\delta(q_5, a) = (q_7, a, R)$	we read second a of 2 or 3;
$\delta(q_6, a) = (q_8, a, R)$	we read second a of 3;
$\delta(q_7, a) = (q_9, a, R)$	this is case $m = 3$;
$\delta(q_7, c) = (q_{10}, c, R)$	this is case $m = 2$;
$\delta(q_8, a) = (q_9, a, R)$	we read the third a
$\delta(q_9, c) = (q_{10}, c, R)$	this is case $m = 3$;
$\delta(q_{10}, \Box) = (q_f, \Box, R)$	we have a right tape so we accept.

    (d) The Turing decider moves only to the right. It will halt when the sequence of symbols is not correct, or it will end up in the final state when the sequence is correct and needs to be accepted. The case $n = 1 \,\&\, m = 2$ goes $q_0, q_1, q_3, q_5, q_7, q_{10}, q_f$, the case $n = 1 \,\&\, m = 3$ goes $q_0, q_1, q_3, q_5, q_7, q_9, q_{10}, q_f$, and the case $n = 2 \,\&\, m = 3$ goes $q_0, q_2, q_4, q_6, q_8, q_9, q_{10}, q_f$,
    See part c) for a more detailed explanation.