Announcement

Traditional competition "Semicolon"takes place on 9th May in the afternoon.

Games Programming

- Combinatorial game is a game of two players. State of the game is given by position of particular items. All items relevant to the game are visible for both players. I.e., combinatorial games are games with full information.
- Examples: Nim, Strange game, Draughts, Chess, Halma, Nine Men's Morris, Poisoned chocolate, Devilish darts,...
- Combinatorial games are *NOT*: Poker, Mau mau, Black jack, formula-race, Doom,...
- We focus on the playing algorithms (not on input/output).
- We expect the players to behave rationally (i.e., they want to win).

Shannon's theorem

Theorem (Shannon)

Each combinatorial game (with finite number of possible moves) has a winning strategy for at least one of the players.

Důkaz.

Sketch: Either at least one of the players may enfoce the game to start cycling (to never finish). Then he never loses and theorem holds. Or the game is finite and we examine predicates:
There exists our (1st player's) move s.t. for all moves of 2nd player there exists our move s.t.,... we win.
For all our moves there exists a move of the 2nd player s.t. for all our moves... we do not win.
Predicates are negation one to another and they are finite (finite quantif.), thus they are decidable and one of them holds. □

Game-graph

- For a game (its instance) we assign an oriented graph:
- Vertices represent states of the game,
- edges represent possibility of transition between states.
- Example for Nim with 1 or 2 matches (on white-board).
- Each state (vertex) may be colored according to who wins (when starting here).

Examples of graphs

- The game is represented by an oriented graph and we are moving a coin over this graph starting in a given vertex.
- We should reach one of terminal vertices; who cannot move, loses (who reaches that state, wins).
- Graph of the game is given, questionable is how to win.
- Note that each game can be represented as Devilish darts.
- Strange game: vertices are individual squares of the board.
- It is enough to say whether the player moving from current vertex wins or loses (or if there is a cycle that both players appreciate).

Game-tree

- We may split a game-graph into a game-tree (if a game is finite).
- Advantage: simpler structure (than game-graph),
- disadvantage: size.
- Example: Nim for 5 matches.

AND-OR-tree

- Considering a game-graph, we may create game-tree instead by splitting one state into possibly more states.
- For this tree we may ask whether there is a branch where we win.
- Such a "branch" is a subtree, s.t., after an odd step for all possibilities (even step) there is an odd step,..., i.e.,...
- either we win in the first son, or in the second, or in the third,...
- while the other player loses in the first (grand-)son and in the second son and in the third,...
- AND- and OR-gates are regularly interlacing, thus AND-OR-tree.

Games with evaluation

Definition

Game with an evaluation is a game where the result of a game is a value. One player tries to maximize this number while the other is trying to minimize it.

Definition

Game with a zero sum is a game where win for one player is a loss for the other player (and these values are the same).

Examples

- **Trip to New York:** With a girl-friend we want to make a trip to New York. While we want to visit as many pubs and technical sights (as possible), she wants to visit as many hair-dressers and museums. Thus we negotiate that on each crossing we will alternate in making decisions. How should we make the decisions?
- **Matrix-traversal:** Given a matrix, we start on the first row. Among a given row the player picks a column. The value in the matrix says us how many he wins. The other player for a picked column picks a row (and the coordinates tell us how much he wins). And for the chosen row the first player continues with picking a column...
- Common question is how to play it.

Algorithm Minimax

- This algorithm gives us a lower bound for the win (if we maximize).
- Can be used for games with evaluation.
- We build the game-tree and evaluate.
- We start from the leaves and based on who playes we either pick minimum or maximum from all the possibilities.
- Note that we generalized AND-OR-tree.

Algorithm Negamax

- Applicable for games with zero sum.
- Exploits the fact that

$$\operatorname{ind}\max_{i \in S} f(i) = \operatorname{ind}\min_{i \in S} -f(i).$$

- It does the same as Minimax, just it is simpler for programming.

Heuristics

- Usually it takes a long time to analyze the game.
- α-β-pruning: If for some son S we may win at least α, why should we analyze further a son T when we found a branch where the minimizer can force us below α?
- For the other player we use β: If the other player can force us below β, why should he even permit us to reach the state from which we may win more than β?

Real games

- We may build the game-tree.
- Unfortunately, this tree is too large and even α-β-pruning does not help much.
- So we define **static evaluating function.** This function evaluates a given position based on experience.
- Thus we define a **horizon** (depth of the game-tree we want to analyze) and when reaching the horizon before the final state, we use the static evaluating function.
- In Chess we may count the difference in material and yet reflect stones that are threatened by the other player (like Colossus on Atari in 1985).

- In this way we change winning/losing game to a game with zero sum.
- In general, we search the game-tree up to the horizon and yet we may employ α-β-pruning. If such an algorithm is not good, we may try further heuristic:
- Try "perspective" branches first (good advice but who tells us what is "perspective" – static evaluating function).
- Note that this "improvement" is just a heuristic that may easily get us in trouble!

Comment about real games

- Before the final part of the game we are building a game-tree, game-graph gets constructed in the final part (because of size).
- Heuristic algorithms may be considered in two ways:
- Either they are trying to find the optimum in the fastest way or they are trying to find suboptimal solution that is not too bad.
- Now, how to use heuristic to find a suboptimal solution?

α-β-heuristic

- There are two possibilities:
- The window-method: We use α and β as the bounds for states that make sense to be analyzed (if the state of the game changes rapidly, something is suspicious).
- Cascade-version: We perform BFS through the tree to find the interesting moves (and to avoid searching non-interesting moves).

- We may be changing values α and β with the depth in the tree, also we may restart the tree-search based on our knowledge.
- Usually we are searching up to silent position, i.e., when not much (when some rapid change takes place).
- If some rapid change appears, usually the state of the game continues to change rapidly. Thus we should *not* finish analysing the game in such a state!

Matrix-games

- Given a matrix, one player picks a row, the other picks a column (independently).
- It makes sense to discuss also so called mixed strategy (saying how to play with what probability).

Theorem

For each combinatorial game with zero sum and finite strategies there exists a value V and a mixed strategy for each player such that

- *If second player plays according to the strategy, the first player cannot force the game to finish with more than V,*
- *if the first player plays according to his strategy, the second player cannot force the game to finish with less than V.*

Nash equilibrium

Definition

Nash equilibrium is a set of mixed strategies (one for each player) in finite games of at least 2 non-cooperating players where no player can increase the gain by changing the strategy.

Theorem (J. Nash)

For each game of n players where each player has only finitely many possible strategies there exist Nash equilibrium.

Exercises on games

- Nim with 1 or 2 matches,
- Nim with 1 .. k matches,
- Nim with two piles (1 or 2 matches, 1 .. k or unbounded number of matches),
- General Nim,
- Poisoned chocolat $2 \times n, 3 \times n, n \times n$.
- Turtles,
- Pawn-blocking,
- Why does the first player in noughts and crosses win?
- Round table.

What remains?

- Median in linear time,
- hashing,
- bucketsort (sorting without comparison),
- randomized algorithms (Monte Carlo, Las Vegas),
- graph algorithms (factor set, topological ordering,...).

That's all for today...

...thank you for your attention.