Exercise: Give a grammar for the language \( \{ a^n b^n | n \geq 1 \} \).

Solution: \( S \rightarrow ab | aSb \)

Exercise: Give a grammar for the language \( \{ a^n b^{2n} | n \geq 1 \} \).

Solution: \( S \rightarrow aSb | abb \)  

This solution can be trivially extended to generate \( \{ a^n b^{nk} | n \geq 1, k \geq 0 \} \).

\( S \rightarrow aSb | a^n b^k \)

Exercise: Give a grammar for palindrome strings on \( \Sigma = \{ a, b \} \), i.e., for the language \( \{ w \in \Sigma^* | w = w^R \} = \{ w w^R | w \in \Sigma^* \} \cup \{ w w^R | w \in \Sigma^+ \text{ and } w \Sigma \} \).

Solution: \( S \rightarrow aSa | bSb | a | b | e \)

Exercise: Give a grammar for the language \( \{ w w^R | w \in (a,b)^+ \} \) (palindromes on \( \Sigma = \{ a, b \} \) having even length).

Solution: \( S \rightarrow aSa | bSb | e \)

Similarly, we can generate \( \{ w w^R | w \in (a,b)^+ \} \) with the grammar \( S \rightarrow aSa | bSb | aa | bb \).
Exercise 2.12. Give a grammar generating the language
\[ \{ a^n b^m c^n \mid m \geq 1 \} . \]

Solution. (Simpler than the one presented in class.)

\[
\begin{align*}
S & \rightarrow aSCc \mid abc \\
S & \rightarrow Bc \\
S & \rightarrow bb
\end{align*}
\]

Exercise 2.13. Give a grammar for the language
\[ L = \{ a^n b^m c^n a^m \mid m \geq 1 \} . \]

Solution.

\[
\begin{align*}
S & \rightarrow aSCCd \mid abcd \\
S & \rightarrow BD \\
D & \rightarrow CD \\
D & \rightarrow BC \\
C & \rightarrow BC \\
B & \rightarrow bB \\
B & \rightarrow bc \quad (\ast) \\
C & \rightarrow cc
\end{align*}
\]

Notice that the production marked with (\ast) can be applied earlier than necessary, leading to strings that do not produce any string of \((V_1)^*\); this is fine, since we do not have "spurious" strings with respect to our desired language.
Exercise

Construct the minimum DFA equivalent to the one in the figure below:

\[
\begin{array}{c}
A \\
\downarrow 1 \\
C \\
\downarrow 1 \\
D \\
\downarrow 1 \\
E \\
\rightarrow \quad 0 \\
\rightarrow F
\end{array}
\]

Solution

We construct the table of distinguishabilities:

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

First, we mark immediately all pairs in which C, D or F appear together with some non-final state (we put mark "0" here).

Then we mark pairs \{(C, F)\} and \{(D, F)\}; in fact for each of such pairs \((q_i, q_j)\) we have that the pair

\[\{\delta(q_i, a), \delta(q_j, a)\}\]

is a pair marked with "0" at the previous step.
We go on, marking analogously \( \{A, E\} \) and \( \{B, E\} \) with "2", and realizing that there is no error here to mark.

The partitions of equivalent blocks are \( \{A, B\} \) and \( \{D, C\} \).

The minimal equivalent DFA is:

```
\[ \text{Diagram of DFA} \]
```
Exercise

Give a grammar for the language \( \{ w w | w \in \{a, b\}^* \} \).

\[
S \rightarrow aAS \mid bBS \mid aA_0 \mid bB_0
\]
\[(A_0, B_0 \text{ mark the end of the string})\]

\[
\begin{align*}
Aa &\rightarrow aA \\
Ba &\rightarrow aB \\
Bb &\rightarrow bB \\
B_0 &\rightarrow aB \\
AA_0 &\rightarrow A_a \\
BA_0 &\rightarrow B_a \\
A_0B_0 &\rightarrow A_b \\
BB_0 &\rightarrow B_b \\
A_0 &\rightarrow a \\
B_0 &\rightarrow b
\end{align*}
\]

Note that if we apply the last two productions too early, we get strings that do not produce any string made of terminal symbols only.
Exercise

Give a grammar for the language \( \{ a^n \mid n \geq 0 \} \).

**Solution**

\[
egin{align*}
S & \rightarrow TaHF | a \\
    & \quad | aH \rightarrow HaK \\
    & \quad | IH \rightarrow IK \\
    & \quad | K a \rightarrow a a K \\
    & \quad | K F \rightarrow HF \\
    & \quad | K F \rightarrow e \\
    & \quad | IH \rightarrow e \\
I & \rightarrow e \\
F & \rightarrow e
\end{align*}
\]

H and K are two markers that multiply the number of \( a \) in the middle of the string by two; I and F mark the beginning and the end of the string respectively. H goes from right to left and turns itself into K when it reaches I; K goes from left to right and turns itself into H when it reaches F.
Exercise (4.1.1 from textbook)

Construct the minimal DFA equivalent to the following:

DFA Diagram:

```
A -- 0 --> B -- 1 --> C
|       |     |
| 0     | 1   |
|   1   |

B -- 0 --> D

C -- 0 --> E

D -- 0 --> F

E -- 0 --> G

G -- 0 --> H
```

Transition Table:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>B</td>
</tr>
<tr>
<td>D</td>
<td>E</td>
<td>D</td>
</tr>
<tr>
<td>E</td>
<td>F</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>D</td>
</tr>
</tbody>
</table>

Solution

The distinguishability table is as follows:

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>H</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```
The equivalence classes are
\[ \{A, G\}, \{B, F\}, \{C, E\}, \{D\}, \{H\} \].

The minimized automaton is as follows:

State \( H \) is not reachable and needs to be eliminated.

Notice that in the initial automaton, states \( E, F, G \) and \( H \) are not reachable. If we eliminate the non-reachable states, we obtain the following automaton:

Which is already minimal. (Verification is left to the reader).