Announcement

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

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.