Tutorial 2: Solutions

Exercise 1: The alphabet Σ is the set of all printable ASCII characters:

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

For each of the following languages, find an accepting grammar:

1. the decimal representations of odd numbers
2. the set of palindromes (words that are equal to their mirrored image)
3. the *arithmetic expressions*: every nonempty sequence of digits is an arithmetic expression, if e_1 and e_2 are two arithmetic expressions, then so are the words $e_1\texttt{+}e_2$, $e_1\texttt{-}e_2$, $e_1\texttt{*}e_2$, $e_1\texttt{/}e_2$, and $\texttt{(}e_1\texttt{)}$.

Solution:

1. The grammar $(\Sigma, \mathscr{N}, S, \mathscr{P})$ where Σ is the set of ASCII characters, $\mathscr{N} = \{S, O, M\}$, and $\mathscr{P}$ contains the following rules:
$$S \to MO \qquad O \to \texttt{1} \mid \texttt{3} \mid \texttt{5} \mid \texttt{7} \mid \texttt{9}$$
$$M \to \texttt{0}M \mid \texttt{1}M \mid \texttt{2}M \mid \texttt{3}M \mid \texttt{4}M \mid \texttt{5}M \mid \texttt{6}M \mid \texttt{7}M \mid \texttt{8}M \mid \texttt{9}M \mid \varepsilon$$
accepts the decimal representation of an odd number.

   The grammar only produces words containing digits and the last digits of the word are odd numbers (since $S \to MO$, and the non-terminal O only produces odd numbers). Then, from the non-terminal M we can either derive ε or any digit.

   Note: The notation $S \to A \mid B$ is equivalent to the (union of) the rules $S \to A$ and $S \to B$.

2. The grammar $(\Sigma, \mathscr{N}, S, \mathscr{P})$ where Σ is the set of ASCII characters, $\mathscr{N} = \{S\}$, and $\mathscr{P}$ is:
$$S \to \varepsilon \mid \alpha S \alpha \mid \alpha \qquad \text{for all} \qquad \alpha \in \Sigma$$
accepts the language of palindrome letters.

   The grammar accept ε ($S \to \varepsilon$), it can derive a palindrome word with an even number of letters applying multiple times the rule $S \to \alpha A \alpha$ (e.g., $S \to \texttt{a}A\texttt{a} \to \texttt{ab}A\texttt{ba} \to \texttt{abba}$), or it can derive a palindrome word with an odd number of letters when applying the rule $S \to \alpha$ (e.g., $S \to \texttt{a}$, $S \to \texttt{a}A\texttt{a} \to \texttt{ab}A\texttt{ba} \to \texttt{abcba}$).

3. The grammar $(\Sigma, \mathscr{N}, S, \mathscr{P})$ where Σ is the set of ASCII characters, $\mathscr{N} = \{S, D\}$, and $\mathscr{P}$ is:
$$S \to D \mid S\texttt{+}S \mid S\texttt{-}S \mid S\texttt{/}S \mid S\texttt{*}S \mid \texttt{(}S\texttt{)}$$
$$D \to \texttt{0}D' \mid \texttt{1}D' \mid \texttt{2}D' \mid \ldots \mid \texttt{9}D'$$
$$D' \to D \mid \varepsilon$$

accepts the language of arithmetic expressions.

The production rules of the grammar follows the inductive definition of an arithmetic expression. An arithmetic expression is either a sequence of a digit (rule $S \to D$, D' is used to end the construction of the Digit at some point), or is obtained as a binary operation on two other arithmetic expressions (rule $S{-}S$), or is an arithmetic expression enclosed in brackets (rule (S)).

Exercise 2: Let L be the language accepted by the grammar $(\Sigma, \mathcal{N}, S, \mathcal{P})$ where:

- $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, $\mathcal{N} = \{S, A, B, C, X, Y\}$, S is the start symbol, and
- the production rules (i.e. the elements of $\mathcal{P}$) are:

$$S \to ABC \qquad A \to A\mathtt{a}X \qquad X\mathtt{a} \to \mathtt{a}X \qquad XB \to B\mathtt{b}X \qquad X\mathtt{b} \to \mathtt{b}X \qquad XC \to C\mathtt{c}$$

$$A \to Y \qquad Y\mathtt{a} \to \mathtt{a}Y \qquad Y\mathtt{b} \to \mathtt{b}Y \qquad YB \to Y \qquad YC \to \varepsilon$$

Find two or three words that belong to L then guess what L is.
Solution: We can derive the following words applying the rule the grammar:

- $S \to ABC \to YBC \to YC \to \varepsilon$
  Applying in order the rules $S \to ABC$, $A \to Y$, $YB \to Y$, $YC \to \varepsilon$.
- $S \to ABC \to A\mathtt{a}XBC \to A\mathtt{a}B\mathtt{b}XC \to A\mathtt{a}B\mathtt{b}C\mathtt{c} \to Y\mathtt{a}B\mathtt{b}C\mathtt{c} \to \mathtt{a}YB\mathtt{b}C\mathtt{c} \to \mathtt{a}Y\mathtt{b}C\mathtt{c} \to \mathtt{ab}YC\mathtt{c} \to \mathtt{abc}$
  Applying in order the rules: $S \to ABC$, $A \to A\mathtt{a}X$, $XB \to B\mathtt{b}X$, $XC \to C\mathtt{c}$, $A \to Y$, $Y\mathtt{a} \to \mathtt{a}Y$, $YB \to Y$, $Y\mathtt{b} \to \mathtt{b}Y$, $YC \to \varepsilon$.
- $S \to ABC \to A\mathtt{a}XBC \to A\mathtt{a}B\mathtt{b}XC \to A\mathtt{a}X\mathtt{a}B\mathtt{b}XC \to A\mathtt{aa}XB\mathtt{b}XC \to A\mathtt{aa}B\mathtt{b}X\mathtt{b}XC \to$

$$A\mathtt{aa}BbbXXC \to A\mathtt{aa}BbbXCc \to A\mathtt{aa}BbbCcc \to Y\mathtt{aa}B\mathtt{bb}C\mathtt{cc} \to \mathtt{a}Y\mathtt{a}B\mathtt{bb}C\mathtt{cc} \to$$

$$\mathtt{aa}YB\mathtt{bb}C\mathtt{cc} \to \mathtt{aa}Y\mathtt{bb}C\mathtt{cc} \to \mathtt{aab}Y\mathtt{b}C\mathtt{cc} \to \mathtt{aabb}YC\mathtt{cc} \to \mathtt{aabbcc}$$

  Applying in order the rules:
  $S \to ABC$, $A \to A\mathtt{a}X$, $XB \to B\mathtt{b}X$, $A \to A\mathtt{a}X$, $X\mathtt{a} \to \mathtt{a}X$, $XB \to B\mathtt{b}X$, $X\mathtt{b} \to \mathtt{b}X$,
  $XC \to C\mathtt{c}$, $XC \to C\mathtt{c}$, $A \to Y$, $\mathtt{a}Y \to Y\mathtt{a}$, $\mathtt{a}Y \to Y\mathtt{a}$,
  $YB \to Y$, $Y\mathtt{b} \to \mathtt{b}Y$, $Y\mathtt{b} \to \mathtt{b}Y$, $YC \to \varepsilon$.

The language accepted by the grammar is the set $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \in \mathbb{N}\}$. The non-terminal A generates an instance of a and "stores" that the grammar generated one a in the non-terminal X. Note that, in order to derive a "new" b character (a b that was not already in the current derivation), the grammar requires to match the subword XB. Similarly, one need to have XC to generate a new c. The grammar has also rules to "move" the non-terminal X on the right of the word (i.e., switching X with the letters a and b). The non-terminal Y is used to remove the instances of the non-terminal A, B, C and derive a word containing only terminal symbols.

Let us prove it formally[1]. We prove that the following properties are invariant when applying each of the rule of $\mathcal{P}$. If w is a word over $\Sigma \cup \mathcal{N}$, and $l \in \Sigma \cup \mathcal{N}$, we let $|w|_l$ be the number of ls in w.

1. S will never appear after the first application of a production rule. If A disappear, it will not disappear. Indeed, the only way to produce a new A is to use $S \to ABC$ that can not be used twice. The same applies for B and C.

[1] We do not expect you to do it in an exam.

2. We have $|w|_A \leq 1$, $|w|_B \leq 1$ and $|w|_C \leq 1$. Indeed, it is the case at start (S) and each of the rule preserve that property.

3. If $|w|_A = 1$ then A is the first letter of w. Indeed, it is the case at start ($|S|_A = 0$) and each rule put A at the beginning of the word if A is already present.

4. If $|w|_B = 1$ then B is before any $\mathtt{b}$ in w.

5. If $|w|_C = 1$ then C is before any $\mathtt{c}$ in w.

6. Any $\mathtt{a}$ is before B when $|w|_B = 1$. Assume it is not true. Let w be built from S with the rules $\mathscr{P}$ and steps $w_0 = S \to w_1 \to \cdots \to w_n \to w$ such that w violates that property. Without loss of generality we assume w to be the first word in the sequence to violate the property. Since S does not violate the property, the sequence is well defined ($n \geq 0$). We the write $w = u_1 B u_2 \mathtt{a} u_3$ with u_i be (possibly empty) words over $\Sigma \cup \mathscr{N}$. We have the following cases

   - $w_n \to w$ implies B. If it is $S \to ABC$, using property 1. we have $w = ABC$ which is not. If it is $YB \to Y$, then $1 \geq |w|_B = |w_n|_B - 1 \leq 1 - 1 = 0$ using property 2., what is not. Then we have just applied $XB \to B\mathtt{b}X$. Using 2., w_n as only one B. We then know that $w_n = u_1 X B u'_2 \mathtt{a} u_3$ with $u_2 = \mathtt{b} X u'_2$. Then w_n violates the property 6. and then contradicts the minimality of w.
   - $w_n \to w$ implies the $\mathtt{a}$ of w we are looking at. Then we do the same thing as the previous case to derive that w_n violates the minimality of w.
   - $w_n \to w$ does not implies the B of the $\mathtt{a}$. Then $w_n = u'_1 B u'_2 \mathtt{a} u'_3$ with $u'_i \to u_i$ for one $i \in \{1,2,3\}$ and $u'_j = u_j$ for $j \neq i$. Then w_n violates the minimality of w.

   Then property 6. is an invariant.

7. Any $\mathtt{b}$ is before C when $|w|_C = 1$. We do the same proof as for property 6.

8. $|w|_{\mathtt{a}} = |w|_{\mathtt{b}} + |w'|_X$ where $w = w'w''$ and w' does not contains any $\mathtt{b}$ and is maximal for this property (w'' is empty or starts with a $\mathtt{b}$). Indeed, the only rule that we need to pay attention is $A \to A\mathtt{a}X$. But using property 3., we conclude that there is no problem.

9. $|w|_{\mathtt{b}} = |w|_{\mathtt{c}} + |w''|_X$ where $w = w'w''$ nd w' does not contains any $\mathtt{b}$ and is maximal for this property (w'' is empty or starts with a $\mathtt{b}$).

10. If $w \neq S$ then every $\mathtt{a}$ is before any $\mathtt{b}$. Indeed, if $|w|_B = 1$ we conclude using 4. and 6. In the sequence to build w, when $|w_i|_B = 0$ for the first time with $i > 0$, it is also the case (the rule applied is $YB \to Y$) and the other rules let the property invariant when there is no more B.

11. If $w \neq S$ then every $\mathtt{b}$ is before any $\mathtt{c}$. We do the same as the previous proof using 5. and 7.

We are now ready to prove that the language is $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \in \mathbb{N}\}$.

- Assume w is accepted. Then it contains only terminal elements, $\mathtt{a}$,$\mathtt{b}$, or $\mathtt{c}$. Using invariants 8. and 9. we conclude that $|w|_{\mathtt{a}} = |w|_{\mathtt{b}} = |w|_{\mathtt{c}}$. Using 10. and 11. we conclude that $w = \mathtt{a}^n\mathtt{b}^m\mathtt{c}^p$. To sum it up, $w \in \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \in \mathbb{N}\}$.

- It remains to prove that if $w \in \{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \in \mathbb{N}\}$, then w is accepted. We just give a sequences of rules to build it :

$$
\begin{array}{lcccc}
S & \to & ABC & \to & A\mathtt{a}XBC \\
 & \to & \cdots & \to & A(\mathtt{a}X)^nBC \\
 & \to & Y(\mathtt{a}X)^nBC & \underbrace{\to^*}_{\text{using } X\mathtt{a}\to\mathtt{a}X \ n \text{ times}} & Y\mathtt{a}^nX^nBC \\
 & \underbrace{\to^*}_{\text{using } XB\to B\mathtt{b}X \ n \text{ times}} & Y\mathtt{a}^nB(\mathtt{b}X)^nC & \underbrace{\to^*}_{\text{using } X\mathtt{b}\to\mathtt{b}X \ n \text{ times}} & Y\mathtt{a}^nB\mathtt{b}^nX^nC \\
 & \underbrace{\to^*}_{\text{using } XC\to C\mathtt{c} \ n \text{ times}} & Y\mathtt{a}^nB\mathtt{b}^nC\mathtt{c}^n & \underbrace{\to^*}_{\text{using } Y\mathtt{a}\to\mathtt{a}Y \ n \text{ times}} & \mathtt{a}^nYB\mathtt{b}^nC\mathtt{c}^n \\
 & \to & \mathtt{a}^nY\mathtt{b}^nC\mathtt{c}^n & \underbrace{\to^*}_{\text{using } Y\mathtt{b}\to\mathtt{b}Y \ n \text{ times}} & \mathtt{a}^n\mathtt{b}^nYC\mathtt{c}^n \\
 & \multicolumn{4}{l}{\to \mathtt{a}^n\mathtt{b}^n\mathtt{c}^n}
\end{array}
$$

Then, w is accepted.

Exercise 3: Given a word w on $\{\mathtt{a},\mathtt{b}\}$ the grammar L_w is defined as follows:

$$S \to AwB \quad A \to AX \quad X\mathtt{ba} \to \mathtt{ab} \quad X\mathtt{ab} \to \mathtt{a}X\mathtt{b} \quad X\mathtt{aa} \to \mathtt{a}X\mathtt{a} \quad X\mathtt{bb} \to \mathtt{b}X\mathtt{b} \quad X\mathtt{b}B \to \mathtt{b}XB$$

$$XB \to Y \quad X\mathtt{a}B \to \mathtt{a}Y \quad \mathtt{a}Y \to Y\mathtt{a} \quad \mathtt{b}Y \to Y\mathtt{b} \quad XY \to Y \quad AY \to \varepsilon$$

1. Compute L_w for each word w in the set $\{\varepsilon,\mathtt{ab},\mathtt{ba},\mathtt{aba}\}$, then guess what L_w is (compute L_w for more instances of w if needed, the pattern should appear clearly).
2. What if we remove the production rule $XY \to Y$ from the grammar L_w ?

Solution:

1. We compute L_w for each word w in the set $\{\varepsilon,\mathtt{ab},\mathtt{ba},\mathtt{aba}\}$:
   - $w = \varepsilon$: $S \to AB \to AXB \to AY \to \varepsilon$.
     The language $L_\varepsilon = \emptyset$.
   - $w = \mathtt{ab}$: $S \to A\mathtt{ab}B \to AX\mathtt{ab}B \to A\mathtt{a}X\mathtt{b}B \to A\mathtt{ab}XB \to A\mathtt{ab}Y \to A\mathtt{a}Y\mathtt{b} \to AY\mathtt{a}b \to \mathtt{ab}$.
     The language $L_{\mathtt{ab}} = \{\mathtt{ab}\}$.
   - $w = \mathtt{ba}$: $S \to A\mathtt{ba}B \to AX\mathtt{ba}B \to A\mathtt{ab}B \xrightarrow{*} \mathtt{ab}$.
     Note that we already saw the derivation from $A\mathtt{ab}$ in the case $w = \mathtt{ab}$. The language $L_{\mathtt{ba}} = \{\mathtt{ab}\}$.
   - $w = \mathtt{aba}$: $S \to A\mathtt{aba}B \to AX\mathtt{aba}B \to A\mathtt{a}X\mathtt{ba}B \to A\mathtt{aab}B \to AX\mathtt{aab}B \to A\mathtt{a}X\mathtt{ab}B \to A\mathtt{aa}X\mathtt{b}B \to A\mathtt{aab}XB \to A\mathtt{aab}y \to A\mathtt{aa}Y\mathtt{b} \to A\mathtt{a}Y\mathtt{ab} \to AY\mathtt{aab} \to \mathtt{aab}$.
     The language $L_{\mathtt{aba}} = \{\mathtt{aab}\}$.

   The language L_w is the language that contains the word $\mathtt{a}^n\mathtt{b}^m$ where n is the number of $\mathtt{a}$ in w and m is the number of $\mathtt{b}$ in w.

2. If we remove $XY \to Y$ any sequence XY cannot be removed anymore and one obtain an infinite derivation that cannot terminate:

   $$S \to AB \to AXB \to AY \to AXY \to AXXY \to \ldots$$

   Instead, with the rule $XY \to Y$, the infinite derivation shown above still exists, but at any point in time we can decide to apply $XY \to Y$ and obtain a finite derivation.

   The language L_w does not change removing the rule $XY \to Y$ from the grammar. Thus, even if the two grammars have different derivations they still accept the same language.

Exercise 4: In a grammar $(\Sigma, \mathscr{N}, S, \mathscr{P})$, a nonterminal symbol A is said to be *persistent* when for *all* production rules $w \to w'$, if A appears in w then A also appears in w'. A production rule is said to be *persistent* when it contains a persistent (nonterminal) symbol. What is the language recognized by the grammar if the start symbol S is persistent? Assuming that S is not persistent, prove that persistent symbols and persistent production rules can be removed from a grammar without changing its accepted language, i.e.

$$[\![\Sigma, \mathscr{N}, S, \mathscr{P}]\!] \quad = \quad [\![\Sigma, \mathscr{N} - \{\text{persistent}\}, S, \mathscr{P} - \{\text{persistent}\}]\!] \,.$$

Solution: If the initial symbol S is persistent then the language $[\![G]\!]$ of G is empty since from S one can only apply persistent rules, hence always obtaining an S in every new derivation. The language is empty because there are no derivations starting from S that terminates with a word $w \in \Sigma^*$.

Let $G = (\Sigma, \mathscr{N}, S, \mathscr{P})$ be a grammar, $\mathscr{N}_P$ be the set of persistent non-terminal symbols of G (not containing S), $\mathscr{P}_P$ be the set of persistent production G, and $G_P = (\Sigma, \mathscr{N} - \mathscr{N}_P, S, \mathscr{P} - \mathscr{P}_P)$ the grammar obtained removing the persistent non-terminals and productions from G.

We prove that $[\![G]\!] = [\![G_P]\!]$.

- $[\![G_P]\!] \subseteq [\![G]\!]$.

  If $w \in [\![G_P]\!]$, then there exists a derivation in G_P such that $S \xrightarrow[G_P]{*} w$ (the notation $\xrightarrow[G_P]{*}$ describes 0 or more derivations using any rule from the grammar G_P). We show by induction on the length of the derivation that for any derivation $S \xrightarrow[G_P]{*} w$ there is a derivation $S \xrightarrow[G]{*} w$.

  - **Base case:** Suppose $S \xrightarrow[G_P]{1} w$. If $S \xrightarrow[G_P]{1} w$, then there exists a rule $(S, w) \in \mathscr{P}_P$. Since $\mathscr{P}_P \subseteq \mathscr{P}$, then $(S, w) \in \mathscr{P}$ and hence $S \xrightarrow[G]{1} w$.
  - **Inductive case:**
    Suppose $S \xrightarrow[G_P]{n+1} w$. By inductive hypothesis we have that $S \xrightarrow[G_P]{n} v \xrightarrow[G_P]{1} w$ and $S \xrightarrow[G]{n} v$. Since $v \xrightarrow[G_P]{1} w$, there exists $(\gamma, \delta) \in \mathscr{P}_P$ such that $v = \alpha\gamma\beta$, for some $\alpha, \gamma \in (\Sigma \cup \mathscr{N})^*$, and $w = \alpha\delta\beta$ (this directly from the definition of a derivation for a grammar). As before, $(\gamma, \delta) \in \mathscr{P}$ and then we can apply this rule to derive $v \xrightarrow[G]{1} w$. Thus we have that $S \xrightarrow[G]{n} v \xrightarrow[G]{1} w$.

  Thus, if $w \in [\![G_P]\!]$ then there exists a derivation $S \xrightarrow[G_P]{*} w$ and hence a derivation $S \xrightarrow[G]{*} w$, and hence $w \in [\![G]\!]$.

- $[\![G]\!] \subseteq [\![G_P]\!]$

  We first show that when we start from a word that includes a persistent non-terminal symbol A, all the possible derivations with the grammar G will always contain a A. In practice, we show that there are no derivations that can "remove" a non-terminal persistent symbol A from a word.

  If A is a persistent non-terminal (i.e., $A \in \mathscr{N}_P$), then any word $w = vAv' \in (\Sigma \cup \mathscr{N})^*$, for some $v, v' \in (\Sigma \cup \mathscr{N})^*$, is such that $w \xrightarrow[G]{*} w'$, then $w' = uAu'$ for some $u, u' \in (\Sigma \cup \mathscr{N})^*$.

  We prove the lemma by induction on the length of the derivation.

  - **Base case:** Suppose $w \xrightarrow[G]{1} w'$ and $w = vAv' \in (\Sigma \cup \mathscr{N})^*$, for some $v, v' \in (\Sigma \cup \mathscr{N})^*$.

    If $w \xrightarrow[G]{1} w'$, then there exists a rule in $(\delta, \gamma) \in \mathscr{P}$ that derives w' from vAv'. There are two possible cases:

1. (δ,γ) is not persistent (i.e., $(\delta,\gamma) \in \mathscr{P} - \mathscr{P}_P$). In this case, A cannot appear in δ, and hence the non-terminal A is not removed from w.
2. (δ,γ) is persistent. In this case $\delta = w_\delta A w'_\delta$ and $\gamma = w_\gamma A w'_\gamma$ for some $w_\delta, w'_\delta, w_\gamma, w'_\gamma \in (\Sigma \cup \mathscr{N})^*$. Thus, the derivation with (δ,γ) will contain at least one occurrence of the non-terminal A (i.e., $vAv' \xrightarrow[G]{1} uAu'$).

– **Inductive case:** Suppose $w \xrightarrow[G]{n} w'' \xrightarrow[G]{1} w'$. By inductive hypothesis, $w'' = zAz', z, z' \in (\Sigma \cup \mathscr{N})^*$. We can prove that $w'' = zAz' \xrightarrow[G]{1} w'' = uAu'$, for some $u, u' \in (\Sigma \cup \mathscr{N})^*$ as above, showing that if A is already in w'' than A is also in w' after a single derivation step.

Now we prove $[\![G]\!] \subseteq [\![G_P]\!]$ by contradiction. Suppose $w \notin G_p$. If $w \notin G_p$, then $S \xrightarrow[G]{n} w$ is such that there is a derivation at a step $0 \le i \le n$ that uses a persistent rule $(w_\delta A w'_\delta, w_\gamma A w'_\gamma)$, for some persistent non-terminal A.

Thus, we have $S \xrightarrow[G]{i-1} v \xrightarrow[G]{i} v' \xrightarrow[G]{n-i} w$. with $v = \alpha A \beta$ and $v' = \alpha' A \beta'$ for some $\alpha, \beta, \alpha', \beta' \in (\Sigma \cup \mathscr{N})^*$.

Since A is in the word v, then A is also in the word w (by the lemma we proved above. But this is a contradiction, since $w \in [\![G]\!]$ and hence cannot contain non-terminals. We conclude that $[\![G]\!] \subseteq [\![G_P]\!]$.

Exercise 5: Prove that <u>if we allow *infinite* sets of production rules and *infinite* sets of nonterminal symbols</u> in the definition of a grammar, then for any language L there is a regular grammar such that $L = [\![G]\!]$.

Solution:

We define a regular grammar (with an infinite set of production rules and an infinite set of nonterminal symbols) that can accept an arbitrary language L.

The grammar is $G = (\Sigma_L, \mathscr{N}, S, \mathscr{P})$ where:

- Σ_L is the alphabet of L;
- $\mathscr{N} = \{S\} \cup \{A_{w_i} \mid w \in L, 1 \le i < |w|\}$.
  We define a non-terminal symbol for every letter $w(i)$ in every (non-empty) word w of L. We will use a non-terminal A_{w_i} to keep track that the grammar generated the prefix $w(1)w(2)\dots w(i)$ of w. We will further add a rule that generates the next character $w(i+1)$ when matching the non-terminal A_{w_i}. We further have an initial symbol S. Note that $|w|$ denotes the length of the word w.
- We define a set of production rules that can generate any word in L:

$$
\begin{aligned}
\mathscr{P} = & \bigcup_{\varepsilon \in L} \{S \to \varepsilon\} \cup && \text{Accept the empty word} \\
& \bigcup_{w \in L \setminus \{\varepsilon\}} \{S \to w(1) A_{w_1}\} \cup && \text{Derives the first character of } w \\
& \bigcup_{w \in L \setminus \{\varepsilon\}} \bigcup_{1 \le i < |w|} \{A_{w_i} \to w(i+1) A_{w_{i+1}}\} \cup && \text{Derives the prefix of length i of } w \\
& \bigcup_{w \in L \setminus \{\varepsilon\}} \{A_{w_{|w|}} \to \varepsilon\} && \text{Accepts the word } w
\end{aligned}
$$

In practice, the grammar can either generate the empty word ε (if it is part of L), or non-deterministically choose to generate the first character of a word $w \in L$, picking one rule $S \to w(1)A_{w_1}$. At this point the grammar can only pick the rule that contains A_{w_1} on the left-hand side, hence starting to derive the word w, that is finally accepted with the rule $A_{w_{|w|}}$.

We prove that $[\![G]\!] = L$.

- $L \subseteq [\![G]\!]$ We prove that if $w \in L$ then $w \in [\![G]\!]$.

  If $|w| = 0$, then $S \xrightarrow[G]{i} \varepsilon$, and hence $\varepsilon \in [\![G]\!]$.

  We then show that for any prefix of w, $|w| > 0$, of lenght $i \leq |w|$ there exists a derivation $S \xrightarrow[G]{i} w(1)w(2)\dots w(i)A_{w_i}$. We prove it by induction on the prefix length.

  - **Base case.** For $i = 1$ we have that $S \xrightarrow[G]{1} w(1)A_{w_1}$, since $S \to w(1)A_{w_1} \in \mathscr{P}$.
  - **Inductive case.** For $i = i+1 \leq |w|$, we have that $S \xrightarrow[G]{i} w(1)\dots w(i)A_{w_1}$ by inductive hypothesis. By the definition of G, there exists a rule $A_{w_i} \to w(i+1)A_{w_{i+1}} \in \mathscr{P}$. Using the inductive hypothesis and the above rule we conclude that $S \xrightarrow[G]{i} w(1)\dots w(i)A_{w_1} \xrightarrow[G]{1} w(1)\dots w(i+1)A_{w_{i+1}}$.

  Thus, there exist a derivation $S \xrightarrow[G]{|w|} w(1)\dots w(|w|)A_{w_{|w|}}$. We conclude that $w \in [\![G]\!]$ since $A_{w_{|w|}} \to \varepsilon \in \mathscr{P}$.

- $[\![G]\!] \subseteq L$

  Let $w \in [\![G]\!]$, and $S \xrightarrow[G]{*} w$ its derivation using the grammar G.

  We show that for any derivation $S \xrightarrow[G]{i} w'$ either w' is part of the language L or $w' = w''(1)\dots w''(i)A_{w_i}$, for a $w'' \in L$.

  We prove it by induction on the length of the derivation.

  - **Base case**. Consider the case $i = 1$ and the derivation $S \xrightarrow[G]{1} w'$

    After one derivation there are two sub-cases to consider: either a) $S \xrightarrow[G]{1} \varepsilon$ or b) $S \xrightarrow[G]{1} w(1)A_{w_1}$ for some word w from the definition of $\mathscr{P}$ (i.e., $\mathscr{P}$ only has rules with S on the left-hand side for the rules $S \to \varepsilon$ and $S \to w(1)A_{w_1}$).

    In case a), $S \xrightarrow[G]{1} \varepsilon$ because of the production $S \to \varepsilon$. By construction we have that $(S \to \varepsilon) \in \mathscr{P}$ iff $\varepsilon \in L$. Thus $w' = \varepsilon$ is a word in L.

    In case b), $S \xrightarrow[G]{1} w(1)A_{w_1}$ because of the production $S \to w(1)A_{w_1}$. By construction, $w(1)$ is a prefix of a word w in L.
  - **Inductive case.** Consider the case $i = i+1$ and the derivation $S \xrightarrow[G]{i+1} w'$.

    By inductive hypothesis we know that $S \xrightarrow[G]{i} v$ is such that either v is a word in Ł or $v = w(1)\dots w(i)A_{w_i}$ for a word $w \in L$. The first case is impossible, since v would only contain non-terminal symbols ($v \in \Sigma^*$), and the grammar is regular (i.e., all the left-hand sides of the rules are non-terminals). Thus, we know that $S \xrightarrow[G]{i} w(i)A_{w_i}$.

We know that the only rules in the $\mathscr{P}$ containing the non-terminal A_{w_i} are either $A_{w_i} \rightarrow w(i+1)A_{w_{i+1}}$, if $i \leq |w|$, or $A_{|w|} \rightarrow \varepsilon$, if $i > |w|$.

In the first case we have that $w' = w(i)\dots w(i+1)A_{w_{i+1}}$ for a word $w \in L$, since $S \xrightarrow[G]{i} w(i)\dots w(i)A_{w_i} \xrightarrow[G]{1} w(i)\dots w(i+1)A_{w_{i+1}}$.

In the second case, we have that $w' \in L$ since $S \xrightarrow[G]{i} w(1)\dots w(|w|)A_{|w|} \xrightarrow[G]{i} w(i)\dots w(|w|)$.

We conclude that if $w \in [\![G]\!]$ then $w \in L$.

Exercise 6: Prove that <u>if we allow *infinite* sets of production rules</u> in the definition of a grammar, then for any language L there is a context-free grammar such that $L = [\![G]\!]$.
Solution:

We define a regular grammar (with an infinite set of production rules) that can accept an arbitrary language L. The grammar is $G = (\Sigma_L, \mathscr{N}, S, \mathscr{P})$ where:

- Σ_L is the alphabet of L;
- $\mathscr{N} = \{S\}$.
- $\mathscr{P} = \bigcup_{w \in L} \{S \rightarrow w\}$

The grammar just define a production rule for every word $w \in L$. Every rule $S \rightarrow w$ can derive a word $w \in L$ from the initial symbol S.

If $w \in L$ then $w \in [\![G]\!]$ because there exists a derivation $S \xrightarrow[G]{1} w$ with the rule $S \rightarrow w \in \mathscr{P}$.

If $w \in [\![G]\!]$, then there exists a derivation $S \xrightarrow[G]{*} w$. Since all the rules in $\mathscr{P}$ have only S in the left-hand side, then $S \xrightarrow[G]{*} w$ is $S \xrightarrow[G]{1} w$, for some rule $S \rightarrow w \in \mathscr{P}$. If $S \rightarrow w \in \mathscr{P}$, then $w \in L$ by the definition if G.

Exercise 7: Given two grammars $G_1 = (\Sigma_1, \mathscr{N}_1, S_1, \mathscr{P}_1)$ and $G_2 = (\Sigma_2, \mathscr{N}_2, S_2, \mathscr{P}_2)$ find two grammars $G_1' = (\Sigma_1', \mathscr{N}_1', S_1', \mathscr{P}_1')$ and $G_2' = (\Sigma_2', \mathscr{N}_2', S_2', \mathscr{P}_2')$ such that:

1. $\Sigma_1' = \Sigma_2'$
2. $\mathscr{N}_1' \cap \mathscr{N}_2' = \emptyset$
3. $[\![G_1]\!] = [\![G_1']\!]$ and $[\![G_2]\!] = [\![G_2']\!]$

Deduce that:

1. the union of two formal / context-free languages is formal / context-free.
2. the concatenation of two formal / context-free languages is formal / context-free.
3. the Kleene star of a formal / context-free language is formal / context-free.

Solution:

We define a function that renames the non-terminal symbols of $\mathscr{N}_2$ ensuring that they are disjoint from $\mathscr{N}_1$:

$$\phi(A) = \begin{cases} A & \text{when } A \in \mathscr{N}_2 \setminus \mathscr{N}_1 \\ A' & \text{otherwise} \end{cases}$$

We assume that each A' is a new symbol (i.e., not contained in any of Σ_1, Σ_1, $\mathscr{N}_1$ $\mathscr{N}_2$), and that $\phi(A) = \phi(B) iff A = B$.

We extend ϕ to words in $(\Sigma_2 \cup \mathscr{N}_2)^*$ where we rename only the symbols in a word that are from $\mathscr{N}_2$:

$$\phi(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ \phi(a)\phi(w') & \text{if } w = aw' \text{ and } a \in \Sigma_2 \\ \phi(A)\phi(w') & \text{if } w = Aw' \text{ and } A \in \mathscr{N}_2 \end{cases}$$

Then, we define:

- $G'_1 = (\Sigma'_1 = \Sigma_1 \cup \Sigma_2, \mathscr{N}_1' = \mathscr{N}_1, S'_1 = S_1, \mathscr{P}'_1 = \mathscr{P}_1)$
- $G'_2 = (\Sigma'_2 = \Sigma_1 \cup \Sigma_2, \mathscr{N}_2' = \{\phi(A) \mid A \in \mathscr{N}_2'\}, S'_2 = \phi(S_2), \mathscr{P}'_2 = \{\phi(w) \to \phi(w') \mid w \to w' \in \mathscr{P}_2\})$

Here we are also assuming that $\mathscr{N}_1 \cap \Sigma_2 = \emptyset$ and $\mathscr{N}_2 \cap \Sigma_1 = \emptyset$ (otherwise, we should further rename $\mathscr{N}_1$ to force it to be dijoint from Σ_2 and consider Σ_1 when renaming $\mathscr{N}_2$).

We can easily prove that $[\![G_1]\!] = [\![G'_1]\!]$ (in fact, the two grammars have the same derivations). We can prove that $[\![G_2]\!] = [\![G'_2]\!]$ showing that they have the same derivations but with the non-terminals renamed with ϕ in G'_2 (we also need an inverse of ϕ when showing the direction $\supseteq$).

Closure properties of Context-free languages

We first show that context-free languages are closed under union, concatenation, and Kleene closure.

If two languages L_1, L_2 are context-free then there exists two context-free grammars $G_1 = (\Sigma_1, \mathscr{N}_1, S_1, \mathscr{P}_1)$ and $G_2 = (\Sigma_2, \mathscr{N}_2, S_2, \mathscr{P}_2)$ such that $L_1 = [\![G_1]\!]$ and $L_2 = [\![G_2]\!]$. We assume that the two grammars G_1, G_2 have two disjoint sets of non-terminal symbols (i.e., $\mathscr{N}_1 \cap \mathscr{N}_2 = \emptyset$). This is not a general restriction since the set of non-terminals can be renamed while still accepting the same language (with the construction shown above).

- **Union.**

  We define the union of the two grammars as $G_1 \cup G_2 = (\Sigma_{G_1 \cup G_2}, \mathscr{N}_{G_1 \cup G_2}, S, \mathscr{P}_{G_1 \cup G_2})$ where:

  - $\Sigma_{G_1 \cup G_2} = \Sigma_1 \cup \Sigma_2$
  - $\mathscr{N}_{G_1 \cup G_2} = \mathscr{N}_1 \cup \mathscr{N}_2 \cup \{S\}$
  - $\mathscr{P}_{G_1 \cup G_2} = \mathscr{P}_1 \cup \mathscr{P}_2 \cup \{S \to S_1, S \to S_2\}$.

  The union of the grammars keeps the same productions of G_1 and G_2 and just adds two productions $S \to S_1$ and $S \to S_2$. In this way, a derivation in $G_1 \cup G_2$ will first non-deterministically "select" to follow either the derivations of G_1 (and hence deriving a word in L_1) or the derivations of G_2 (and hence deriving a word in L_2).

  We first show that if L_1 and L_2 are context free, then $G_1 \cup G_2$ is context free. Then we will show that the formal grammar $G_1 \cup G_2$ recognizes the same language of $L_1 \cup L_2$.

  We show that if G_1 and G_2 are context free grammars (they exists since L_1 and L_2 are context-free), then also $G_1 \cup G_2$ is a context-free grammar. It is easy to see that all the production rules in $\mathscr{P}_{G_1 \cup G_2}$ are of the form $A \to w$, where $A \in \mathscr{N}_{G_1 \cup G_2}$ and $w \in (\Sigma_{G_1 \cup G_2})^*$. A rule in $w \to w' \in \mathscr{P}_{G_1 \cup G_2}$ is either:

– A rule $w \to w' \in \mathscr{P}_1$. Since G_1 is context-free, then $w \to w'$ is such that $w \in \mathscr{N}_1$ and $w' \in (\Sigma_1 \cup \mathscr{N}_1)^*$.
– A rule $A \to w \in \mathscr{P}_2$. This is a context-free rule for $\mathscr{P}_{G_1 \cup G_2}$, similarly to the rules from $\mathscr{P}_1$.
– $S \to S_1$: this rule is context-free.
– $S \to S_2$: this rule is context-free.

Now we show that $[\![G_1 \cup G_2]\!] = L_1 \cup L_2$.

– $[\![G_1 \cup G_2]\!] \supseteq L_1 \cup L_2$.
If $w \in L_1 \cup L_2$ then either $w \in L_1$ or $w \in L_2$.
If $w \in L_1$ then $w \in [\![G_1]\!]$, and then there exists a derivation $S_1 \xrightarrow[G_1]{*} w$. We have that $S \xrightarrow[G_1 \cup G_2]{1} S_1$ (since $S \to S_1 \in \mathscr{P}_{G_1 \cup G_2}$) and that $\mathscr{P}_1 \subseteq \mathscr{P}_{G_1 \cup G_2}$. Thus, we have a derivation $S \xrightarrow[G_1 \cup G_2]{1} S_1 \xrightarrow[G_1 \cup G_2]{*} w$ showing that $w \in [\![G_1 \cup G_2]\!]$.
If $w \in L_2$, we have a simmetric proof showing that $w \in [\![G_1 \cup G_2]\!]$.
– $[\![G_1 \cup G_2]\!] \subseteq L_1 \cup L_2$
If $w \in [\![G_1 \cup G_2]\!]$ then there exists a derivation $S \xrightarrow[G_1 \cup G_2]{*} w$.
Since the only derivation containing S on the left-hand side are $S \to S_1$ and $S \to S_2$, then we have either that a) $S \xrightarrow[G_1 \cup G_2]{1} S_1 \xrightarrow[G_1 \cup G_2]{*} w$ or b) $S \xrightarrow[G_1 \cup G_2]{1} S_2 \xrightarrow[G_1 \cup G_2]{*} w$.
In the case a), since $\mathscr{N}_1$ and $\mathscr{N}_2$ are disjoint and the grammar is context-free (i.e., every left-hand side of a rule is a single non-terminal) then the only possible derivations from the word S_1 uses productions rules from G_1. Thus, there exists a derivation $S_1 \xrightarrow[G_1]{*} w$, and hence $w \in [\![G_1]\!]$, $w \in L_1$, and $w \in L_1 \cup L_2$.
The case b) is simmetric to the case a).

Since $G_1 \cup G_2$ is a context-free gramar and $[\![G_1 \cup G_2]\!] = L_1 \cup L_2$, then the union $L_1 \cup L_2$ is a context-free language.

- **Concatenation.**
We define the concatenation of two grammar as: $G_1G_2 = (\Sigma_{G_1G_2} = \Sigma_1 \cup \Sigma_2, \mathscr{N}_{G_1G_2} = \mathscr{N}_1 \cup \mathscr{N}_2 \cup \{S\}, S, \mathscr{P}_{G_1G_2} = \mathscr{P}_1 \cup \mathscr{P}_2 \cup \{S \to S_1S_2\})$.
The production $S \to S_1S_2$ allows the grammar to accept the concatenation of a word derived following only productions in G_1 (and hence in L_1) to a word derived following only productions in G_2 (and hence in L_2).
All the production rules in $\mathscr{P}_{G_1G_2}$ are of the form $A \to w$, where $A \in \mathscr{N}_{G_1G_2}$ and $w \in (\Sigma_{G_1G_2} \cup \mathscr{N}_{G_1G_2})^*$. The only "new" rule is $S \to S_1S_2$, which is context-free. Thus the grammar is context-free.
Now we show that $[\![G_1G_2]\!] = L_1L_2$.

– $[\![G_1G_2]\!] \supseteq L_1L_2$.
If $w \in L_1 \cap L_2$ then there exists two words $w_1 \in L_1$ and $w_2 \in L_2$.
Then there exist two derivations $S_1 \xrightarrow[G_1]{n} w_1$ and $S_2 \xrightarrow[G_2]{m} w_2$.
We can construct a derivation in G_1G_2 from S to $w = w_1w_2$ as: $S \xrightarrow[G_1G_2]{1} S_1S_2 \xrightarrow[G_1]{n} w_1S_2 \xrightarrow[G_2]{m} w_1w_2$

- $[\![G_1G_2]\!] \subseteq L_1L_2$.

  If $w \in [\![G_1G_2]\!]$ then there exists a derivation $S \xrightarrow[G_1G_2]{*} w$.

  We show that a derivation in G_1G_2 always derives a word $w = w_1w_2$ such that w_1 can be derived from G_1 and w_2 can be derived from G_2 (i.e., $S \xrightarrow[G_1]{*} w_1$ and $S_2 \xrightarrow[G_2]{*} w_2$). This is sufficient to show that a derivation in G_1G_2 deriving a word $w \in (\Sigma_1 \cup \Sigma_2)^*$ is the concatenation of two words derived from G_1 and G_2.

  * **Base case.**
    In the first step the only possible derivation is $S \xrightarrow[G_1G_2]{1} S_1S_2$. S_1 and S_2 are two possible derivations of G_1 and G_2 (with 0 steps).
  * **Inductive case.** Suppose we have $S \xrightarrow[G_1G_2]{n} w'_1w'_2 \xrightarrow[G_1G_2]{1} w$.
    By inductive hypothesis we have that $S \xrightarrow[G_1]{n} w'_1$ and $S_2 \xrightarrow[G_2]{n} w'_2$, and that $w'_1 \in (\Sigma_1 \cup \mathscr{N}_1)^*$ and $w'_2 \in (\Sigma_2 \cup \mathscr{N}_2)^*$.
    Thus, there is a rule $(\gamma, \delta) \in \mathscr{P}_{G_1G_2}$ such that $w'_1w'_2 \xrightarrow[G_1G_2]{1} w$. We know that both G_1 and G_2 are context-free, and their non-terminal symbols are disjoint. Thus, it is the case that the derivation is done with a rule in $\mathscr{P}_{G_1G_2}$ that is either from $\mathscr{P}_1$ or $\mathscr{P}_2$. In practice, we have either $w'_1w'_2 \xrightarrow[G_1G_2]{1} w_1w'_2$ or $w'_1w'_2 \xrightarrow[G_1G_2]{1} w'_1w_2$.
    To see this, notice that $w'_1 \in (\Sigma_1 \cup \mathscr{N}_1)^*$ and a rule from $\mathscr{P}_{G_1G_2}$ is of the form (A, δ), with either a) $A = S$, b) $A \in \mathscr{N}_1$, or c) $A \in \mathscr{N}_2$. We cannot be in case a), since the derivation is at step $n > 1$. We cannot be in case c) either, since $w'_1 \in (\Sigma_1 \cup \mathscr{N}_1)^*$ and $(\Sigma_1 \cup \mathscr{N}_1) \cap \mathscr{N}_2 = \emptyset$. Thus, the only possible case to apply a rule matching a substring in w_1 is b). The same reasoning works for w'_2.

  Thus, if $S \xrightarrow[G_1G_2]{*} w = w_1w_2$, then $S \xrightarrow[G_1]{*} w_1$ and $S \xrightarrow[G_2]{*} w_2$. Thus, if w is a word on $[\![G_1G_2]\!]$, then $w = w_1w_2$ and $w_1 \in [\![G_1]\!] = L_1$ and $w_2 \in [\![G_2]\!] = L_2$, and hence $w \in L_1L_2$.

- **Klenee closure.** Let $G_1 = (\Sigma_1, \mathscr{N}_1, S_1, \mathscr{P}_1)$ be the grammar that accepts the formal language L_1.

  We define the Kleene closure of G_1 as: $G_1^* = (\Sigma_{G_1^*} = \Sigma_1, \mathscr{N}_{G_1^*} = \mathscr{N}_1 \cup \{S\}, S, \mathscr{P}_{G_1^*} = \mathscr{P}_1 \cup \{S \to \varepsilon, S \to S_1S\})$.

  The rule $S \to \varepsilon$ allows the grammar to recognize $\varepsilon \in L_1^0$ and terminate a derivation, while $S \to S_1S$ allows the grammar to derive a word $w \in L_1$ concatenated to a word that can be derived again from S.

  G_1^* is a context-free grammar, since G_1 is a context-free grammar and the additional rules to $\mathscr{P}_1$ are $S \to \varepsilon$ and $S \to S_1S$.

  Now we show that $[\![G_1^*]\!] = L_1^*$.

  - $[\![G_1^*]\!] \supseteq L_1^*$.

    We show that if $w \in L_1^*$ then there exists a derivation $S \xrightarrow[G_1^*]{*} w$, and thus we conclude that $w \in [\![G_1^*]\!]$.
    If $w \in L_1^*$, then we have that $w \in \bigcup_{i \geq 0} L_1^i$, from the definition of the Kleene closure.

    We have that if $w = \varepsilon$ (i.e., $w \in L_1^0$), then $S \xrightarrow[G_1^*]{1} \varepsilon$ (because $S \to \varepsilon \in \mathscr{P}_{G_1^*}$).

We then show that for any $w' \in L_1^n$, $n > 1$, the grammar G_1^* can derive the word $w'S$. That is, the grammar can derive a word $w' \in L_1^n$ and always concatenate a new word (including a word from $w'' \in L_1$) with derivation from the non-terminal S. This allows us to conclude the first language inclusion. We prove this by induction on the number of words from L_1 concatenated in w.

* **Base case.** Suppose $w \in L_1$.
  Then there exists a derivation $S_1 \xrightarrow[G_1]{*} w$ of G_1 that we can use to show that there is the following derivation in G_1^*, since all the productions of G_1 are in G_1^*: $S \xrightarrow[G_1^*]{1} S_1S \xrightarrow[G_1^*]{*} wS_1$.
* **Inductive case.** Suppose $w \in L_1^n$. Thus, we know that $w = w''w'$, for some $w'' \in L_1^{n-1}$ and $w' \in L_1$. So, we also have a derivation in G_1 such that $S_1 \xrightarrow[G]{*} w'$
  By inductive hypothesis, we have that there exists a derivation $S \xrightarrow[G_1^*]{*} w''S$ and hence we have the following derivation:
  $S \xrightarrow[G_1^*]{*} w''S \xrightarrow[G_1^*]{1} w''S_1S \xrightarrow[G_1^*]{*} w''w'S$

Applying the above result, we have that if $w \in L_1^*$ then there exists a derivation $S \xrightarrow[G_1^*]{*} wS \xrightarrow[G_1^*]{1} w$, and hence we conclude that $w \in [\![G_1^*]\!]$.

– $[\![G_1^*]\!] \subseteq L_1$.
Given a word $w \in [\![G_1^*]\!]$ we want to show that $w \in L_1$. As usual, we know that there exists a derivation $S \xrightarrow[G_1^*]{*} w$.

Showing that the derivation $S \xrightarrow[G_1^*]{*} w$ produces a word in L_1^* is more involved, because the grammar can "start" several derivations from several non-terminals S_1. For example:
$S \xrightarrow{G_1^*} S_1S \xrightarrow{G_1^*} S_1S_1S \xrightarrow{G_1^*} S_1S_1S_1S \ldots$
We can show that, in any possible derivations of G_1^*, we obtain a word $w = u_1 \ldots u_n$ that is the concatenation of either: a) a word from L_1 (i.e., $u_i \in L_1$), b) a word (including terminal and non-terminals) that can be derived from G_1, c) the non-terminal S.
Thus, when the derivation accepts a word $w \in \Sigma_1^*$, we know that $w \in L_1^*$.
We prove this by induction on the number of derivations in $S \xrightarrow[G_1^*]{*} w$

* **Base case.** If $S \xrightarrow[G_1^*]{1} w$ we either have that $S \xrightarrow[G_1^*]{1} \varepsilon$ or $S \xrightarrow[G_1^*]{1} S_1S$.
* **Inductive case.** If $S \xrightarrow[G_1^*]{n} w$, we know by inductive hypothesis that $S \xrightarrow[G_1^*]{n-1} u'_1 \ldots u'_k$ where a u_i is either a) $u'_i \in L_1$, b) $u'_i \in (\Sigma_1 \cup \mathcal{N}_1)^*$ and $S \xrightarrow[G_1]{*} u'_i$, or c) $u'_i = S$.
  Since the grammar is context-free, then the derivation rule at the step n is only applied to one of the substrings u_i if they are either in the cases b) or in c).
  In the case b), then one must apply a rule from G_1 (since u_i only contains non-terminals from G_1). Since we have that $S \xrightarrow[G_1]{*} u'_i$ and we apply a rule from G_1, then we have that $S \xrightarrow[G_1]{*} u'_i \xrightarrow[G_1]{*} u_i$, with u_i again in case b) or a) (if the word $u_1 \in \Sigma_1^*$).
  In the case c) the only derivations are with the rules $S \to \varepsilon$ or $S \to S_1S$. In the first case,

we have the derivation: $S \xrightarrow[G_1^*]{n-1} u'_1 \dots u'_n \xrightarrow[G_1^*]{1} u'_1 \dots u'_{i-1} u'_{i+1} \dots u'_k$. In the second case, we have the derivation: $S \xrightarrow[G_1^*]{n-1} u'_1 \dots u'_n \xrightarrow[G_1^*]{1} u'_1 \dots u'_{i-1} S_1 S u'_{i+1} \dots u'_k$.

Since $S \xrightarrow[G_1^*]{n} w$ with $w \in \Sigma^*$, then $w \in L_1$.

Formal languages. Formal languages are closed under union, concatenation, and Kleene closure.

However, the grammars provided before for the union/concatenation/Kleene closure of context-free may not recognize the union/concatenation/Kleene closure if the languages are not context-free.

To see this, consider the grammars with the following productions (S_1 and S_2 are the starting symbols):

$$S_1 \to b$$

and:

$$S_2 \to AabA \qquad Aa \to a \qquad bA \to bb$$

We have that $[\![G_1]\!] = \{b\}$, $[\![G_2]\!] = \{abb\}$, and $[\![G_1]\!][\![G_2]\!] = \{babb\}$. However, the language obtained with the grammar G_1G_2 as defined above would be $[\![G_1 \cup G_2]\!] = \{babb, bbabb\}$.

The word *bbabb* is not part of the concatenation $[\![G_1]\!][\![G_2]\!]$ and is obtained with the derivation:

$$S \to S_1S_2 \to bS_2 \to bAabA \to bbabA \to bbabb$$

The issue here is that a rule from G_2 is applied also on a terminal from G_2. This example shows that the above definition of contatenation of grammars is not correct to show that unconstrained grammars are closed under concatenation. We can find similar examples for the union and the Kleene closure.

Exercise 8: Let Σ_1 and Σ_2 be two alphabets. A language morphism is a mapping $h : \Sigma_1^* \to \Sigma_2^*$ such that for all words $w, w' \in \Sigma_1^*$ we have $h(ww') = h(w)h(w')$.

1. Prove that h preserves the empty word (i.e. $h(\varepsilon) = \varepsilon$).
2. Prove that h is entirely determined by its restriction to Σ_1 (i.e. the words of length 1).
3. Prove that any map $f : \Sigma_1 \to \Sigma_2^*$ extends to a language morphism in a unique way.

   Let $\mathscr{N}$ be a set of nonterminal symbols and $\tilde{h}$ be the unique extension of h to $(\Sigma_1 \cup \mathscr{N})^*$ such that $\tilde{h}(A) = A$ for all $A \in \mathscr{N}$. Given a grammar G on $\Sigma_1 \cup \mathscr{N}$, we define $\tilde{h}(G)$ as the grammar on $\Sigma_2 \cup \mathscr{N}$ whose production rules are $(\tilde{h}(w), \tilde{h}(w'))$ where (w, w') is a production rule of $\mathscr{P}_1$.
   Prove the following statements:

4. $h([\![G]\!]) \subseteq [\![\tilde{h}(G)]\!]$.
5. If the grammar G is context-free, then so is $\tilde{h}(G)$ and $h([\![G]\!]) = [\![\tilde{h}(G)]\!]$.
6. If the language $[\![G]\!]$ is regular then so is $[\![\tilde{h}(G)]\!]$. ⚠: In general, the grammar $\tilde{h}(G)$ is not regular.

Solution:

1. We prove that $h(\varepsilon) = \varepsilon$ proving that the length of $|h(\varepsilon)|$ is 0.

$$\begin{aligned} |h(\varepsilon)| =& |h(\varepsilon\varepsilon)| && \varepsilon = \varepsilon\varepsilon \\ & |h(\varepsilon)h(\varepsilon)| && \text{Definition of morphism } h(ww') = h(w)h(w') \\ & |h(\varepsilon)| + |h(\varepsilon)| && |ww'| = |w| + |w'| \end{aligned}$$

Now we can substract $|h(\varepsilon)|$ on both sides:

$$\begin{aligned} |h(\varepsilon)| - |h(\varepsilon)| &= |h(\varepsilon)| + |h(\varepsilon)| - |h(\varepsilon)| \\ 0 &= |h(\varepsilon)| \end{aligned}$$

Since $|h(\varepsilon)| = 0$, then $h(\varepsilon) = \varepsilon$.

2. We need to prove that $h(w) = h(w(1))h(w(2))\ldots h(w(n))$ (i.e., it is completely determined by the restriction of h to a single letter of the alphabet).

   We can prove it by induction on the length of the word w.

   - If $w = \varepsilon$ then $h(\varepsilon) = \varepsilon$ (from the result in 8.1).
   - Consider a word w such that $|w| = n > 0$. We have $h(w(1)\ldots w(n-1)) = h(w(1))\ldots h(w(n-1))$ by inductive hypothesis.

$$\begin{aligned} h(w) =& h(w(1)\ldots w(n-1)w(n)) & \\ & h(w(1)\ldots w(n-1))h(w(n)) & \text{Since } h(ww') = h(w)h(w') \\ & h(w(1))\ldots h(w(n-1))h(w(n)) & \text{By inductive hypothesis} \end{aligned}$$

3. We want to show that any map $f : \Sigma_1 \to \Sigma_2^*$ extends to a morphism $h : \Sigma_1^* \to \Sigma_2^*$.

   We can define a morphism h as the extension of f as follows:

$$h(w) = \begin{cases} \varepsilon & \text{if } w = \varepsilon \\ h(w')f(a) & \text{if } w = w'a, \text{ for some } w' \in \Sigma^* \end{cases}$$

   We know that (if h is a morphism), it is uniquely determined by its restrictions to Σ_1. The map f is such restriction: $h(a) = h(\varepsilon)f(a) = \varepsilon f(a) = f(a)$.

   We just need to prove that h is a morphism showing that for every $w, w' \in \Sigma^*$ $h(ww') = h(w)h(w')$. We prove it by induction on the length of strings $w' \in \Sigma^*$:

   - When $w' = \varepsilon$ we have $h(w\varepsilon) = h(w)$
   - Let $|w'| = m > 0$ and $|w| = n$.

$$\begin{aligned} h(ww') =& h(w(1)\ldots w(n)w'(1)\ldots w'(m)) & \\ =& h(w(1)\ldots w(n)w'(1)\ldots w'(m-1))f(w'(m)) & \text{Defintion of } h \\ =& h(w(1)\ldots w(n))h(w'(1)\ldots w'(m-1))f(w'(m)) & \text{By inductive hypothesis} \\ =& h(w(1)\ldots w(n))h(w'(1)\ldots w'(m-1))h(w'(m)) & \forall a \in \Sigma . h(a) = f(a) \\ =& h(w(1)\ldots w(n))h(w'(1)\ldots w'(m-1)w'(m)) & \text{Definition of } h \end{aligned}$$

4. Let $G = (\Sigma_1, \mathscr{N}, S, \mathscr{P}_1)$ and $\tilde{h}(G) = (\Sigma_2, \mathscr{N}, S, \tilde{\mathscr{P}})$ where $\tilde{h}(\mathscr{P}) = \{(\tilde{h}(w), \tilde{h}(w')) \mid (w, w') \in \mathscr{P}_1\}$. We prove that $h([\![G]\!]) \subseteq [\![\tilde{h}(G)]\!]$.

   $w \in h([\![G]\!])$ means that there exists a word $v \in \Sigma_1^*$ and a derivation in G such that $h(v) = w$ and $S \xrightarrow[G]{*} v$.

   We show by induction on the length of the derivation that for any derivation $S \xrightarrow[G]{*} v$ there exists a derivation $S \xrightarrow[\tilde{h}(G)]{*} w$ and $h(v) = w$.

- **Base case:** If $S \xrightarrow[G]{1} v$ then there exists $S \to v \in \mathscr{P}$, and hence. $S \to \tilde{h}(v) \in \tilde{h}(\mathscr{P}_1)$. Thus there exists a derivation $S \xrightarrow[\tilde{h}(G)]{1} \tilde{h}(v)$.

- **Inductive case:** Suppose $S \xrightarrow[G]{n} v' \xrightarrow[G]{1} v$.

  Then there exists a rule $(w, w') \in \mathscr{P}_1$ such that for two $\alpha, \beta \in \Sigma_1^*$, $v' = \alpha w \beta \xrightarrow[G]{1} \alpha w' \beta = v$.

  By inductive hypothesis we also have that $S \xrightarrow[\tilde{h}(G)]{n} \tilde{h}(v')$.

  Also, we have that $(\tilde{h}(w), \tilde{h}(w')) \in \tilde{h}(\mathscr{P})$ and that, since $\tilde{h}$ is a morphism, $\tilde{h}(v') = \tilde{h}(\alpha)\tilde{h}(w)\tilde{h}(\beta)$.

  We thus have the following derivation in $\tilde{h}(G)$: $v' = \tilde{h}(\alpha)\tilde{h}(w)\tilde{h}(\beta) \xrightarrow[\tilde{h}(G)]{1} \tilde{h}(\alpha)\tilde{h}(w')\tilde{h}(\beta)$.

  Then, since $\tilde{h}(h)$ is a morphism, we have that $\tilde{h}(\alpha)\tilde{h}(w')\tilde{h}(\beta) = \tilde{h}(\alpha w' \beta) = \tilde{h}(v)$

  We conclude that there is a derivation: $S \xrightarrow[\tilde{h}(G)]{n} \tilde{h}(v') \xrightarrow[\tilde{h}(G)]{1} \tilde{h}(v)$.

Since for any derivation $S \xrightarrow[G]{*} v$ there exists a derivation $S \xrightarrow[\tilde{h}(G)]{*} w$ and $h(v) = w$, we conclude that if $w \in [\![G]\!]$ then $w \in [\![\tilde{h}(G)]\!]$

5. We start showing that if G is context-free then $\tilde{h}(G)$ is context free.

   We show that $\forall (w, w') \in \tilde{h}(\mathscr{P})$, (w, w') is context-free. If $(w, w') \in \tilde{h}(\mathscr{P})$ then there is a rule $(v, v') \in \mathscr{P}_1$ such that $\tilde{h}(v) = w$ and $\tilde{h}(v') = w'$. Since G is context-free, then (v, v') is such that $v = A$, for some non-terminal symbol $A \in \mathscr{N}$, and $v' \in (\Sigma_1 \cap \mathscr{N})^*$. Then, from the definition of $\tilde{h}(G)$ and $\tilde{h}$, we know that $w = \tilde{h}(v) = \tilde{h}(A) = A$, $w' = \tilde{h}(v')$, and $\tilde{h}(v') \in (\Sigma_2 \cap \mathscr{N})^*$. Thus, any rule $(w, w') \in \tilde{h}(\mathscr{P})$ is also context-free. We conclude that $\tilde{h}(G)$ is context-free.

   We shown above in 8.4 that $h([\![G]\!]) \subseteq [\![\tilde{h}(G)]\!]$ for an unrestricted grammar. Here we prove that if G is context-free, then $h([\![G]\!]) \supseteq [\![\tilde{h}(G)]\!]$. We show that if $w \in [\![\tilde{h}(G)]\!]$ then $w \in h([\![G]\!])$.

   If $w \in [\![\tilde{h}(G)]\!]$ then there exists a derivation $S \xrightarrow[\tilde{h}(G)]{*} w$. We show that there exists a derivation $S \xrightarrow[G]{*} v$ such that $h(v) = w$, and hence that if $w \in [\![\tilde{h}(G)]\!]$, then there exists a $v \in \Sigma_1^*$ such that $v \in [\![G]\!]$ and $h(v) = w$, and hence that $w \in h([\![G]\!])$.

   We prove the existence of the derivation by induction on the derivation length.

   - if $S \xrightarrow[\tilde{h}(G)]{1} w$, then $(S, w) \in \tilde{h}(\mathscr{P})$. But there exists $(S, v) \in \mathscr{P}_1$ such that $w = \tilde{h}(v)$. Hence $S \xrightarrow[G]{1} v$, and $h(v) = w$.

   - Suppose $S \xrightarrow[\tilde{h}(G)]{n} w' \xrightarrow[\tilde{h}(G)]{1} w$.

     By inductive hypothesis we also know that $S \xrightarrow[G]{n} v'$ with $\tilde{h}(v') = w'$.

     We miss to show that $v' \xrightarrow[G]{1} v$ with $\tilde{h}(v) = w$.

     Since G is context-free, the derivation $w' \xrightarrow[\tilde{h}(G)]{1} w$ uses a rule $(A, \delta) \in \tilde{h}(\mathscr{P})$ of the grammar $\tilde{h}(G)$, where $A \in \mathscr{N}_1$, such that, for some $\alpha, \beta, \in \Sigma_2^*$, $w = \alpha A \beta \xrightarrow[\tilde{h}(G)]{1} \alpha \delta \beta = w'$.

Since $(A,\delta) \in \tilde{h}(\mathscr{P})$, then there exists $(A,\delta') \in \mathscr{P}$ such that $\tilde{h}(A) = A$ and $\tilde{h}(\delta') = \delta$.

We have that

$$\tilde{h}(v') = w' = \alpha A \beta = \tilde{h}(\alpha')\tilde{h}(A)\tilde{h}(\beta')$$

for some $\alpha', \beta' \in \Sigma_1^*$, and hence the derivation: $v' = \alpha' A \beta' \xrightarrow[G]{1} \alpha'\delta'\beta' = v$

We further now that $\tilde{h}(v = \alpha'\delta'\beta') = \tilde{h}(\alpha')\tilde{h}(\delta)\tilde{h}(\beta') = w$, and hence that $v' \xrightarrow[G]{1} v$ with $\tilde{h}(v) = w$.

Question: why did we need the assumtpion for G to be context-free? Can you find a counter-example showing that if G is an unrestricted grammar (i.e., a formal grammar without restrictions on the left or right-hand side, apart from avoiding ε in the left-hand side) then $h(\llbracket G \rrbracket) \not\supseteq \llbracket \tilde{h}(G) \rrbracket$?

6. We want to show that if the language $\llbracket G \rrbracket$ is regular, then so is $\llbracket \tilde{h}(G) \rrbracket$.

   If $\llbracket G \rrbracket$ is regular then there exists a regular grammar G' such that $\llbracket G \rrbracket = \llbracket G' \rrbracket$.

   We show that if G' is a regular grammar, then also $\tilde{h}(G')$ is a regular grammar. Given a rule in (w,w') in the production rules of $\tilde{h}(G')$, then there is a rule (v,v') in the production rules of G' such that $\tilde{h}(v) = w$ and $\tilde{h}(v') = w'$. Since G' is regular, then (v,v') can either be in one of the three admissible forms for a rule in a regular grammar. We show that (w,w') is also of one of such form. The rule (v,v') can either be such that:

   - $v = A$ and $v' = \varepsilon$. In this case, we have that $w = \tilde{h}(A) = A$ and $w' = \tilde{h}(\varepsilon) = \varepsilon$;
   - $v = A$ and $v' = a$, for some $a \in \Sigma_1^*$. In this case $w = \tilde{h}(A) = A$ and $w' = \tilde{h}(a)$. By definition we know that $\tilde{h}(a) = b$ for some $b \in \Sigma_2$;
   - $v = A$ and $v' = aA$, for some $a \in \Sigma_1^*$ and $A \in \mathscr{N}$. In this case $w = \tilde{h}(A) = A$ and $w' = \tilde{h}(aA) = \tilde{h}(a)\tilde{h}(A) = \tilde{h}(a)A = bA$, where $\tilde{h}(a) = b \in \Sigma_2$.

   All the rules in $\tilde{h}(G')$ are of one of the kind admissible for a regular grammar and thus $\tilde{h}(G')$ is a regular grammar.

   Since $\llbracket G \rrbracket = \llbracket G' \rrbracket$, then we have that $h\llbracket G \rrbracket = h\llbracket G' \rrbracket$

   From 8.5, we also now that $h(\llbracket G \rrbracket) = \llbracket \tilde{h}(G) \rrbracket$ and $h(\llbracket G' \rrbracket) = \llbracket \tilde{h}(G') \rrbracket$.

   Thus we have:

$$\llbracket \tilde{h}(G) \rrbracket = h(\llbracket G \rrbracket) = h(\llbracket G' \rrbracket) = \llbracket \tilde{h}(G') \rrbracket$$

   Since $\llbracket \tilde{h}(G) \rrbracket = \llbracket \tilde{h}(G') \rrbracket$, and $\llbracket \tilde{h}(G') \rrbracket$ is regular, then also $\llbracket \tilde{h}(G) \rrbracket$ is regular.