COMP4141 Theory of Computation

Games

Ron van der Meyden

CSE, UNSW

Revision: 2013/05/12

(Credits: D Dill, K Engelhardt, M Sipser, W Thomas, T Wilke)

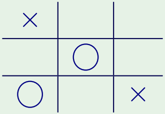
Games

Game theory nowadays plays an important role in economics, politics, and computer science.

Let's consider two-player turn-based games such as Tic-Tac-Toe, Checkers, Chess, Go, Geography Game, etc. only.

A *position* is a comprehensive description of the game state.

Example



is such a description for Tic-Tac-Toe (assuming that the player placing $\times$ always moves first).

Histories of moves are always usable as positions, however, they may be not concise enough for practical purposes.

Graphs and Winning Strategies

In general, the possible positions of (finite) games form directed graphs where the edges are sometimes labelled with the player who can move from the position at the source of the edge to the position at the destination.

When we use histories as positions then graphs become trees. In two-player games where players alternate, odd level tree moves belong to one player whereas even level moves belong to the other.

The leaves in such a tree describe final positions of the game. These leaves are marked depending on who wins the game when it ends in this position.

A player has a *winning strategy* in a position if, by choosing only his moves from that position onwards, he can force a win, regardless of the other player's moves.

Connection to Logic

The question "does player E have a winning strategy in position p of game G" can often be re-phrased as a quantified Boolean formula

$$\exists m_1 \, (\forall r_1 \, (\exists m_2 \, (\ldots \phi)))$$

where the m_i and r_i range over player E's, resp. A's moves and ϕ expresses that E wins.

Formula Game

Let's consider a simple formula game. Let ψ be a Boolean formula over propositions $x_1, \ldots, x_k$. In move i a player chooses a truth value for x_i. Player E takes the odd turns and player A takes the even turns. Player E wins if the formula turns out true, otherwise A wins.

Player E has a winning strategy iff the quantified Boolean formula

$$\exists x_1 \, (\forall x_2 \, (\exists x_r \, (\ldots \psi)))$$

is true.

Formula Game cont.

Since it is pretty much the same as *QBF* it follows that

$$\{ \langle\psi\rangle \mid E \text{ has a winning strategy for } \psi \}$$

is also **PSPACE**-complete.

Sipser puts the quantifiers into ψ but this doesn't make a difference because we assume an order on the propositions and from that the quantifier sequence follows.

Geography

Sipser writes that kids play a game in which one starts with a city name and the next kid always has to come up with a *new* city name that starts with the last letter of the previous one. The kid who cannot name another city (not previously named) in this sequence loses.

We could re-phrase this as a graph problem by making all the known city names nodes and have edges from a name wa to bv iff $a = b$.

Or we could generalise this even further to an arbitrary directed graph and forget about the city names.

$$GG = \{\ \langle G, b \rangle \mid \text{Kid 1 has a w. s. in } G \text{ starting at } b\ \}$$

Geography cont.

Theorem

GG is **PSPACE***-complete.*

Proof of GG $\in$ PSPACE.

That GG is in **PSPACE** follows from having a polynomial space TM M which decides GG. On input $\langle (V, E), b\rangle$:

1. Reject if b has no successor nodes in (V, E) because kid 1 loses.
2. Remove b from (V, E): let $V_1 = V \setminus \{b\}$ and $E_1 = E \cap (V_1 \times V_1)$.
3. For each b_1 such that $(b, b_1) \in E$ run M on $\langle (V_1, E_1), b_1\rangle$.
4. If all of these recursive calls accept, then kid 2 has a winning strategy in the game so reject, otherwise accept.

At each point we only need to store the description of the game (i.e. the input) and the stack of crossed out nodes. □

Geography cont.

Proof of GG is PSPACE-hard.

by reduction from *QBF*. □

Real Games

Real games such as Chess and Go have fixed board sizes and thus finite numbers of positions. Therefore the **PSPACE** arguments don't really apply.

But we can generalize these games to $n \times n$ boards and get results like:

Theorem

The problem of deciding if a given $n \times n$ GO position is winning is PSPACE-hard.

(GO is PSPACE-hard, Lichtenstein & Sipser, JACM 1980)

—THE END—