Definition. Let $G$ be a CFG, then $L(G)$ is the language generated by $G$.

6.16)
Claim: The set of languages which can be generated by some grammar in which every production takes one of the following four forms $B \rightarrow a$, $B \rightarrow Ca$, $B \rightarrow aC$, or $B \rightarrow \Lambda$, where $B$ and $C$ are variables and $a \in \Sigma$, is strictly larger than the set of regular languages.

Proof:
Inclusion: By Definition 6.3, a grammar is regular if every production takes one of the following two forms $B \rightarrow a$ or $B \rightarrow aC$, where $B$ and $C$ are variables and $a \in \Sigma$, and by Theorem 6.2 a language, $L$, is regular if $L - \Lambda$ can be generated by a regular grammar. Note that if $L$ is a regular language then so is $L - \Lambda$: since a language only includes $\Lambda$ if the equivalent regular expression contains $\text{Kl}e$ene* or consists of a regular expression unioned with $\Lambda$. In the latter case removing $\Lambda$ clearly results in a regular language. In the former case, if $A^*$ occurs in the regular expression, then $A$ is a regular language, and we may replace $A^*$ in the regular expression by $AA^*$ resulting in another regular language. Hence given a regular language $L$, we may generate the regular language $L - \Lambda$ with a regular grammar which only uses two of the four production forms available. If $\Lambda \in L$ add $S \rightarrow \Lambda$ to, $\delta$, the collection of productions, otherwise don’t change them. The resultant grammar generates $L$.

Grammars of this form can generate non-regular languages: The productions $S \rightarrow A1|\Lambda$ and $A \rightarrow 0S$ generate the non-regular language $\{0^n1^n|n \geq 0\}$.

6.35) Given a CFG, $G$, specify a CFG, $G'$, with neither unit nor $\Lambda$ productions, such that $L(G') = L(G) - \{\Lambda\}$.

a) Productions of $G$: $S \rightarrow ABA$, $A \rightarrow aA|\Lambda$, and $B \rightarrow bB|\Lambda$.

Productions of $G'$: $S \rightarrow ABA|BA|AB|AA|BB|A|B$, $A \rightarrow aA$, and $B \rightarrow bB$.

6.39) Given a CFG, $G$, find a CFG, $G'$, in Chomsky normal form such that $L(G') = L(G) - \{\Lambda\}$.

a) Productions of $G$: $S \rightarrow SS|(S)|\Lambda$.

Productions of $G'$: $S \rightarrow SS|X_iSX_i|X_iX_i$, $X_i \rightarrow (, \text{ and } X_i \rightarrow )$.

b) Productions of $G$: $S \rightarrow S(S)|\Lambda$.

Productions of $G'$: $S \rightarrow SS|XiT_i|X_iX_i$, $T_i \rightarrow SX_i$, $X_i \rightarrow (, \text{ and } X_i \rightarrow )$.

6.56)
Claim: Let $f: \Sigma_1 \rightarrow \Sigma_2$ be a homomorphism, as defined in Exercise 4.46. Then $f(L)$ is a CFL whenever $L$ is a CFL.

Proof: Let $G$ be the grammar such that $L(G) = L$. We may assume that $G$ in Chomsky normal form, i.e., every production is of the form $A \rightarrow BC$ or $A \rightarrow a$, where $A$, $B$, and $C$ are variables and $a \in \Sigma_1$. It suffices to show that if we replace every production of the form $A \rightarrow a$, in $G$, by the production $A \rightarrow f(a)$, then the resulting CFG, $G'$, generates $f(L)$, i.e. $L(G') = f(L)$. It is clear that $G$ can generate the string $a_1a_2\ldots$ if and only if $G'$ can generate the string $f(a_1)f(a_2)\ldots$. Hence, it suffices to show that for any $x = a_1a_2\ldots \in \Sigma_1^*$, $f(x) = f(a_1)f(a_2)\ldots$. But this follows, by induction, from the definition of homomorphism.

7.8) Claim: Every regular language can be accepted by a PDA with only two states in which there are no $\Lambda$ transitions and no symbols are every removed from the stack.
Proof: Let \( \{Q, \Sigma, q_0, A, \delta\} \) be the FA accepting the language \( L \). Define \( Q' = \{\text{True}, \text{False}\} \) and \( \sigma : Q \times \Sigma \rightarrow Q' \) to be such that \( \sigma(q, a) = \text{True} \) if and only if \( \Delta(q, a) \in A \). Define \( \Gamma = Q \) and \( \delta' : Q' \times \Sigma \times \Gamma \rightarrow Q' \times \Gamma^* \) such that \( \delta'(q', a, q) = (\sigma(q, a), \delta(q, a)q) \) for all \( q' \in Q' \). We claim that the PDA \( \{Q', \Sigma, \Gamma, q_0, A, Z_0, \delta'\} \), where \( \delta'(\text{True}, \Lambda, q) \) signals acceptance, accepts the language \( L \). But this follows from the observation that after processing the first \( n \) elements of a string, the top of the stack of the PDA and the current state of the FA are the same.