Complexité avancée - Homework 3

Benjamin Bordais
October 7, 2020
Due at 8.30 a.m., October 14, 2020

PSPACE and games The Geography game is played as follows:

- 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 *Nice*;
- the second player gives then another city name, also starting with the last letter of the previous city, for instance *Evry*;
- 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 directed 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 chooses 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 (i.e.: a path with no cycles), 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).

**Generalized Geography** (GG for short) is the following problem:

- **INPUT:** a directed graph \(G\) and an initial vertex \(s\).
- **QUESTION:** does Player 1 have a winning strategy for a GG game played on \(G\) from \(s\)?

1. Show that GG is in PSPACE.

2. Exhibit a logarithmic space reduction \(tr\) from QBF to GG. Carefully prove that the reduction is logspace and the equivalence “\(w \in \text{QBF} \iff tr(w) \in \text{GG}\)”.

3. What can you deduce about GG?
Solution:

1. We define a recursive function \( \operatorname{win} : (G, s, F) \mapsto \) “True iff the 1st player has a winning strategy on the graph \( G = (V, E) \), starting from \( s \in V \), for a version of the game where it is forbidden to play a vertex in \( F \subseteq V \).” This is equivalent to: “the 1st player has a winning strategy on the restriction \( G[V \setminus F] \).” Note that in the specification of \( \operatorname{win} \), it is required that \( s \notin F \).

Now \( \operatorname{win} \) is a simple recursive procedure:

\[
\operatorname{win}(G, s, F) = \exists (s, t) \in E \text{ s.t. } t \notin F' \land \operatorname{win}(G, t, F') \quad \text{for } F' = F \cup \{ s \}.
\]

An algorithm implementing this procedure explores every possible successors and calls itself recursively. It terminates since each call has a larger \( F \) and when \( F \) is equal to the whole set of vertices \( V \), it stops. Therefore, there can be at most \( |V| \) nested calls so the program will use a stack with a linear number of frames, each frame being of linear size (as one stores the set \( F \) and the current vertex \( s \)). This is polynomial space.

2. We construct a logarithmic space reduction \( \operatorname{tr} \) from an instance of QBF to an instance of GG so that \( \phi \in \text{QBF} \iff \operatorname{tr}(\phi) \in \text{GG} \). Consider a QBF formula \( \phi = Q_1x_1 \land Q_2x_2 \land \cdots \land Q_nx_n \cdot S \) where \( S \) is a propositional formula in CNF whose variable are in \( \{x_1, \ldots, x_n\} \) and \( Q_i \in \{ \exists, \forall \} \) for all \( 1 \leq i \leq n \). We first translate \( \phi \) into an equivalent formula such that we have a strict alternation of existential and universal quantifiers (it may require to add dummy variables that do not appear in \( \phi \) ). We obtain \( \phi' = \exists y_1 \cdot \forall y_2 \cdots \exists y_{2k-1} \cdot \forall y_{2k} \cdot S \) for some \( k \leq n \). Let \( S = \forall_{1 \leq j \leq m}C_j \) with \( C_j = \land_{1 \leq i \leq a_j}l_{i,j} \) where \( l_{i,j} \) is a literal that is equal to \( y_i \) or \( \neg y_i \) for some \( 1 \leq i \leq 2k \).

We formally define \( \operatorname{tr}(\phi) \) as the graph \( G_\phi = (V, E) \) where \( V = \{ s_i, y_i, \neg y_i, s'_i \mid 1 \leq i \leq 2k \} \cup \{ C_j \mid 1 \leq j \leq m \} \cup \{ l_{i,j} \mid 1 \leq j \leq m, 1 \leq i \leq 3 \} \) and \( E = \{ (s_i, y_i), (s_i, \neg y_i), (y_i, s'_i), (\neg y_i, s'_i) \mid 1 \leq i \leq 2k \} \cup \{ (s'_i, s_{i+1}) \mid 1 \leq i \leq 2k - 1 \} \cup \{ (s'_{2k}, C_j) \mid 1 \leq j \leq m \} \cup \{ (C_j, l_{i,j}) \mid 1 \leq j \leq m, 1 \leq i \leq a_j \} \cup \{ (l_{i,j}, l_{i,j}) \mid 1 \leq j \leq m, 1 \leq i \leq a_j \} \). The initial vertex of the graph is \( s_1 \). Note that, for all \( i \), Player 1 plays in the vertices \( s_i \) and \( s'_i \) if and only if \( i \) is odd. It is also Player 1’s turn in vertex \( C_j \) for all \( 1 \leq j \leq m \).

![Figure 1: The graph \( G_\phi \) for \( \phi' = \exists x_1 \forall x_2 \cdots \forall x_{2k} \left( (x_3 \lor x_1 \lor \neg x_2) \land C_2 \land \cdots \land C_m \right) \).](image)

First note that this graph can be constructed in logarithmic space. Indeed, to do so, we need a fix number of pointers ranging over the variables and the clauses of the formula.

Now, let us prove that \( \phi \in \text{QBF} \) if and only if the first player has a winning strategy in the game of generalized geography \( G_\phi \). First, we define inductively the path
corresponding to a valuation for the first $0 \leq l \leq 2k$ variables: $p(\emptyset) = \epsilon$ and $p(\nu') = \nu \cdot \{y_i \rightarrow \bot\} = p(\nu) \cdot s_i \cdot a \cdot s_f$ with $a$ the vertex/literal either equal to $y_i$ or $-y_i$ ensuring $\nu' \models a$. Then, for a valuation $\nu$ of the first $1 \leq l \leq 2k$ variables, when we say that the game starts after the path $p(\nu)$ is seen, it means that the vertices in $p(\nu)$ are forbidden, the initial vertex is $s_{i+1}$ and it is Player 1’s turn if and only if $l+1$ is odd. In the case where $l = 2k$, then the game may start in any vertex $c_j$ (according to Player 2’s choice in $s'_{2k}$) at Player 1’s turn.

We prove by induction the following property on $1 \leq i \leq 2k + 1$, $P(i)$ : “for all valuation $\nu$ of the variables $y_1, \ldots, y_{i-1}$, we have $\nu \models Q_i y_i \cdots \forall y_{2k} \cdot S$ if and only if Player 1 wins in the game that starts after $p(\nu)$ is seen”.

Let us first prove $P(2k + 1)$. Consider a valuation $\nu$ of the variables. Assume $\nu$ satisfies $S$ and that the game starts at $c_j$ for some $1 \leq j \leq m$ after $p(\nu)$ is seen. Let us prove that Player 1 wins. Since $\nu \models S$, there exists a literal $l_{i,j}$ such that $\nu \models l_{i,j}$ for some $1 \leq i \leq a_j$. Assume that Player 1 chooses $l_{i,j}$ as next vertex. Then, the only possible successor is $l_{i,j}$ and by definition of $p(\nu)$, since $\nu \models l_{i,j}$, it is forbidden in the graph. Hence, Player 1 wins. Now, assume that Player 1 wins from $y_1 \cdots y_{i-1}$. We prove by induction the following property on $1 \leq i \leq 2k$ for all vertices $\{c_j \mid 1 \leq j \leq m\}$ after $p(\nu)$ is seen. Let us prove that $\nu \models S$. Consider a clause $C_j$ for some $1 \leq j \leq m$ and the choice made by Player 1 in vertex $c_j$. The vertex $l_{i,j}'$ is reached for some $1 \leq i \leq a_j$. Since Player 1 wins, it follows that the vertex $l_{i,j}$ is forbidden, otherwise Player 2 would chose it as next vertex and Player 1 loses since, in any case, the successor of $l_{i,j}$, that is $s'_{i,j}$, is forbidden. By definition of $p(\nu)$, the fact that the vertex $l_{i,j}$ is forbidden means that the valuation $\nu$ ensures $\nu \models l_{i,j}$. It follows that $\nu \models C_j$. As this holds for all $1 \leq j \leq 2k$, we have $\nu \models S$.

Let us now prove that $P(i+1) \Rightarrow P(i)$ for all $1 \leq i \leq 2k$. Assume $P(i+1)$ holds for some $i$. We deal with the case $i$ is odd, the other is analogous. Consider a valuation $\nu$ of the variables $y_1, \ldots, y_{i-1}$ and the formula $\exists y_i \cdots \forall y_{2k} \cdot S$. It is Player 1’s turn at vertex $s_i$. Assume Player 1 wins in the game after $p(\nu)$ is seen. Then consider his choice of next vertex $l = (\neg) y_i$. We consider the valuation $\nu' = \nu \cdot \{y_i \rightarrow \bot\}$ such that $\nu' \models l_i$ (note that this is possible since the valuation $\nu$ does not deal with variable $y_i$). Then, since Player 1 wins after $p(\nu)$ is seen, he also wins after $p(\nu')$ is seen as $p(\nu')$ extends $p(\nu)$ by following his choice. Then, by $P(i+1)$, $\nu' \models \forall y_{i+1} \cdots \forall y_{2k} \cdot S$. It follows that $\nu \models \exists y_i \cdots \forall y_{2k} \cdot S$. Now, assume that $\nu \models \exists y_i \cdots \forall y_{2k} \cdot S$ and let us prove that Player 1 wins in the game after $p(\nu)$ is seen. By definition of the semantics of the satisfiability of a quantified formula, $\nu \models \exists y_i \cdots \forall y_{2k} \cdot S$ means that there exists $\nu'$ extending $\nu$ to variable $y_i$ such that $\nu' \models \forall y_{i+1} \cdots \forall y_{2k} \cdot S$. Then, by $P(i+1)$, we have that Player 1 wins in the game after $p(\nu')$ is seen. Hence, in $s_i$, Player 1 can choose the next vertex so that it mimics the final part of the path $p(\nu')$. Thus, since Player 1 wins after $p(\nu')$ is seen, Player 1 also wins after $p(\nu)$ is seen. Overall, $P(i)$ holds.

We can conclude that $P(1)$ holds, which exactly corresponds to the equivalence $\phi \in \text{QBF} \Leftrightarrow \text{tr}(\phi) \in \text{GG}$.

3. **QBF** being **PSPACE**-complete for logarithmic space reductions, and since the composition of logspace reduction can be done in logspace, it follows that **GG** is **PSPACE**-hard. As it is also in **PSPACE** by question 1, we can deduce that **GG** is **PSPACE**-complete.