

## Problem Set 4

*Due: Wednesday, November 12th, 2014*

**Problem 1.** Given a graph  $G = (V, E)$ , a connected dominating set  $D \subseteq V$  is a set of vertices such that the subgraph of  $G$  induced by  $D$  is connected, and each vertex in  $V$  is adjacent to at least one vertex in  $D$ . The Connected Dominating Set problem takes as input a graph  $G = (V, E)$  and a positive integer  $k$ , and asks whether there exists a connected dominating set of size  $k$ .

- (a) Show that Connected Dominating Set is  $W[2]$ -hard with respect to the parameter  $k$ .
- (b) Show that Connected Dominating Set is in  $W[2]$ . **Hint:** Construct a weft-2 circuit of size  $f(k)n^{O(1)}$  with  $nk$  inputs.

**Problem 2.** Recall the problem of Minesweeper. A *Minesweeper board* consists of a rectangular grid with dimensions  $m \times n$ , a subset  $R \subseteq [1, m] \times [1, n]$  of revealed squares in the grid, and a function  $f : R \rightarrow [0, 8]$  mapping from cells in  $R$  to the number of adjacent bombs. A *solution* to a Minesweeper board  $\langle m, n, R, f \rangle$  is a set  $X \subseteq ([1, m] \times [1, n]) \setminus R$  of bomb positions such that, for every square  $(i, j) \in R$ , the number of squares adjacent to  $(i, j)$  that are contained in  $X$  is  $f(i, j)$ . The Minesweeper problem is the obvious NP search problem: find a solution to a given Minesweeper board. The #Minesweeper problem is the corresponding problem of counting solutions.

In each part of this problem, we give a reduction from Planar Circuit SAT to Minesweeper (slight modifications from those in lecture), intended to show that #Minesweeper is #P-hard. For each reduction, do one of the following: prove that it is parsimonious, prove that it is  $c$ -monious, or show that the number of solutions does not change by a fixed multiplicative factor (and thus that the reduction does not yield the desired #P-hardness results).

- (i) In this reduction, data is conveyed by a wire like the one depicted in Figure 1(a). By construction, if there is a bomb in the leftmost empty square, then there cannot be a bomb in the rightmost empty square. If the wire is oriented from left to right, this case represents having the value TRUE conveyed along the wire. Similarly, if there is no bomb in the leftmost empty square, then there must be a bomb in the rightmost empty square. If the wire is oriented from left to right, this case represents having the value FALSE conveyed along the wire.

Wires can be terminated using one of two gadgets. The gadget in Figure 1(b) is a terminal that allows the incoming wire to carry either value. The gadget in Figure 1(c) is a terminal that forces the incoming wire to carry the value TRUE (and is used only once, to ensure that the output of the simulated circuit is TRUE). The shifter gadget depicted in Figure 1(d) can be used to adjust the length of the wire modulo 3.

The gadget in Figure 1(e) can be used for several purposes. For this gadget, if the leftmost empty square contains a bomb, then the topmost and bottommost squares cannot contain a bomb (thus creating the value of TRUE on the wires if the top wire is oriented upwards and the bottom wire is oriented downwards), and the rightmost square must contain a bomb (thus creating the value of FALSE on the right wire, if it's oriented to the right). Hence, by blocking off the top and bottom wires with a terminator, this gadget can be used to negate

a value incoming from the left. By blocking off the top and right wires with a terminator, this gadget can be used to turn. By blocking off the right wire with a terminator, this gadget can be used to split a value coming in from the left. Thus, this one gadget gives us splitters, turns, and negations.

The final gadget is an AND gate, shown in Figure 1(f). The two inputs are on the left, coming in from the top and bottom; the output is on the right.

- (ii) The wires for this reduction, depicted in Figure 2(a), are the same as the wires used in part (i). The terminator gadgets are set up similarly to those in part (i), with the terminator gadget in Figure 2(b) allowing the wire to take on any value, and the terminator gadget in Figure 2(c) forcing the incoming wire to be TRUE. However, instead of using a shifter to adjust the length of wires modulo 3, the shifter in this reduction is used to move the wire slightly in the direction orthogonal to the wire, as shown in Figure 2(d).

As in part (i), we have a combination splitter/negation/turn gadget, depicted in Figure 2(e). For this gadget, if there is a bomb on the leftmost empty square, then there is also a bomb on the rightmost empty square and the bottommost empty square, so if a wire is coming in from the left, then the negated value will emerge from the wire on the right and the wire on the bottom. By terminating the wire on the bottom, this can be converted into a negation gadget. By terminating the wire on the right and adding a copy of the negation gadget, this can be used to turn. By adding a copy of the negation gadget to negate the input value, this becomes a splitter.

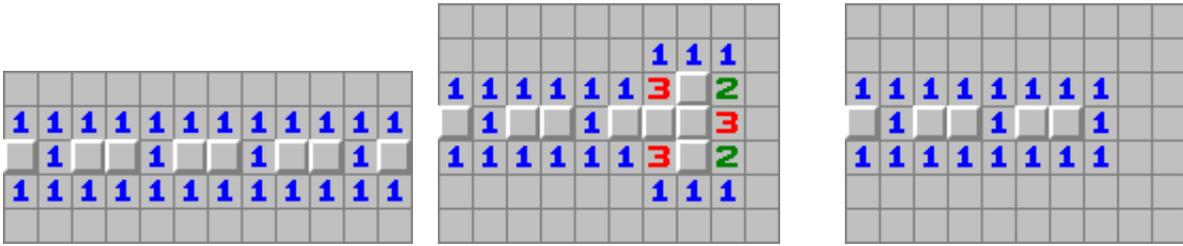
The final gadget is an OR gate, shown in Figure 2(f). The two inputs come in from the left and from the top; the right is the output.

**Problem 3.** Given a simple directed graph  $G = (V, E)$ , a set of edges  $E' \subseteq E$  is *cycle-removing* if  $G' = (V, E \setminus E')$  contains no directed cycles. Two cycle-removing sets are *neighbors* if one set can be obtained from the other by removing a single edge (or, symmetrically, by adding a single edge).

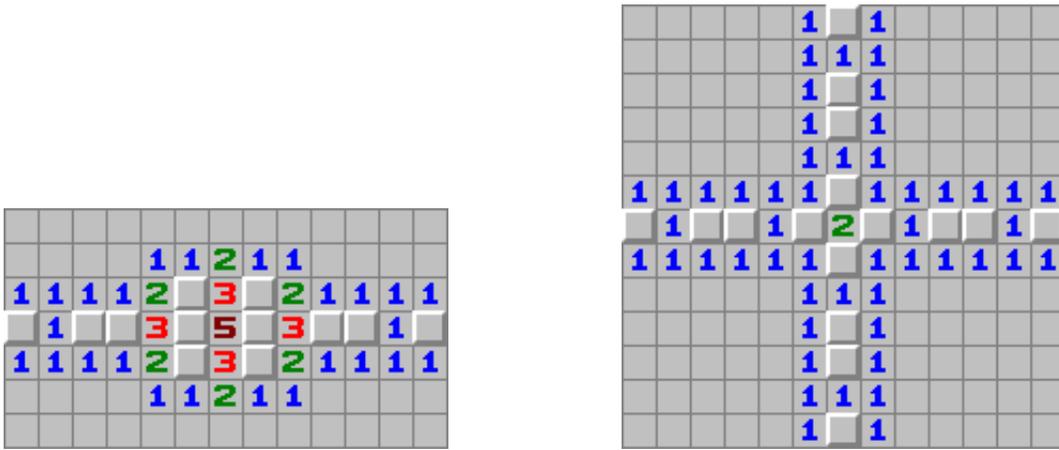
Suppose that you are given a graph  $G = (V, E)$ , an integer  $k$ , and two cycle-removing sets of size  $k$ :  $E_{start}$  and  $E_{end}$ . Call a set of edges  $E' \subseteq E$  *valid* if it is a cycle-removing set of size  $k$  or  $k + 1$ . Show that it is PSPACE-hard to determine whether there exists a path of valid sets  $E_{start} = E_0, E_1, E_2, \dots, E_m = E_{end}$  such that  $E_i$  and  $E_{i+1}$  are neighbors for all  $0 \leq i < m$ .

**Problem 4.** *Expansion* is a two-player game played on a simple, connected graph. Each node is a different ‘territory’, which can be in one of three states: unowned, owned by Player 1, or owned by Player 2. Initially, each player owns one territory; the remaining territories are unowned. Players take turns claiming ownership of an unowned territory. Each unowned territory  $v$  has a number  $k_v$  which dictates how many adjacent territories a player must own in order to claim  $v$ . Play proceeds alternately until neither player can claim ownership of a territory. The player with the most owned territories at the end of the game wins. Prove that deciding whether Player 1 can win in this game is PSPACE-complete.

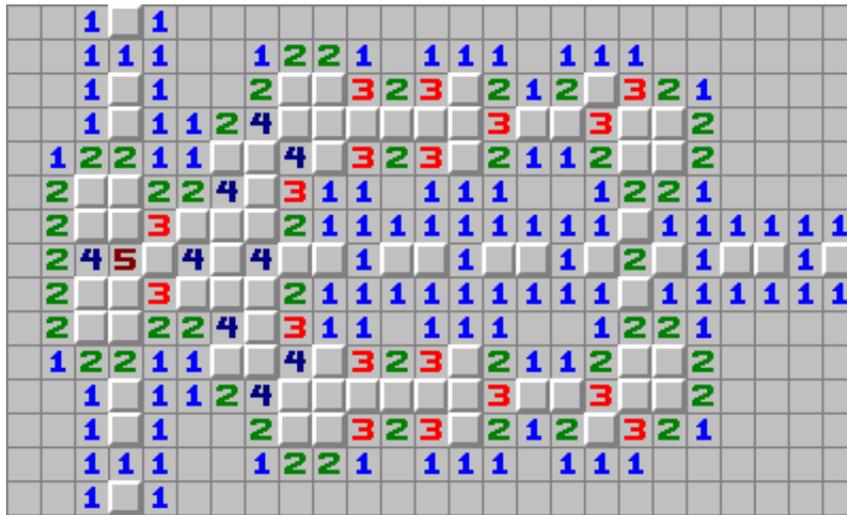
**Problem 5.** *Graph Runner* is a game played on a simple, connected, directed graph  $G = (V, E)$  with two players. Player 1 has a token on vertex  $v_1$ , the starting vertex. During their turn, Player 1 can move the token along an outgoing directed edge to an adjacent vertex, and gain 1 point; if 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. Prove that it is PSPACE-complete to determine whether Player 1 can achieve a score of at least  $k$ , where  $k$  is polynomial in  $|E|$ .



(a) A wire. (b) Normal terminator. (c) Terminator forcing TRUE.



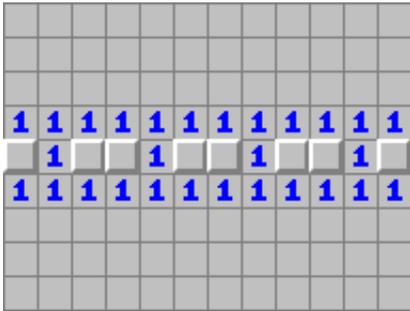
(d) Wire shifter. (e) Splitter gadget.



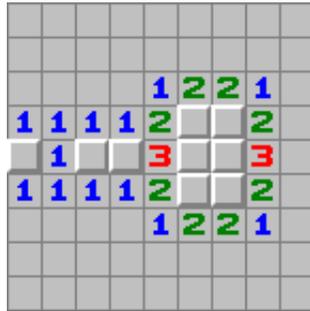
(f) AND gadget.

Figure 1: The gadgets used in the Planar Circuit SAT to Minesweeper reduction in Problem 2(i).

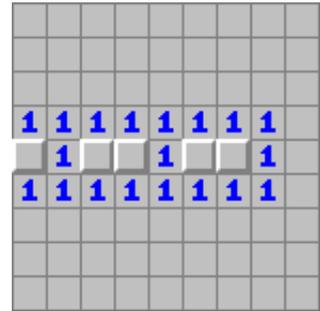
© Microsoft. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/fairuse>.



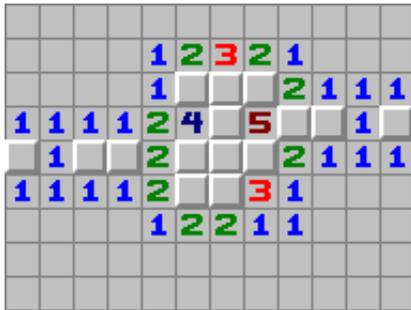
(a) A wire.



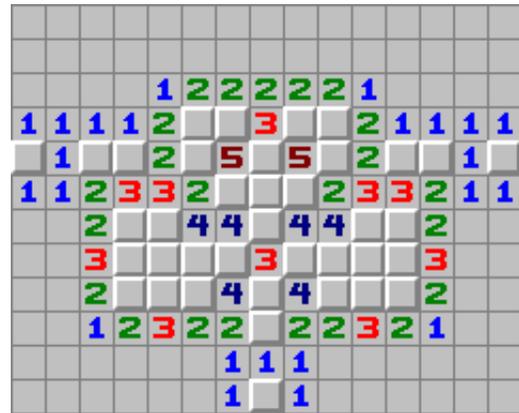
(b) Normal terminator.



(c) Terminator forcing TRUE.



(d) Wire shifter.



(e) Splitter gadget.



(f) OR gadget.

Figure 2: The gadgets used in the Planar Circuit SAT to Minesweeper reduction in Problem 2(ii).

O IFURVRIW \$ ØWJ KW UHVHUYHG 7KL V FROMQWLV H[ FOXGHG IURP RXU&UHDWYH  
&RP P ROV ØFHOVH )RUP RUH LQIRUP DWRO VHH KWS RFZ P UWHGX IDLXVH

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.890 Algorithmic Lower Bounds: Fun with Hardness Proofs  
Fall 2014

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.