# Complexité avancée
# TD 3

Cristina Sirangelo - LSV, ENS-Cachan

October 8, 2014

**Exercise 1. Geography game** Principle :

- The game starts with a given name of a city, for instance *Cachan* ;
- the first player gives the name of a city whose first letter coincides with the last letter of the previous city, for instance *Nanterre* ;
- the second player gives then another city name, always starting with the last letter of the previous city, for instance *Evian* ;
- 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).

1. The geography game problem ($JG$) is the problem of checking, given an input arbitrary graph and initial vertex, whether the first player has a winning strategy for the above mentioned set of rules (i.e. a strategy that allows Player 1 to win, no matter the moves of Player 2).
   Show that $JG$ is in **PSPACE** by describing a polynomial space Turing machine for it.
2. Show that $JG$ is **PSPACE**-complete.

**Exercise 2. NFA universality** Show that the problem of deciding – given a non-deterministic finite state automaton $A$ over alphabet $\Sigma$ – whether the language accepted by $A$ is $\Sigma^*$, is **PSPACE**-complete.

**Exercise 3. NFA equivalence** Show that the problem of deciding – given two non-deterministic finite state automata $A_1$ and $A_2$ over alphabet $\Sigma$ – whether $L(A_1) = L(A_2)$, is **PSPACE**-complete.

**Exercise 4. Intersection of deterministic finite state automata** Show that the problem of deciding $\bigcap_{i=1}^{m} L(A_i) = \emptyset$ for a given set of deterministic finite state automata, $A_i, i = 1..m$, is **PSPACE**-complete.

**Exercise 5. Non-trivial FO theories** Recall that the FO-theory of an FO-structure $\mathbf{A}$ of vocabulary $\sigma$ is the set of all FO $\sigma$-sentences which are satisfied by $A$. The *decision problem* for an FO theory is to determine whether a given sentence $\varphi$ belongs to the theory (i.e to determine whether $\mathbf{A} \models \varphi$).

Come up with a definition of *trivial* for FO theories, in terms of constraints on the structure $\mathbf{A}$. Prove that for your definition of trivial :

1. The decision problem of every non-trivial theory is **PSPACE**-hard.
2. The decision problem of every non-trivial theory of a finite structure is **PSPACE**-complete.
3. The decision problem of every trivial theory is in **P**.

**Exercise 6. First-order theory of natural numbers with linear order** Determine the complexity of the decision problem for the FO-theory of the structure $(\omega, \leq)$, where $\omega$ is the set of natural numbers and $\leq$ is the usual linear order on $\omega$.

For the upper bound proceed as follows. For $k$-tuples $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$ of natural numbers, let $a_0 = b_0 = 0$ and define

$$a_1, \ldots, a_k \equiv_k^m b_1, \ldots b_k$$

if there exists a permutation $\pi : \{0, \ldots, k\} \to \{0, \ldots, k\}$ such that

- $a_{\pi(0)} \leq \cdots \leq a_{\pi(k)}$ and $b_{\pi(0)} \leq \cdots \leq b_{\pi(k)}$ and
- $min(2^m, a_{\pi(i+1)} - a_{\pi(i)}) = min(2^m, b_{\pi(i+1)} - b_{\pi(i)})$ for all $i = 0..k-1$

1. Prove the following lemma :

   **Lemma 0.1** *If $a_1, \ldots, a_k \equiv_k^m b_1, \ldots, b_k$ then for all $a_{k+1}$ there exists $b_{k+1}$ such that $a_1, \ldots, a_k, a_{k+1} \equiv_{k+1}^{m-1} b_1, \ldots, b_k, b_{k+1}$.*

2. Use the lemma above to show that if $a_1, \ldots, a_k \equiv_k^m b_1, \ldots, b_k$ and $\varphi(x_1, \ldots x_k)$ is a formula with $k$ free variables of quantifier rank $m$, then $\varphi(a_1, \ldots a_k)$ holds in $(\omega, \leq)$ iff $\varphi(b_1, \ldots b_k)$ does. (The quantifier rank of a formula is the depth of nesting of quantifiers ; a quantifier-free formula is of rank 0.)
3. Using 2. determine the complexity of checking whether a given FO sentence holds in $(\omega, \leq)$.