Exercise 1: My very first \textit{PSPACE}-complete problem

Show that the following problem is \textit{PSPACE}-complete (not assuming anything about QBF):

\begin{itemize}
  \item INPUT: a Turing Machine $M$ and a word $w$ and a number $t$ written in unary
  \item QUESTION: does $M$ accepts $w$ within space $|t|$ ?
\end{itemize}

Exercise 2: Padding argument

Show that the following problem is \textit{PSPACE}-complete (not assuming anything about QBF):

\begin{itemize}
  \item INPUT: a context-sensitive language $L$ and a word $w$
  \item QUESTION: does $w \in L$ ?
\end{itemize}

\textit{Hint: You can find inspiration in the exercise 7 of last week’s exercise sheet, and in proposition 3.8 from the lecture notes.}

Exercise 3: \textit{PSPACE} and games

\textbf{FORMULA-GAME} is the following game. There are two players, Player 1 and Player 2 which alternatively make moves on a given board. The board is a boolean formula $\phi(x_1, \ldots, x_{2n})$, and the moves of the players consist in picking truth values for the variables $x_1, \ldots, x_{2n}$ in this order. Specifically, Player 1 choses the value of $x_1$, then Player 2 choses the value of $x_2$, then Player 1 choses the value of $x_3$, and so on. Player 1 wins the game if $\phi$ is true under the variable assignment produced in the game. Player 1 has a winning strategy if he has a way of choosing his moves so that he wins the game no matter the moves of Player 2.

Show that the following problem is \textit{PSPACE}-complete:

\begin{itemize}
  \item INPUT: a boolean formula $\phi$
  \item QUESTION: does Player 1 have a winning strategy for \textit{FORMULA-GAME} on board $\phi$?
\end{itemize}

Exercise 4: \textit{PSPACE} and games

The Geography game is played as follow:

\begin{itemize}
  \item The game starts with a given name of a city, for instance \textit{Cachan};
  \item the first player gives the name of a city whose first letter coincides with the last letter of the previous city, for instance \textit{Nice};
  \item the second player gives then another city name, always starting with the last letter of the previous city, for instance \textit{Evry};
\end{itemize}
- the first player plays again, and so on – with the restriction that no player is allowed to give the name of a city already used in the game;

- the loser is the first player who does not find a new city name to continue.

This game can be described using a graph whose vertices represent cities and where an edge \((X, Y)\) means that the last letter of the city \(X\) is the same as the first letter of the city \(Y\). This graph has also a vertex marked as the initial vertex of the game (the initial city). Each player choses a vertex of the graph, the first player choses first, and the two players alternate their moves. At each move, the sequence of vertices chosen by the two players must form a simple path in the graph, starting from the distinguished initial vertex.

Player 1 wins the game if, after some number of moves, Player 2 has no valid move (that is no move that forms a simple path with the sequence of previous moves).

**GEOGRAPHY** is the following problem:

- **INPUT:** a graph \(G\) and an initial vertex \(s\).
- **QUESTION:** does player 1 have a winning strategy for the game on \(G\) starting at \(s\) ?

Show that **GEOGRAPHY** is PSPACE-complete.

**Exercise 5: Language theory**

Show that the following problems are PSPACE-complete:

1. **NFA Universality:**
   - **INPUT:** a non-deterministic automaton \(A\) over alphabet \(\Sigma\)
   - **QUESTION:** \(L(A) = \Sigma^*\) ?
   
   Bonus: what is the complexity of this problem for a DFA ?

2. **NFA Equivalence**
   - **INPUT:** two non-deterministic automata \(A_1\) and \(A_2\) over the same alphabet \(\Sigma\)
   - **QUESTION:** \(L(A_1) = L(A_2)\)
   
   Bonus: what is the complexity of this problem for a DFA ?

3. **DFA Intersection Vacuity:**
   - **INPUT:** deterministic automata \(A_1, \ldots, A_m\) for some \(m\)
   - **QUESTION:** \(\bigcap_{i=1}^m L(A_i) = \emptyset\) ?

**Exercise 6: A translation result**

Show that if \(P = \text{PSPACE}\), then \(\text{EXPTIME} = \text{EXPSPACE}\).

**Exercise 7: Descriptive complexity**

adapted from an exercise by Cristina Sirangelo

1. Let **FO-SAT** be the following problem:
   - **INPUT:** a first-order formula \(\phi\)
   - **QUESTION:** is \(\phi\) satisfiable? That is, does \(\phi\) have a model \(M\) (denoted \(M \models \phi\))?
What is the complexity of this problem?

2. Define FO-Combined Complexity:
   - INPUT: a first-order formula $\phi$ and a finite structure $M$ (on the same signature)
   - QUESTION: does $M \models \phi$?

   Show that this problem is \textit{PSPACE}-complete.

3. What if the model is fixed? That is, for a fixed model $M$, one want to decide:
   - INPUT: a first-order formula $\phi$
   - QUESTION: does $M \models \phi$?

4. Given a first-order formula $\phi$, Data Complexity is the problem:
   - INPUT: a model $M$
   - QUESTION: does $M \models \phi$?

   Show that this problem is in L.