1. (5+5 points) Consider the following grammar $G$. Note that the grammar does not contain $\lambda$-rules except at $S$.

$$
S \rightarrow aSb \mid DEF \mid D \mid \lambda \\
D \rightarrow E \mid EF \mid abEF \\
E \rightarrow eEff \mid a \mid F \\
F \rightarrow ffFe \mid a
$$

(a) Use algorithm 4.3.1 to construct the CHAIN sets for the variables in $V$.

(b) Construct an equivalent grammar $G_c$ that does not contain chain rules.

2. (5+5 points) Consider the following grammar $G$:

$$
S \rightarrow a \mid aA \mid BC \\
A \rightarrow aB \mid b \\
B \rightarrow Aa \\
C \rightarrow cCD \\
D \rightarrow ddd
$$

(a) Construct the TERM set for $G$.

(b) Use the TERM set to construct an equivalent grammar $G_T$ that does not contain variables that do not generate strings of terminals.

3. (5+5 points) Consider the following grammar $G$ where $\Sigma$ contains every word listed in the rules. $\Sigma = \{ \text{Michigan, Tech, ..., cool} \}$.

$$
S \rightarrow \text{Michigan Tech CS gives } N \mid \text{Having a graduate degree is } R \\
T \rightarrow \text{Being in a computing field is } D \\
N \rightarrow \text{BSc degrees} \mid \text{MSc degrees} \mid \text{PhD degrees} \\
R \rightarrow \text{fun} \mid \text{intellectually challenging} \mid \text{financially rewarding} \mid \text{not as hard as one would think} \mid \text{a worthwhile option to explore} \\
D \rightarrow \text{fun} \mid \text{awesome} \mid \text{cool}
$$

(a) Construct the REACH set for $G$.

(b) Use the REACH set to construct an equivalent grammar $G_U$ that does not contain unreachable variables.
4. (10 points) Convert the following grammar $G$ into Chomsky normal form. Show your steps clearly. Note that $G$ already satisfies the conditions on the start symbol $S$, $\lambda$-rules, useless symbols, and chain rules.

$$ S \rightarrow bT \quad T \rightarrow aAA \mid AbAT \quad A \rightarrow aT \mid bT \mid a $$

5. (10+5 points) Consider the grammar $G$ from Example 4.5.2:

$$ S \rightarrow AT \mid AB \quad T \rightarrow XB \quad X \rightarrow AT \mid AB \quad A \rightarrow a \quad B \rightarrow b $$

(a) Give the upper diagonal matrix produced by the CYK algorithm when run with $G$ and the input string $abbb$. **Show all your work.**

(b) Is $abbb \in L(G)$? Why? Provide the reason based on the upper diagonal matrix you constructed.

6. (10+5+10+5+5 points) Consider the following grammar $G$. Note that $G$ was obtained by transforming the grammar $S \rightarrow aSa \mid bSb \mid a \mid b \mid \lambda$ to Chomsky Normal Form.

$$ S \rightarrow AR \mid BX \mid AA \mid BB \mid a \mid b \mid \lambda $$

$$ T \rightarrow AR \mid BX \mid AA \mid BB \mid a \mid b $$

$$ R \rightarrow TA $$

$$ X \rightarrow TB $$

$$ A \rightarrow a $$

$$ B \rightarrow b $$

(a) Give the upper diagonal matrix produced by the CYK algorithm when run with $G$ and the input string $abba$. **Show all your work.**

(b) Is $abba \in L(G)$? Why? Provide the reason based on the upper diagonal matrix you constructed.

(c) Give the upper diagonal matrix produced by the CYK algorithm when run with $G$ and the input string $abbb$. **Show all your work.**

(d) Is $abbb \in L(G)$? Why? Provide the reason based on the upper diagonal matrix you constructed.

(e) Is $bbb \in L(G)$? Why? Provide the reason based on the upper diagonal matrix you constructed.

7. (6+2+2 points) Consider the following grammar $G$:

$$ S \rightarrow AB \mid CB $$

$$ A \rightarrow aaAbb \mid aaaAbbb \mid Ae \mid e $$

$$ B \rightarrow Bb \mid d $$

$$ C \rightarrow Cee \mid Cff \mid gg $$

(a) Construct a grammar $G'$ that contains no left-recursive rules and is equivalent to $G$.

(b) Give a leftmost derivation on the string $aaebbeeddd$ in grammar $G$.

(c) Give a leftmost derivation on the string $aaebbeeddd$ in grammar $G'$. 