6.890: **Algorithmic Lower Bounds: Fun With Hardness Proofs** Fall 2014

Lecture 19 Scribe Notes

Prof. Erik Demaine

1 Overview

Today, we'll cover a few more 2 player games with polynomially bounded moves by reducing from Bounded 2CL, which is PSPACE-complete. Afterwards, we'll move on to unbounded 2 player games, which are generally EXPTIME-complete. Lastly, we look at some even harder variants of unbounded 2-player games, that are EXPSPACE or even 2EXPTIME-complete.

2 Amazons

Amazons is played by two players - black and white - with an equal number of black and white queens on a chess board. Each turn, a player must make a "queen move" with one of his queens, and then shoots an arrow onto any square reachable by a "queen move" from the new position of the queen. Here, a queen move is any non-zero move in a straight line diagonally, horizontally, or vertically. Queens may not move through or shoot through squares occupied by arrows or other queens. The player who is able to move last wins.

We have a polynomial number of moves because every turn, a square on the board is consumed. So, we can show that Amazons is PSPACE-complete by reduction from Bounded 2CL.

Our wire gadget is a single corridor of queens. Activating a gadget for white is equivalent to moving a queen one square down and shooting an arrow one square down. This leaves enough space for the next queen to activate, propagating the signal. Diagonally offsetting two wires gives a shift, which we can use to fix any parity issues. We can also get a turn gadget, as shown in Figure 1. Signal can only flow one way through this turn gadget.

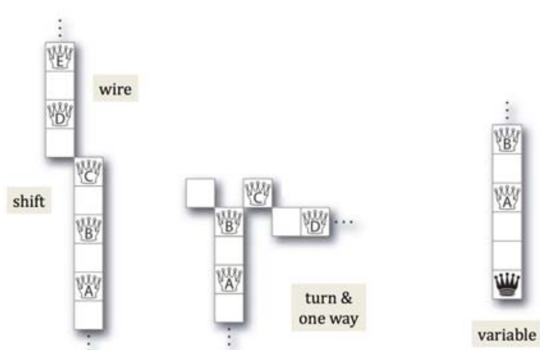

Figure 1: From left to right, a wire and shift gadget, a turn and one-way gadget, and a variable gadget.

Our variable gadget consists of a black queen at the end of a wire. If black moves his queen first, white can no longer activate the wire. If white moves first, he activates the wire. Players alternate picking variables initially.

The victory gadget (Figure 2) for white consists of two large rooms, where white can only traverse both rooms if the gadget is activated. Black is given a single movable piece in a very large area, and the rooms are sized so that white can only outlast black if he can reach both rooms. If white can only reach one, he loses.

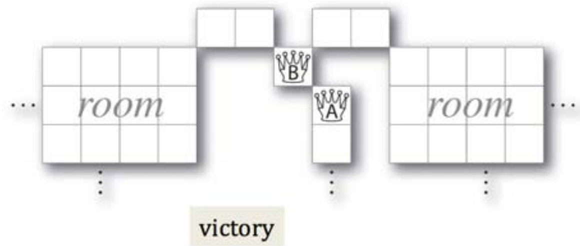

Figure 2: An Amazons victory gadget. White can only access both rooms by moving A down and shooting downward. Then White, moves B up and left, shooting an arrow where A was, then B left, shooting into the left room, and finally traversing the left room before moving through the corridor into the right room.

We can create an AND gadget, where D can only activate if both B and A are activated, allowing C to move down and shoot diagonally. Adding an extra free square between A, B, and C results in an OR gadget (where A and B are inputs) that doubles as a CHOICE gadget (where either A or B is the entrance). Finally, we can make the last gadget we need - a SPLIT gadget - by taking advantage of diagonal movement. This completes the reduction from Bounded 2CL, proving Amazons is PSPACE-complete.

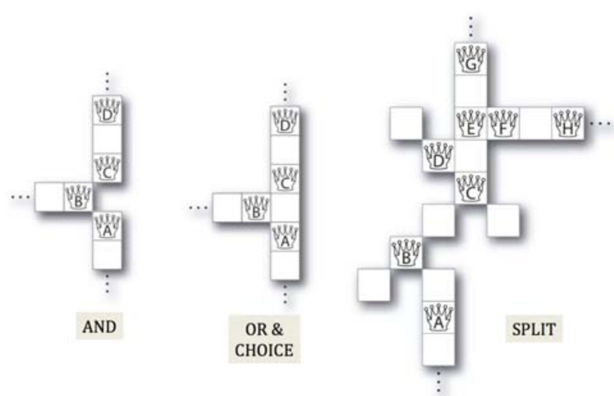

Figure 3: Three gadgets. AND: in order to activate, C must move into B and shoot into A. OR: C can move into the middle square and shoot into either A or B. SPLIT: A, B, and C move and shoot in similar directions, packing tightly. D moves all the way diagonally, and shoots up and left. E moves to where C was and shoots down and left. F moves down, left and shoots where D was. This allows G and H to both activate.

3 Konane

Konane is played with 2 colors of stones - black and white - on a board. Each turn, a player can perform with a single piece, 1 or more jumps over opponent pieces, as long as they all lie in a straight line. The jumped pieces are removed. The last player to move wins.

Again, moves are polynomially bounded as each move removes a piece and we show Konane is PSPACE-hard by reduction from Bounded 2CL.

Many of our gadgets, such as the turn gadget in Figure 4, will rely on the fact that a sandwiched white piece cannot be jumped. After the white enters the gadget and ends his turn, black cannot jump this piece.

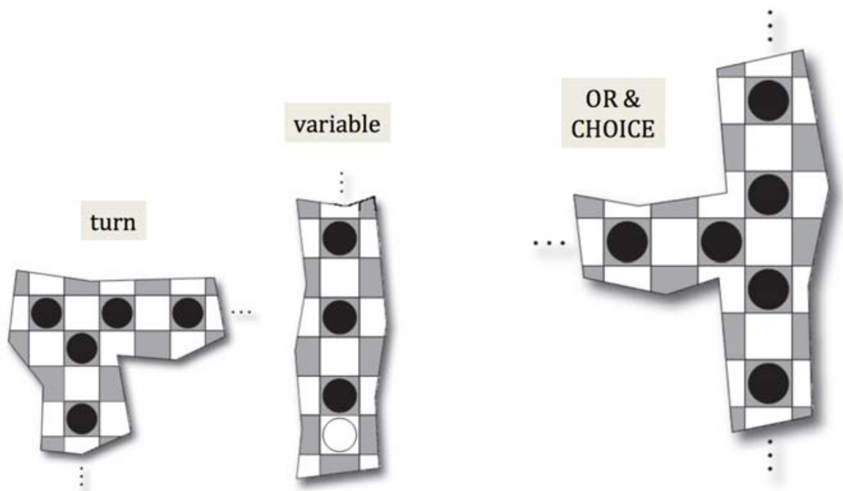

Figure 4: Turn, variable, and OR/choice gadgets. Turn: White's piece can move into a sandwiched position, where it cannot be captured. Variable: White's piece is vulnerable, but can avoid capture if he chooses to activate. OR/choice: White can either move straight up from the bottom, or move up or down from the left.

Our variable gadget consists of a white and black pieces next to each other. If white plays the variable first, his piece moves into a turn where it can wait safely. If black plays the variable first and takes white's piece then white can no longer activate this variable.

The OR gadget is a T shaped intersection. White can activate the top path if he activates either the left or the bottom path. Again, the OR can be reused as a choice if white enters from the left.

AND, SPLIT, and SHIFTs are all done through the same gadget. We see that we can activate output 2 only if input 1 activates first, followed by input 2. For ANDs, we treat output1 as garbage. For SPLITs and SHIFTs, we have a piece on standby along input 2 where it is sandwiched by black pieces. Once input 1 activates, this piece can also activate, letting us activate both output1 and output2.

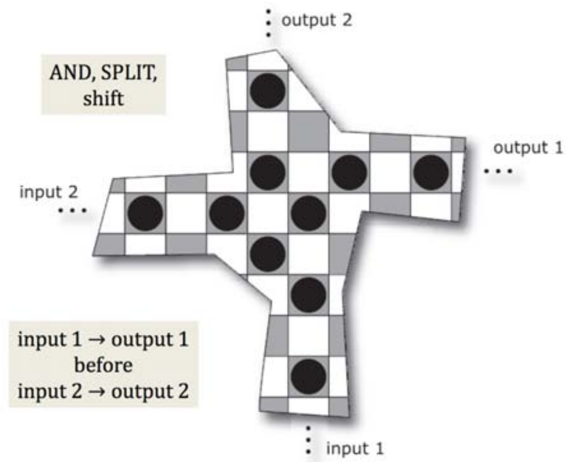

Figure 5: An AND/SPLIT/SHIFT gadget. Output2 can be activated by input2 only after output1 has been activated.

4 Cross Purposes

Cross Purposes is played with black and white stones on the intersection of a Go board. A black stone represents towers of two blocks, and a move consists of "pushing" a black stone over, resulting in two white stones either above, below, left, or right of the stone. The two players are named Vertical and Horizontal, with Vertical moving first. Vertical may only tip over a stone up or down, and Horizontal may only tip over the stone left or right, so long as the two spaces are unoccupied before hand. The last player to be able to move wins. An example of a couple moves in a game is given in Figure 6.

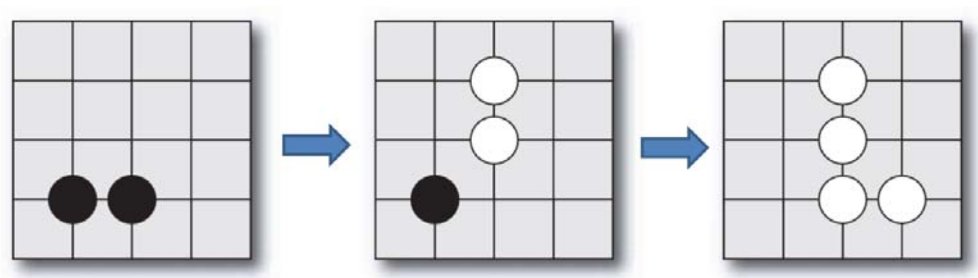

Figure 6: A couple of moves in Cross Purposes. First Vertical pushes up on the right black stone. Then Horizontal pushes right on the left black stone.

Again, the number of moves in this game is polynomially-bounded since each piece can be tipped at most once. So, the game is in PSPACE. As before, we show PSPACE-hardness by reducing from Bounded 2CL.

For our variable gadget, Vertical can activate the variable by pushing an initial tower downwards, creating space to push the next tower in the chain. However, if Horizontal chooses a variable first, he stops Vertical from activating the variable.

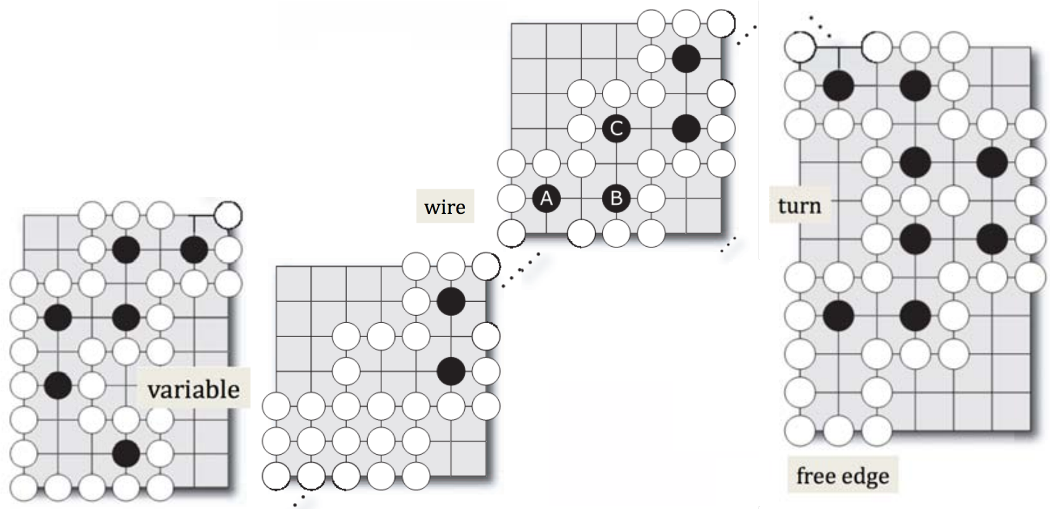

Figure 7: Variable, wire, turn, and free edge gadgets. Each chain is activated by alternating Vertical and Horizontal moves.

After setting variables, Horizontal is forced to assist Vertical in activating chains, as there will be no other available moves to play except the one opened by Vertical's last play. With this idea, we can obtain basic wire and turn gadgets, as in Figure 7.

We again need an OR gadget in Figure 8. If Vertical activates either input along the bottom, horizontal is forced to cooperate until vertical activates the output. However, if Vertical can activate the other side of the OR, then horizontal will no longer be able to play - hence the need for a protected OR. This can be constructed with the help of OR, AND, and choice gadgets as in Figure 9

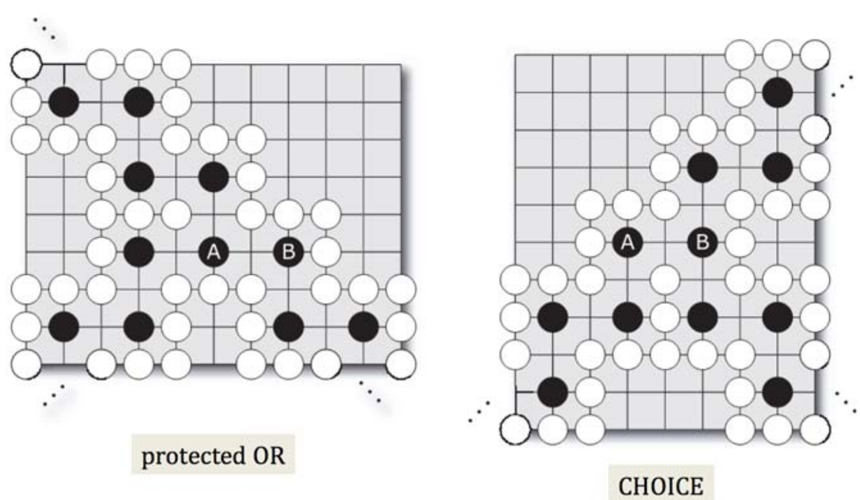

Figure 8: OR and Choice gadgets.

The Choice gadget is similar to the OR gadget: it is only rotated 90 degrees. Thus, when horizontal topples B, vertical gets to make the choice of whether the top or bottom channel will be activated.

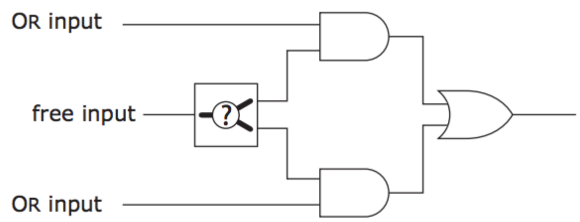

Figure 9: A protected OR made from OR, AND, and choice gadgets. Only one input to the OR can be activated, as only one of them can make it through the AND gadget.

Finally, as before, we have a multi-purpose AND, SPLIT, or shift gadget, where we terminate one output for an AND, and provide a free edge for one of the inputs for SPLIT and shift.

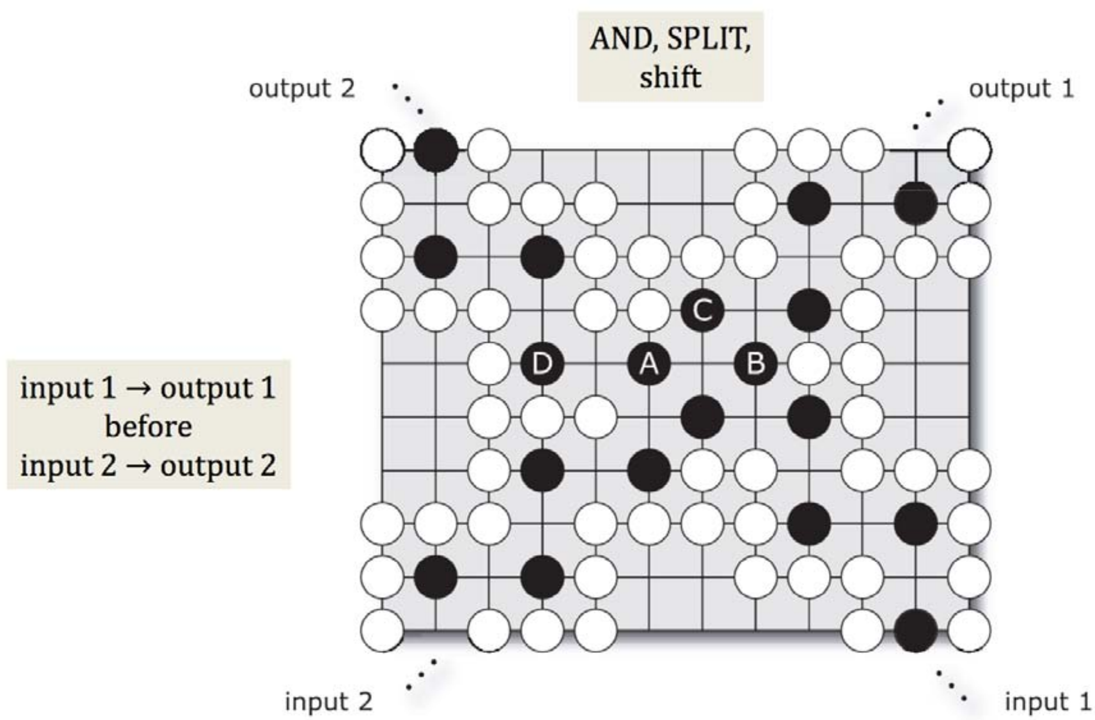

Figure 10: An AND, SPLIT, and shift gadget. If input 2 is activated before input 1, then when A is pushed down, Horizontal can push B instead of D, which stops the gadget.

5 Stochastic Games

In addition to standard 2 player games, we can also consider Stochastic games, where one of the players plays randomly. For Bounded 2-player stochastic games, we ask whether one player can force a win with a probability greater than $\frac{1}{2}$. This question is PSPACE-complete.

More formally, we have Stochastic SAT which asks

$$\exists x_1 : \mathcal{R}x_2 : \exists x_3 : \cdots : \Pr(Win) > \frac{1}{2}$$

where we use $\mathcal{R}x_2$ to denote a randomly chosen x_2. Interestingly replacing random choice with a for all choice seen in standard 2 player games doesn't change the hardness.

6 Unbounded 2 Player Games

We now move on to the class of 2 player games with an unbounded number of moves. This class of games is EXPTIME-complete [7]. An interesting thing about EXPTIME is that all games that are EXPTIME-complete automatically require exponential time, and we don't need to condition on whether $P = NP$.

In unbounded 2 player games, we start with an arbitrary variable assignment. All variables are partizan, i.e. each variable is either black or white and only the player of the corresponding color can touch it. These variables can then be set to 0 or 1 potentially infinitely many times. There is an additional "turn variable" whose value denotes which player's turn it is.

We have the following classes of Unbounded 2 player games, all EXPTIME-complete, defined by what constitutes a move and the victory condition:

- G1 - A move consists of setting all variables of your color, and then set the turn variable to 0 for player 2 or 1 for player 1. The first player to satisfy a 4DNF formula common to both players loses.
- G2 - A move consists of setting a single variable of your color. Passing is allowed. Both players have their own 12 DNF formula. The first player to satisfy their formula wins.
- G3 - A move consists of flipping a variable of your color. Passing is not allowed. Both players have their own 12 DNF formula. The first to satisfy their formula loses.
- G4 - A move consists of setting a single variable of your color. Passing is allowed. The first to satisfy a 13 DNF formula common to both players wins.
- G5 - A move consists of setting one variable of your color. Passing is allowed. Player 1 wins if anyone satisfies a common formula.
- G6 - the same as G5, but the formula must be in CNF.

6.1 Peek

Peek consists of a stack of plates with holes, where one hole is fixed. Each plate is either black or white and can be either in or out. Each turn, a player can move a plate of his color or pass. The player who gets a hole all the way through the plates wins. This is a direct reduction from G4.

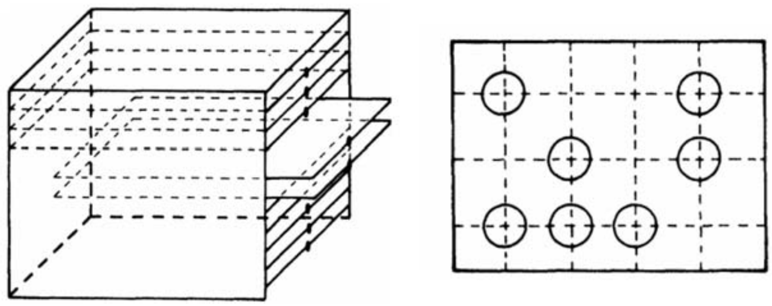

Figure 11: The game of peek. On the left, a stack of plates. On the right, an example of a plate.

7 Unbounded Graph Games

Here we present some unbounded graph games that are EXPTIME-complete [7].

7.1 HAM

In HAM, we start with a simple, undirected graph where each edge is colored either black or white. In addition to a color, each edge also has a state - in or out. Each turn, a player must toggle the state of an edge of his color. Player 1 wins if at any point in the game, the edges that are ?in? form a Hamiltonian cycle. This is EXPTIME-hard by reduction from G6.

7.2 BLOCK

Block is another graph game. We start with 3 graphs over the same vertices. Some of these vertices contain tokens that are either black or white. A vertex can have at most one token. Each turn, a player must slide a token of his color along any path in one of the 3 graphs, as long as the target vertex and all intermediate vertices on the path don't have any tokens. Each player i has some set of "victory vertices" W_i. If a player can move one of his tokens to a vertex in his set of victory vertices, he wins. We can reduce this from G3.

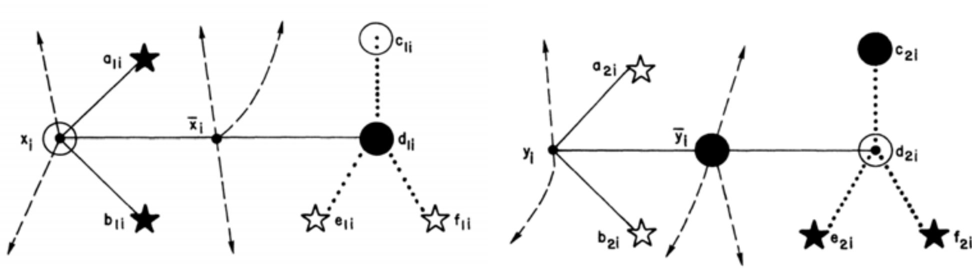

Figure 12: Variable gadgets for both players (white left, black right). Stars are winning vertices, and the dashed, dotted, and solid lines represent edges in each of the graphs. If either player deviates from setting variables, the other can instantly win.

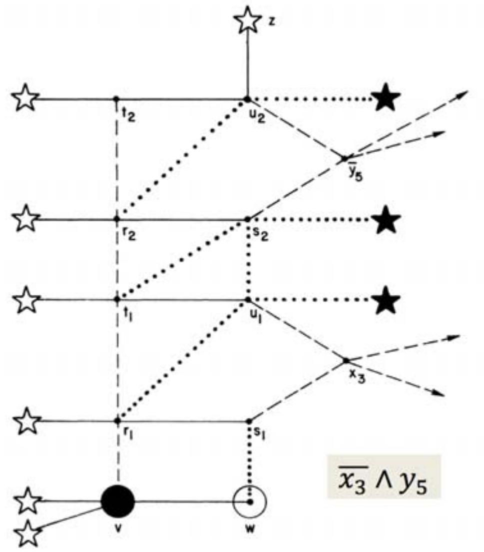

Figure 13: A gadget for the formula $\overline{x_3} \wedge y_5$. Once white activates by moving up, both black and white are forced to move up one at a time - or else the other player wins instantly. If x_3 and $\overline{y_5}$ are blocked, then white can't move up, in which case black wins and white should not have activated the formula.

8 Checkers, Chess, and Go

We can show that many real games are EXPTIME-complete by reduction from G3. These include Checkers [6], Chess [1], and Go [4]. These were all covered very briefly, and the images summarizing the proofs are in the slides. Additional results for slight variations on the ruleset are given later - and may make the games even harder than EXPTIME.

8.1 Checkers

The core of idea in this game is to use constant threats of losing to force player actions.

In checkers, we exploit the fact that a player must capture a piece when it is possible to. The board is split into 2 regions - an inner region where the clauses and variables are represented, and an outer region which can be triggered by a player to win (Figure 14).

Initially, players start by moving their own Kings between either a true or a false position. If at any point, a player satisfies their own DNF clause represented by pieces in the middle, the other player can activate an attack. A successful attack results in "free moves" - where the opponent has a configuration that requires a capture, but the other player does not. After accumulating enough free moves, the player can maneouvre his pieces in the outer spiral where the opponent is forced to jump into a position where all of his pieces in the spiral can be taken in one jump in the next turn. This gives enough of a material advantage to guarantee a win.

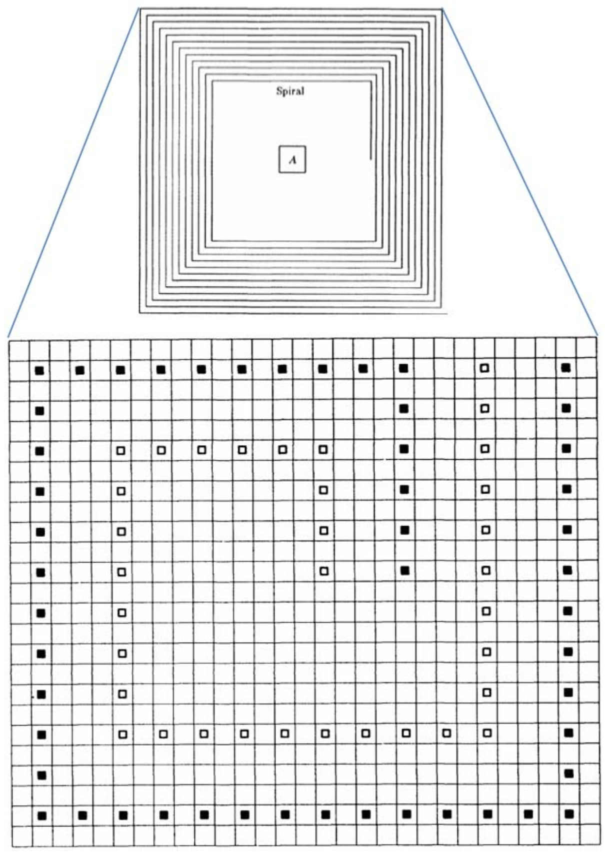

Figure 14: The spiral set up around the variables and clauses. With enough free moves, white can move into a position to destroy blacks spiral.

8.2 Chess

If there is no artificial move bound, this is also EXPTIME-complete for a variation using only pawns and bishops.

8.3 Go

With the Japanese ruleset, where positions are only not allowed to immediately repeat, Go is EXPTIME-complete using a reduction from G6. Here the upper bound is achieved since we only need to compare two states at a time.

9 Unbounded 2CL

In 2 Player Constraint Logic, we have each edge as either white or black. Each player has his own target edge which will make them win if they can flip it.

We reduce from G6. The basic idea will be that both players will sit around flipping variables. However, as soon as the formula is satisfied, white is able to lock the variable in pace by reversing the true/false edge. This forces black to respond with a fixed sequence of moves to prevent white from winning very quickly, giving white time to finish locking the variable into the formula. White can then lock the remaining variables in one go.

Once white began locking the variable though, black is able to start towards a slow win, where he is able to win in only 1 more move than it takes white to satisfy the formula and win. This prevents white from going back to flipping variables after locking. Basically, white will win if the states of the variables satisfy the formula. However, if he locks too early and the variables don't yet satisfy, black will be able to win through his slow win path.

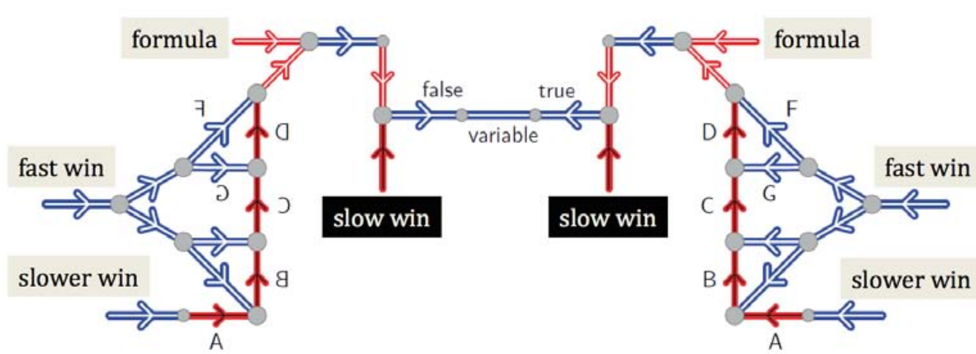

Figure 15: A variable gadget for white. For black variables, only the middle edge changes to black. Players are prevented from flipping edges early by using win paths. An even slower win sequence for white prevents black from flipping A early.

To finish the construction, we only require a path equalizer which slows edge traversal so all satisfying assignments take the same time, and a crossover gadget.

10 Some "Hard" Variants

For these unbounded 2 player games, there are some additional rules which can make the games even harder.

10.1 No-Repeat

This rule makes a player lose if they ever repeat a past game configuration. With the addition of this rule, G1, G2, and G3 become EXPSPACE-complete. So, Chess and Checkers also become EXPSPACE-complete. However, it's still an open problem whether Go with no-repeat (also known as the superko rules) is also EXPSPACE-complete. The intuition here is that we have to track and compare against all past game states.

10.2 Conditional No-Repeat

We introduce two special variables x and y. A player now loses if they repeat a past game configuration and at most 1 of x and y have changed since that configuration was played. Adding this rule makes G1 games 2EXPTIME-complete. Here, the intuition is that we have to track x and y state temporarily, in addition to all past game states.

10.3 Private Information games

In this variation, you can see some, but not all of an opponent's state. This makes G1 with 5DNF and G2 with DNF 2EXPTIME-complete. An example of this game is Peek with a partial barrier obscuring the state of some of each players plates to the other player.

10.4 Blind games

Here, Player 1's entire state is hidden from Player 2. This makes G2 with DNF EXPSPACE-complete. An example of this kind of game is Peek with a full barrier.

References

[1] A.S. Fraenkel and D. Lichtenstein. Computing a perfect strategy for $n \times n$ chess requires time exponential in n. *Journal of Combinatorial Theory (Series A)*, 31:199–214, 1981.

[2] Robert A. Hearn. Amazons, konane, and cross purposes are pspace-complete. *Games of No Chance 3*, 56:287–306, 2009.

[3] John H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984.

[4] J.M. Robson. The complexity of go. *Proceedings of the International Federation for Information Processing*, pages 413–417, 1983.

[5] J.M. Robson. Combinatorial games with exponential space complete decision problems. *Proc. Mathematical Foundations of Computer Science*, 176, 1984.

[6] J.M. Robson. N by n checkers is exptime complete. *Siam Journal on Computing*, 13:252–267, 1984.

[7] Larry J. Stockmeyer and Ashok K. Chandra. Provably difficult combinatorial games. *Siam Journal on Computing*, 8(2):151–174, May 1979.

6.890 Algorithmic Lower Bounds: Fun with Hardness Proofs
Fall 2014