6.5440: ALGORITHMIC LOWER BOUNDS, FALL 2023
Prof. Erik Demaine, Josh Brunner, Lily Chung, Jenny Diomidova

Problem Set 7 Solution

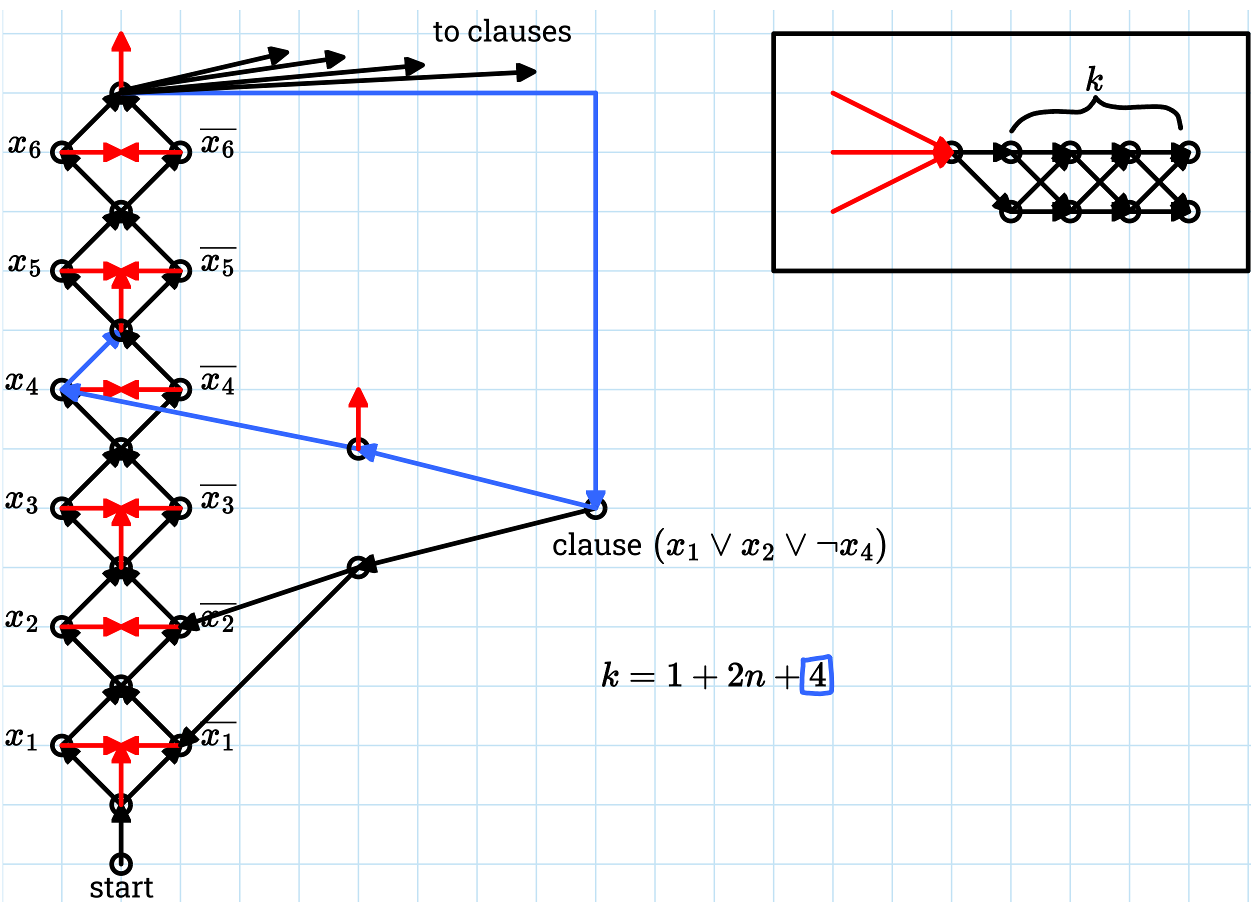
Due: Monday, October 30, 2023 at noon

Problem 7.1 [Graph Runner: Catching Games].

Graph Runner is a two-player game played on a directed graph $G = (V, E)$. Player 1 has a token, starting at a given vertex $s \in V$. During Player 1's turn, they can move the token along an outgoing directed edge to an adjacent vertex, and gain 1 point; if they cannot move, the game ends. During Player 2's turn, they can delete an outgoing edge from the node the token is currently on. Player 1's goal is to maximize their score, while Player 2's goal is to minimize it. Prove that it is PSPACE-complete to determine, given (G, s, k), whether Player 1 can achieve a score of at least k.

You must include a drawing or diagram in your submission.

Solution:



We reduce from QSAT; see the figure. Red edges lead to a location from which Player 1 can always win; so Player 2 must delete those edges whenever they are reached. Let the QSAT instance be $\forall x_1 : \exists x_2 : \forall x_3 : \cdots : \exists x_n : \phi(x_1, \ldots, x_n)$ where ϕ is a 3CNF formula. We create n diamond-shaped variable gadgets corresponding to the n variables. Universally quantified variable gadgets allow Player 1 to choose which side of the diamond to take, while existentially quantified variable gadgets allow Player 2 to make this choice. Finally, we create a vertex for each clause, and allow Player 1 to choose a clause. Then Player 2 chooses the negation of a literal of that clause, moving the token to that side of the corresponding variable gadget. By setting $k = 2n + 5$ we can ensure that Player 1 wins the game exactly when the chosen side of the variable gadget has not been visited already, since if it was visited then Player 2 will have already deleted the red edge, so they can delete the remaining edge and end the game just in time. Thus Player 1 wins if and only if there is an unsatisfied clause, which occurs if and only if the QSAT instance was false. Therefore the problem is complete for coPSPACE = PSPACE.

As a note, the decision problem which asks "Is it possible for Player 1 to go from vertex s to vertex t?" (with no mention of points) is solvable in polynomial time, even if the graph is a multigraph. The idea is to set $T \leftarrow \{t\}$ and then repeatedly add to T any vertex v which has two outgoing edges that both point into T. It can be checked that Player 1 wins if and only if $s \in T$ at the end of this process.