Chap 10

1) a) $S \to X \mid Y \mid aPAbQb$

$X \to aaX \mid \lambda$

$Y \to bbY \mid \lambda$

$P \to aPA \mid \lambda$

$Q \to bQb \mid E$

$Ab \to bA$

$AE \to a$

$Aa \to aa$

b) $S \to ABCDS \mid T_D$

$DA \to AD$

$DB \to BD$

$DC \to CD$

$CA \to AC$

$BA \to AB$

$DT_D \to T_D d$

$T_D \to T_C$

$CT_C \to T_C c$

$T_C \to T_B$

$BT_B \to T_B b$

$T_B \to T_A$

$AT_A \to T_A a$

$T_A \to \lambda$

2) $S \to aAbc$

$\to aaAbCbc$

$\to aaaAbCbCbc$

So, the sentential forms have the shape: $a^i(bCbC)^{i-1}bc$ after 1 application of the first rule and $i-1$ applications of the A rule and 1 application of

$\to$

the non-recursive A rule. It can be reordered by the Cb rules to the form: $a^i b^i C^i A C$. This in turn gives us $a^i b^i c^i$.

4/ a) We have two RE languages, A and B. We create a non-det TM by creating a new start state q0 that non deterministically goes to the start state qA and qB, the start states of the accepters: $M' = \delta(q0, B) = \{[qA, B, R], [qB, B, R]\}$. Then $w \in A \cup B$ if M' accepts.

b) $w \in A \cap B$ if it is in both. So, run A_A and A_B sequencially. If they both halt, and accept, then w is in the language.

c) w decomposed into pairs $uv = w$, wher $u \in A$ and $v \in B$. Create a TM that produces all pairs from w: $(\lambda, w), (w_{0,0}, w_{1,n}) (w_{0,1}, w_{2,n}), \ldots$ Then run each pair through A_A and A_B. If one of the pairs is accepted, it's in the language.

d) Construct a loop so we keep piping the string back into A. If $w \in A^*$ then there must be a division uv such that $u \in A$ and $v \in A^*$. Enumerate the divisions then applications of Kleene.

e) L_1 is RE and so is the image. Run enumerator for L_1 to enumerate the strings in L_1 and convert it to $h(w)$. Then we compare the two strings. If w is in the image, we will enumerate it and accept.

5/ a) b(ab)*

b/ S → aT[a] [q0 B] bT[b]

T[→ aT[A | bT[B_1 | [q0 B

Aa → aA

Ab → bA

B_1a → aB_1

B_1b → bB_1

A] → a]

B_1] → b]

q0Ba → Bq1 a

q0Bb → Bq1b

q1ba → bq2a]

q1bb → bq2b]

q2aa → aq1 a

q2ab → aq1 b

q0B] → Bq1B]

q1 b] → bq2 B]

q2a] → aq1B]

q2b → ER

q2B → ER

ERa → ER

ER] → EL

aEL → EL

bEL → EL

BEL → EL

[EL → Λ

4 [q0Bbab B]

[Bq1bab B]

[Bbq2ab B]

[Bbaq1b B]

[Bbabq2 B]

S → bT[b]

baT[ab]

babT[B_1Ab]

bab[q0 B B_1 b A]

bab[q0BbB$_1$ A]

bab[q0BbB$_1$ a]

bab[q0BbaB_1]

bab[q0 Bbab]

bab[Bq1 ba b]

bab[Bbq2ab]

bab[Bbaq1b]

bab[Bbabq2B]

bab[BbabER]

bab[BbabEL

bab[BbaEL

bab[BbEL

bab[BEL

bab[EL

bab