Algorithmic Aspects of Game Theory

Tomasz Garbus

2019

Contents

1 Lecture 1 (27 II 2019)

2 Tutorials 1 (28 II 2019)

Hex, choquet: 2 players.
We say a game is **determined** if either of players has a winning strategy, If σ is a winning strategy of P, then $\forall_\pi G(\sigma, \pi) \leftarrow$ wins P.

1. Is the choquet game determined if we replace $\mathcal{R}$ with $\mathcal{Q}$ (and its topology)?
If so, who has a winning strategy?

2. Let's consider a variant of choquet games on topological spaces. We have a property: If X is not a Baire space[1] $\implies$ E has a winning strategy (E means Empty, not Eve!).
Example with rational numbers:
$G^q \leftarrow$ set $\mathcal{Q} \setminus q$ dense, open.
Q is countable.
$F \subset Q$
$\cap_{q \in F} G^q = G^F$
$|F| < \mathfrak{c}$
Our strategy:

- we start with set G^{q_0}
- opponent plays a set, say S_1
- we play a set $S1 \cap G^{q_0} \cap G^{q_1}$

3. If X is complete then NE has w.s.
A complete space is also a Baire space.

4. Consider Nim game.
Setup: n heaps with tokens $h_1, h_2, ..., h_n$.
Move: choose a heap and remove $r > 0$ tokens.
Win: The last move.
We have two players: E and $\forall$, Eve move first. Q: Who has a winning strategy? When is the game determined?

$n = 1 \leftarrow$ Eve always wins
$n = 2 \leftarrow$ ((1, 1) wins Adam, (2, 1) wins Eve, (2, 2) wins Adam)
$(h_1, h_2) \rightarrow$ equalise them if possible
Eve has a winning strategy iff $h_1 \neq h_2$
General case: Eve wins if the xor of stack heaps is non-zero. Proof: The winning configuration has xor 0. From a situation with xor $\neq 0$ is always able to produce a situation with xor $= 0$ and if xor $= 0$, it's impossible to make a move such that xor $= 0$ after the move.

1. $(0, ..., 0, h_j, 0, ..., 0)$ is a winning position for Eve.
2. if $h_1 \otimes h_2 ... \otimes h_n = 0$ then the position is *balanced*. Balanced positions are winning positions.
3. Show strategy (next tutorials)

3 Lecture 2 (6 III 2019)

Determinacy

If we have a **game of finite duration** with 2 players, we can expand the game in a tree, where a leaf signifies the end of the game. A leaf maps to one of three possible situations:

- existential player ($\exists$) wins
- universal player ($\forall$) wins
- draw

If we map those situations to values accordingly: $1, -1, 0$, the existential player aims to maximize (and universal to minimize) the outcome value.

Let's consider **infinite** games now. Suppose we have 2 players and draw is not possible in the game. If the player does not

[1] X is Baire if:
$G_i \leftarrow$ are dense and open for $i \in \mathcal{N}$ then $\cap_{i>0} G_i \neq \emptyset$

know the winning strategy, it is possible that they may "lo-op" in a position with winning strategy but never proceed with it.

There **exist** indeterminate perfect information games.

Infinite XOR game: E and A alternately play words $w_0, w_1, w_2, ... \in \{0,1\}^+$ which are concatenated to $w_0 w_1 w_2 ....$

Infinite XOR: any function $f : \{0,1\}^{|\mathbb{N}|} \to \{0,1\}$ such that if v, w differ by one bit then $f(v) \neq f(w)$.
$v \sim w$ iff differ by a finite number of bits.
We can choose set S s.t. $\{0,1\}^{|\mathbb{N}|} \supseteq S$ has $\exists!$ element for each equivalence class (from *Axiom of Choice*).
Each equivalence class of $\sim$ is countable, thus there is continuum of equivalence classes.
Eve wins iff $f(w_0 w_1 ...) = 0$, Adam otherwise. No player has a winning strategy in this game.
1. Suppose Adam wins. In the first play:

```
E  0      w_2       w_4
A    w_1       w_3
```

Then in the next game Eve can steal his strategy:

```
E  1 w_1        w_3       w_5
A         w_2        w_4
```

2. Suppose Eve wins. In the first play:

```
E  w_0    w_2      w_4
A       0      w_3
```

Then in the next play:

```
E  w_0         w_3
A       1 w_2       w_4
```

Game on graph

An *arena* is a directed graph, consisting of:

- the set of *positions* Pos
- the set of *moves* $Moves \subseteq Pos \times Pos$

$Pos = Pos_\exists \cup Pos_\forall$,
$Pos_\exists \cup Pos_\forall \neq \emptyset$.
A *play* is a finite or infinite sequence of moves:
$q_0 \to q_1 \to q_2 \to ... \to q_k (\to ...)$.

Game equation

$$X = (E \cap \diamond X) \cup (A \cap \Box X) = Eve(X)$$
$$Y = (E \cap \Box Y) \cup (A \cap \diamond Y) = Adam(Y)$$

$E = Pos_\exists$,
$A = Pos_\forall$,
$X, Y \in \mathcal{P}(Pos)$
"Modal logic" symbols here:
$\diamond Z = \{p : (\exists_q) Moves(p, q) \wedge q \in Z\}$ [2]
$\Box Z = \{p : (\forall_q)(p \to q) \Rightarrow q \in Z\}$

[2] $p \to q$ also denotes $Moves(p, q)$ below. A position p, such that $(\forall_p) p \not\to q$ is called *terminal*, which we also write $p \not\to$.

Knaster-Tarski Theorem: $\langle L, \leqslant \rangle$ complete lattice[3], $f : L \to L$ monotonic, then there exists a least fixed point
$\mu x.f(x) = \bigwedge \{d : f(d) \leqslant d\}$ and a greatest fixed point:
$\nu y.f(y) = \bigvee \{d : d \leqslant f(d)\}$.
Proof: We show the proof for the greatest fixed point. Let $a = \bigvee A, A = \{z : z \leqslant f(z)\}$
Because f is monotonic, $z \leqslant a$ implies $f(z) \leqslant f(a)$. For $z \in A$, this also means $z \leqslant f(z) \leqslant f(a)$. Hence, $f(a)$ is an upper bound of A, which follows $a \leqslant f(a)$. Using the monotonicity of f once more, we obtain $f(a) \leqslant f(f(a))$. Hence $f(a) \in A$, which follows the converse inequality $f(a) \leqslant a$.

We consider mappings *Eve* and *Adam* defined in the complete lattice $\langle \mathcal{P}, \leqslant \rangle$. $Eve(Z)$ is a set of such positions from which Eve can win, and $Adam(Z)$ is a set of such positions from which Adam can win.

Traps and gardens of Eden

A set of positions $Z \subseteq Pos$ is a trap for Adam if $Z \subseteq Eve(Z)$. It means that Adam cannot go out of there.
A set of positions $Z \subseteq Pos$ is Garden of Eden for Eve if $Eve(Z) \subseteq Z$. It means that Adam cannot enter those positions.

The *greatest* trap for Adam is a garden of Eden for Even.
The *least* garden of Eden for Eve is a trap for Adam.

We use the notation: $\overline{Z} = Pos - Z$.

Lemma

$$\overline{Eve(X)} = Adam(\overline{X})$$

Proof. We have:
$\overline{Eve(X)} = \overline{(E \cap \diamond X) \cup (A \cap \Box X)}$
$= \overline{(E \cap \diamond X)} \cap \overline{(A \cap \Box X)}$
$= (\overline{E} \cup \overline{\diamond X}) \cap (\overline{A} \cup \overline{\Box X})$
$= (A \cup \Box \overline{X}) \cap (E \cup \diamond \overline{X})$
$= (A \cap \diamond \overline{X}) \cup (E \cap \diamond \overline{X}) \cup (A \cap E) \cup (\diamond \overline{X} \cap \Box \overline{X}) = Adam(\overline{X})$

Proposition: Pos can be divided to three disjoint sets: $\mu X.Eve(X), \mu X.Adam(X), (\nu Y.Eve(Y)) \cap (\nu Y.Adam(Y))$

Definition: strategy

A strategy (for Eve) is a set of finite plays s.t.:

- if $last(w) \in Pos_\exists$ then $\exists! q$ s.t. $last(w) \to q$ and wq is in S
- if $last(w) \in Pos_\forall$ then $\forall (q)(last(w) \to q) \Rightarrow wq \in S$
- S is closed under initial segments, i.e., if $s_0 s_1 ... s_k \in S$, then $s_0 s_1 ... s_i \in S$, for $0 \leqslant i \leqslant k$j

[3] A *complete lattice* is a partially ordered set $\langle L, \leqslant \rangle$, such that each subset $Z \subseteq L$ has the least upper bound $\bigvee Z$, and the greates lower bound $\bigwedge Z$. In particular, $\bigvee \emptyset$ is the least element denoted $\perp$, and $\bigwedge \emptyset$ is the greatest element denoted $\top$.

4 Tutorials 2 (7 III 2019)

1. Consider NIM game.
We have n stacks of heights $h_1, h_2..., h_n, h \in \{0, 1, 2, ...\}$.
$W_E = \{x \in \mathbb{N}^R : \text{Eve has a w.s.}\}$.
Let's take $x \in \mathbb{N}^R, x = (h_1, ..., h_n)$.
$h_1^b \otimes h_2^b \otimes ... \otimes h_n^b \neq 0 \rightarrow$ Eve wins from x.[4]
The proof consists of 3 observations:

- Final position (all empty stacks) has xor equal to 0.
- From a position s.t. xor $\neq 0$, it is always possible to zero the xor. Let xor be equal some y. Let d be the position of leftmost (most important) bit in binary representation of xor. Thus in some stack, the binary representation must have The d-th bit activated. We can then deactivate $d-$th bit and set appropriate values on all less significant bits (to zero the xor), and the resulting stack height will be lower.
- From a position s.t. xor $= 0$, all moves lead to xor $\neq 0$. If we take a non-zero number of tokens from a stack, it means we alter a non-zero number of binary digits in the representation of xor, thus it is no longer 0.

Thus, $W_E = \{x \in \mathbb{N}^R : \otimes x \neq \vec{0}\}$.

2. Represent NIM as a game on graph.
The set of positions is the set of all stack configurations. In general, the space of positions is infinite, but for a fixed game we have finite set of positions.
However, this is not enough (in a game on graph, we want Pos to be disjoint set $Pos = Pos_E \cup Pos_A, Pos_E \cap Pos_A = \emptyset$). Thus we also add information who moves next.
Graph: $G = \langle V_E \dot{\cup} V_A, E\rangle$.
Strategy is a function $\sigma : V^*V \rightarrow V$, $\sigma(v_0v_1v_2...v_nv_{n+1}) \rightarrow v$.
Let $w = v_1...v_n...$ be a winning position. We say that first player wins if their strategy $L \in V^w$.
In NIM, the winning condition does not depend on history, so we can collapse the states with the same stacks configuration in the last move (and of course the same currrent player).

3. Chocolate game.
We have a grid $m \times n$. A player chooses a field and everything on the right and up is erased. One restriction: you cannot choose position $(1, 1)$. The player who makes last move wins. [5] Is the game determined? Who has a winning strategy?

$1 \times n$: Eve wins (obvious)

$2 \times n$: Eve wins: she eats the position $(n, 2)$ and always maintains the bottom row has one piece more than the top one.

$\omega \times \omega$: [6] Eve wins: eat the position $(2, 2)$ and then maintain $(1, n); (n, 1)$

$m \times n$: [7] Assume the 2nd player wins. That means that for any move made in the first move, there exists such move from a second player, that the second player has a winning strategy after the second move.
We can use a strategy stealing argument: First player removes a rectangle larger than 1×1. Then by assumption the second player has a winning strategy. But instead we can start by removing only one piece of chocolate. Then second player must make a move such that only a single rectangular area is removed (which could be made in one move by the first player). Then first player can copy second player's winning strategy. Thus second player does not have a winning strategy.

Above we have shown that the second player has no w.s.
$\neg\exists_\pi\forall_\sigma\pi$ wins with $\sigma \;\Leftrightarrow\; \forall_\pi\exists_\sigma\sigma$ wins with π. But this doesn't imply $\exists_\sigma\forall_\pi\sigma > \pi$ (although there is an implication the other way).
We will just show the game is determined, without showing the winning strategy:
We can build the graph of positions. All paths from $m \times n$ (starting position) are finite, the size of graph is finite as well. We can thus infer the winning position by searching the graph bottom-up (from node $0, 0$ to m, n). We have thus shown the game is determined for a finite size. Since Adam does not have a winning strategy, thus Eve must have it.
For $\omega \times \omega$ we use an argument that after the first move the game graph must have a finite height.

At home: think about chess determinacy, also Armageddon version (black wins if he doesn't lose, also for draw).

5 Lecture 3 (13 III 2019)

Arena: $Pos = Pos_E \dot{\cup} Pos_A$
Move $\in Pos \times Pos$

Zermelo: in chess, either White or Black have winning strategies, or both have a strategy for a draw (at least).
Theorem: In a graph game, for each position γ either Eve has a strategy to win in finite time, or Adam has a strategy to win in finite time, or both have strategies to survive. Moreover, all strategies can be positional [8].
Strategy (for Eve): *I will tell you what to do if you have obeyed me so far.*
for $w \in S$:
if $last(w) \in Pos_E$ then $(\exists! p) wp \in S$
if $last(w) \in Pos_A$ then $(\forall p)((last(w) \rightarrow p) \Rightarrow wp \in S)$

Strategy S is positional (memory-less) if $last(w) = last(v)$ then $f_s(w) = f_s(v)$.

f_S can be viewed as a partial function on positions of Eve.

If $X \subseteq Eve(X)$ then any function $dom_f = X \cap Pos_E$ such that $(\forall x \in dom_f) f(x) \in X$ witnesses the trap X.
A function is safe if it is a witness of a trap.
Prop. A partial function $f : Pos_E \supseteq dom_f \rightarrow Pos$ is f_S for some positional strategy iff it is safe. Moreover if f is a witness of trap

[4] (h^b means binary representation)
[5] Other definition: players eat chocolate, position $(1, 1)$ is poisoned.
[6] $\omega = \{0, 1, ...\}$
[7] $m, n \in \mathbb{N}$
[8] A strategy S is *positional* (or *memory-less*) if this function depends only on the current position.

Z then $f = f_S$ such that $start(S) = Z$[9].

<u>Proof</u>: Suppose f is a witness for trap Z. We define $S = \bigcup_{n=0} S_n$, $S_0 = Z$, S_n

A position p is safe for Eve if $p \in S$ for some strategy for Eve.
Lemma 1 $Safe_E$ – the set of all safe positions for E. $Safe_E = \nu X.Eve(X)$. Moreover, Eve has a positional strategy S, with $start(S) = Safe_E$.
Lemma 2 Win_E^{fin} – the set of positions, from where Eve can win in finite time. Then $Win_E^{fin} = \mu x.Eve(X)$. Moreover Eve has a positional strategy S finitely winning with $Start(S) = Win_E^{fin}$.
Proposition (from Lecture 2) *The complement of a trap for Adam is a garden of Eden for him; similarly for Eve.*
<u>Proposition</u> (from Lecture 2)
$\overline{\mu X.Eve(X)} = \nu Y.Adam(Y)$
$\overline{\nu X.Eve(X)} = \mu Y.Adam(Y)$
Exercise (from Lecture 2) *Show that the union of any family of traps for a player is again a trap for this player. Note that this implies that the intersection of any family of gardens of Eden for a player is again a garden of Eden for this player. Which more general property of ordered sets underlines these facts? (Remember the Knaster-Tarski Theorem.)*
From definition, a set of positions $Z \subseteq Pos$ is a trap for Adam if $Z \subseteq Eve(Z)$. Let $ZS = Z_1, Z_2, ...$ be a (not necessarily finite) family of traps. For every Z_i and position $p \in Z_i$, $p \in Eve(ZS)$, thus $ZS \subseteq Eve(ZS)$ so we arrive to the definition of trap again.

6 Tutorials 3 (14 III 2019)

Ordinal numbers:
$0, 1, ..., \omega, \omega + 1, ..., \omega + n, ...2\omega, ..., n\omega, \omega^2, ..., \omega^5, ..., \omega^\omega, ...$
$\alpha \quad \alpha + 1 \mid \alpha \cup \{\alpha\}\ 0 = \{\}$
Set of ordinals is well founded (no infinite descending sequence).
$\beta \quad \bigcup_{\alpha<\beta} \alpha$

1. Nim on ordinals. $G = (h_1, .., h_n), h_i < \omega^\omega$.

1) Is G determined (for which starting positions)?
2) If so, who has a winning strategy?

1. <u>Yes.</u> A graph of states is a DAG and can be divided into disjoint states of winning positions for both players.
2.

a) $h < \omega^\omega$
   $h = a_0 + a_1\omega + a_2\omega^2 + ... + a_n\omega^n \quad (a_i \in \mathbb{N})$ (it is easy to show that this fits under ω^ω)
   In finite case ($a_i = 0$ if $i > 0$) we aim to zero the xor of all stacks. We will try to apply this strategy to ordinals. Let's define xor on ordinals.
   α, β – ordinals
   $\alpha \oplus \beta = (\alpha_0 \oplus \beta_0) + (\alpha_1 \oplus \beta_1)\omega + ... + (\alpha_n \oplus \beta_n)\omega^n$
   <u>Statement</u> In game $G = (h_1, .., h_n)$ Eve has a winning strategy iff $\bigoplus_{i>0} h_i \neq \emptyset$. We need to prove that:

   1° If Eve plays from $\oplus \neq 0$ then she can always maintain $\oplus = 0$.
   2° $\oplus = 0 \Rightarrow$ Every Adam's move makes it $\neq 0$.
   3° Ends after a finite number of steps in necessarily Adam's position.

2. Infinite XOR is an undetermined game such that its graph of positions has inifinite branching. Try to find a game such that is undetermined but has finite branching.
We modify the game so that player can only place a single letter from the alphabet $\Gamma = \{0, 1, 2, 3, \#\}$ and we have an interpretation function $f : \{0, 1, 2, 3, \#\}^\omega \to \{0, 1\}^\omega$. If a player does not place a hash (end of a word) in a finite time, they lose. Once a player places a hash, its the other player's turn.

Gale-Stewart games: $\langle \Gamma, W \subseteq \Gamma^\omega \rangle$, game of perfect information.

Zermelo: Let $G = \langle V, \to \rangle$ be a graph game. Then there exists a partition $W_E \dot\cup W_A \dot\cup W_N = V$ s.t. player $P \in \{E, A\}$ has a <u>positional</u> winning strategy in position $v \in W_P$.

Reachability game: We select a node in graph, Eve wants to reach that node, Adam does not want to ever reach that node. If the play is infinite and looped without reaching the selected node, Adam wins.

7 Lecture 4 (20 III 2019)

<u>Generalized Zermelo Theorem</u>: For any position p in a deterministic, two-person game with perfect information (and players make moves alternatingly):

- one of the players has a winning strategy (winning in finite time)
- or both players have a surviving strategy

Moreover, those strategies can be positional.
$Eve(X) = (E \cap \diamond X) \cup (A \cap \Box X)$ – those positions, from which X can be achieved in one move. If it is Eve's move, there must be one move to position in X, if Adam's – all moves must lead him to X.
We want to find the smallest fixed point $\mu X.Eve(X)$.
$\bigvee f^\zeta(\perp) = \mu x.f(x) = \bigwedge\{d : f(d) \leqslant d\}$
$\perp$ – minimal element
$f^0(\perp) = \perp$
$f^{\zeta+1}(\perp) = f(f^\zeta(\perp))$
η – limit element:
$f^\eta(\perp) = \bigvee_{\zeta<\eta} f^\zeta(\perp)$

Algorithm for finding winning positions for player $X \in \{Eve, Adam\}$:

```
Program Win(X)     X ∈ {Eve, Adam}
W : Pos  → {⊥,∃,∀}
Forall p ∈ Pos
 W(p) = ⊥
 pred(p) = ∅
 nb(p) = 0
Forall (p, q) ∈ Move
 pred(q) := pred(q) ∪ {p}
 nb(p) := nb(p) + 1
```

[9] $Start(S) = S \bigcap Pos$

```
Forall p ∈ Pos_X̄
 if nb(p) = 0 then Propagate(p, X)

Propagate(q, X)
 if W(q) = ⊥ then W(q) = X
 Forall p ∈ pred(q)
  if W(p) = ⊥ then
   if p ∈ Pos_X then
    W(p) = X
    Propagate(p, X)
   else (* p ∈ Pos_X̄ *) then
    nb(p) := nb(p) - 1
    if nb(p) = 0 then
     Propagate(p, X)
```

Parity games

$Pos_E, Pos_A, Move, rank : Pos \to C$
$Win_E \subseteq C^\omega$
$Win_A \subseteq C^\omega$
$Win_E \cap Win_A = \emptyset$
Parity games: $C \subseteq \mathbb{N}$
$Win_E = \{u \in C^\omega \ : \ \limsup_{n\to\infty} u_n \text{ is even}\}$
$Win_A = \overline{Win_E}$
Every position is winning for one of the players, with a positional strategy.

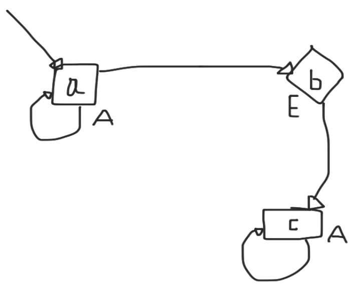
8 Tutorials 4 (21 III 2019)

A game $G = \langle B, W\rangle$
$B = \langle V, E, \lambda, v_I\rangle$
$E \subseteq V$, $\lambda \ : \to \Gamma$, $v_I \in V$
$W \subseteq \Gamma^* \cup \Gamma^\omega$
$\Phi \ : \ \Gamma^\omega \to [0, 1]$

$$\Phi_W(x) = \begin{cases} 1, & x \in W \\ 0, & x \notin W \end{cases}$$

Example:

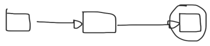
Fig. 1: Consider this simple game on graph

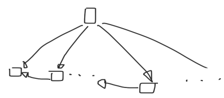


$W = \{w \in \{a, b, c\}^\omega \mid \exists_{n<\omega} w = a^n b^n c^\omega\}$ – Adam has a winning strategy (he can infinitely loop in `a`).

Strategies for Eve and Adam:
$E :$ $\delta \ : \ V^*V_E \to V$ s.t. $\delta(w, v) \to (v')$ (and $v, v' \in E$)
$A :$ $\pi \ : \ V^*V_A \to V$
Or positional strategies: $\delta \ : \ V_E \to V$
$\pi \ : \ V_A \to V$
A play:
$G < \delta, \pi > \to v_0v_1v_2...v_n.... = p$
$w = \lambda(v_0)\lambda(v_1)...\lambda(v_n)...$
$w \in W \Leftrightarrow \delta$ wins with π

(still considering the game above)
$W_2 = W \cup \{a^\omega\}$ – Eve has a winning strategy (but not a positional one)

Reachability game (same graph as before):
$W = \Gamma^* c \Gamma^\omega \cup \Gamma^* c$
$W = V^* F V^\omega \cup V^* F V^*$
$F \subseteq V$ (we want to reach F)
1. Reachability condition, after how many steps every game ends. In this case, the bound is infinity, since Adam can stay in node `a`.
2. Create a game that forces the end after 2 steps  (we can extend this for any natural number)
3. Enforce that the game ends after ω steps (you cannot bound it from below). [10]

We create such infinite graph: 
4. Same for $\omega + 1$ – add one more v_I to the previous graph
5. $\omega + \omega$ – copy the graph, the output from first copy goes into the initial vertex of the second one

$f : V \to V$
$f(S) = S'$, $S \subseteq S'$
$S' = \{v \in V \mid \forall_{v'} E(v, v') \to v' \in S\} \cup \{v \in V \mid \exists_{v'} A(v, v') \to (v, v') \in E\} \cup S$
$f^\alpha(F) = S_\alpha$
$S_0 = F$
$S_\alpha \leftarrow$ set of winning positions of Eve
strenght of game = smallest α s.t. $v_I \in f^\alpha(F)$

6. Given a reachability game $G = \langle V, W\rangle$, compute the set W_E ($W_E = \{v \mid$ Eve has a winning strategy from $v\}$).

```
S = F;
S' = f(S);
while(S != S') {
 S = S';
 S' = f(S);
}
return S;
```

Parity games

Parity condition:
$\{w \in \Gamma^\omega \mid w = a_0a_1a_2...a_n..., \limsup a_n \text{ is even } \}$ where $\Gamma \subseteq \mathbb{N}$
Theorem: Parity games are positionally determined.

7. Let G be a parity game, i.e. $G = \langle B, \textsc{Parity}\rangle$, s.t. $V_A = \emptyset$ (only Eve moves). Compute the set of winning positions for Eve.[11]

[10] The smallest number of steps required for reachability game to finish is sometimes called the **strength of a reachability game**.
[11] Since we are looking for an algorithm, assume graph G is finite in size.

Observation $v \in V$ is a winning position iff we can reach from v a loop $x_0...x_n$ s.t. the highest priority in the loop is even. Proof

- $\Leftarrow$ easy

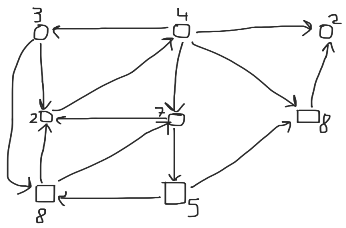
- $\Rightarrow$ $\exists \rightarrow (v = v_1)v_2v_3...v_n...$ with ranks $a_1a_2a_3...a_n... \rightarrow$ winning. The highest priority occurring infinitely often is even, let's call it a.
  Thus $\underset{n_0}{\exists} : \underset{n \geqslant n_0}{\forall}\ a_n \leqslant a$
  Take a sequence $a_{n \geqslant n_0}...a...a....a....a$. So we choose a path between two occurrences of a and loop it.

HOME Describe the algorithm basing on the observation.

9 Lecture 5 (27 III 2019)

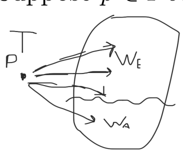
Parity games: Eve wins a play $\pi = p_0p_1...$ if $\limsup\limits_{n\to\infty}(rank(p_n))$ is even, Adam otherwise.

Fig. 2: What are W_E, W_A here? (squares are Adam, circles Eve)



General games on labelled graphs:
$Pos_E, Pos_A, Move \in Pos \times Pos$
$rank\ :\ Pos \longrightarrow C$
Winning criteria: $W_E \subseteq C^\omega, W_A \subseteq C^\omega, W_E \cap W_A = \emptyset$
Theorem: Parity games are positionally determined. We prove for finite arenas. Some positions are marked $\top$ immediately winning for Eve, $\bot$ immediately winning for Adam. Induction on #positions, not marked $\top, \bot$.
Induction step: Choose a position p with the highest possible rank d, assume (wlog) d is even.

1 Suppose $p \in Pos_E$. Mark p by $\top$.



(1a) There is a move from p to W_E.

(1b) All moves from p go to W_A.

2 $p \in Pos_A$

(2a) All moves from p go to W_E.

(2b) There is a move from p to W_A.

Parity game – n vertices, d ranks. "Simple" algorithm – $n^{\frac{d}{2}+\Theta(1)}$

10 Tutorials 5 (28 III 2019)

1. Problem
In: a parity game G and a positional strategy of Eve σ
Out: is σ winning?
σ is winning $\leftrightarrow \forall_\pi G(\sigma, \pi) \in$ Parity
$\sigma\ :\ V \rightarrow V$

2. In: a parity game G, and an initial position v_I.
Out: Is v_I winning for Eve?
Solution in NP:

- guess the positional strategy of Eve $\rightarrow \sigma_p$

- check if σ_p is winning

Solution in coNP – check for strategy for Adam.
The problem is (at least) in NP $\cap$ coNP

3.
Muller $C \subseteq \mathbb{N}, \mathcal{F} \subseteq 2^C$
$C^\omega \ni p = a_1a_2...a_n...$ if $Inf(p) \in \mathcal{F} \longleftarrow$ set of letters in p that appears ∞-often
Parity – $max(Inf(p))$ is even

1) Parity is Muller

2) Muller is not positionally determined

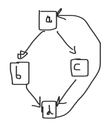
3) Muller is not Parity

4) Show that Muller is determined, what is the required memory? Show that if G is a Muller game, then for every initial vertex v_I, one of the players has a winning strategy.

Solutions:

1) Parity game of index (i, k) can be represented as a Muller game, where $\mathcal{F}$ consists of subsets of $\{i, i+1, ..., k\}$ with even maximum.

2) Example below:



$\mathcal{F} = \{\{a, b, c, d\}\}$
Every positional strategy removes either b or c and we require both.

3) From (2)

4) Idea: We will try to transform it to some Parity game, i.e.
$G = \langle V, E, \lambda, v_I, \mathcal{F}\rangle \rightsquigarrow G' = \langle V', E', \lambda', v_I', \text{Parity}\rangle$
$E \subseteq V \times V$
$\lambda\ :\ V \rightarrow C$
s.t. v_I is winning for Eve in $G \Leftrightarrow v_I'$ is winning for Eve in G' (we need the same for Adam). In fact, we only need the implication to the left side.

LARs

LAR (*latest appearance record*) $w \in C \cup \{\natural\}(=\Gamma)$, every letter occurs at most once, $\natural$ occurs always.
up : LAR $\times \Gamma \to \Gamma$

$$\mathbf{up}(v_i\natural w_i, a) = \begin{cases} v_i\natural w_i a & a \notin v_i, a \notin w_i \\ [v_i w_i]_{a\mapsto\natural,\natural\mapsto\emptyset} a & \text{else} \end{cases}$$

<u>Theorem</u>: The following cylindrification of G is a solution to task (4): $rank(p, v\natural a_1 ... a_l) = \begin{cases} 2l & \text{if}\{a_1, ..., a_l\} \in \mathcal{F} \\ 2l+1 & \text{otherwise} \end{cases}$

11 Lecture 6 (3 IV 2019)

G^{++}
$\{0, 1, ..., n\}^{k+1} \approx k+1$-digit numbers in base $n+1$
Overflow $= (n+1)^{k+1}$

$\mathbf{up}(2i+1, \underbrace{a_0a_1a_2...a_k}_{m}) = m + (n+1)^i$
$\mathbf{up}(2i, a_0a_1...a_k) = 0...0a_ia_{i+1}...a_k$

G^{++} is equivalent to G (i.e. if Adam has winning strategy in G, he has a winning strategy in $G++$)

12 Tutorials 6 (4 IV 2019)

Last week

<u>Theorem</u> If G is a finite Muller game, then one of the players has a finite memory winning strategy. $\langle V, E, v_I, \textsc{Muller}\rangle$
<u>Proof</u> G' – a parity game s.t.
Adam wins in $G' \Rightarrow$ Adam wins in G
Eve wins in $G' \Rightarrow$ Eve wins in G
LARs

Automata

$\mathcal{A} = \langle \Gamma, Q, q_i, \delta, F\rangle$
Word: $q_I \in Q, \delta : Q \times \Gamma \times Q, (q, a) \rightsquigarrow \{q_1, q_2, q_3, ...\}$

Infinite automata

Büchi condition $F \subseteq Q$ (the states appearing infinitely often)
Müller condition $F \subseteq 2^Q$
Parity
Safety – Parity with condition [0]
<u>Theorem</u> Every regular language of infinite words is recognizable by a deterministic Müller automaton.

1. Show that if G is a finite Müller game, then set of plays is regular.

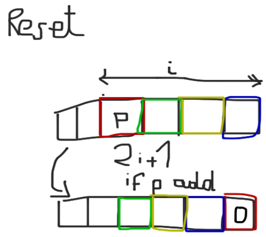
P – set of plays
$P \subseteq V^\omega$
The automaton is the graph. $\langle V, V, v_I, \{(v_1, v_2, v_3) \mid E\langle v_1, v_2\rangle\}\rangle$

<u>Def</u> Game is ω-regular if the winning condition is a regular set of winning plays.
$G = \langle V, L\rangle$. ($L \subseteq \Gamma^\omega$, Eve wins if $p \in L$)

2. Show that if G is ω-regular finite game then one of the players has a finite winning strategy.

$G = \langle E, V_I, V_A, V_E\rangle, V = V_A \cup V_E$
$G_A = \langle V'_A = V_A \times Q, V'_E = V_E \times Q, V'_I, E'\rangle$
$G_A = \langle V', E', V'_I\rangle$
$V' = V \times Q$
$E' = \{((v_1, q_1), (v_2, q_2)) \;:\; (v_1, v_2) \in E, q_2 = \delta(q_1, v_1)\}$
$v'_I = (v_I, q_I)$
$W = inf(L(P)) \in \mathcal{F}$
$L = \pi_2(p)$
<u>Lemma</u>: E has a finite memory winning strategy in $G_A \Rightarrow E$ has a winning strategy in G.

13 Lecture 7 (10 IV 2019)



Parity games n, $\underbrace{d}_{rank}$.
Karoliina Lehtinen – algorithm.
G – a finite parity game.
We create a game $\mathcal{R}^E_k(G)$ with k registers containing priorities $Mem \subseteq \{0, 1, ..., d\}^k$. $Pos_E = (Pos \times Mem \times \{0\}) \cup (Pos_E \times Mem \times \{1\})$
$Pos_A = Pos_A \times Mem \times \{1\}$.

If $p \to q$ in G then
$(p, \alpha, 1) \xrightarrow[1]{(} q, up(\alpha), 0)$
$(p, \alpha, 0) \to (p, reset(\alpha), 1)$
$(p, \alpha, 0) \xrightarrow[1]{\text{[skip]}} (p, \alpha, 1)$

$[up(\alpha)]_i = max(\alpha_i, rank(q))$

Lemma 1 If Adam wins original game G from position p, he wins $\mathcal{R}^E_k(G)$ from position $(p, \alpha, 1)$ for any number of registers k. Adam plays the strategy from original game, we will show that any play is winning. Let q be the maximal odd rank. Let i be the deepest register, which resets infinitely often. We will show, that infinitely often, in the moment of reset of the register i, it contains q.

Lemma 2 If Eve wins the game G from position p, then $(\exists_k)$ Eve wins $\mathcal{R}^E_k(G)$ from position $(p, \alpha, 0)$.
k: number of ranks from the original game.
To each even rank d we dedicate some register. When in original game there appears an odd rank d, Eve resets the register dedicated to rank d.

14 Tutorials 7 (11 IV 2019)

Büchi-Landweber thm.

Let G be an ω-regular game, i.e. the winning strategy is ω-regular. Then, ether of the players has a winning strategy.

$\langle V, E, v_I, \lambda, L\rangle$
$\lambda(v_0 ... v_n ...) \in L$.
1) L is regular $\Rightarrow$ L recognized by Deterministic Müller Automaton $\mathcal{A}$.
2) G' is a Müller Game created as a product of $(V, E, \lambda) \times \mathcal{A}$.
3) Adam/Eve wins in G' $\Rightarrow$ Adam/Eve wins in G.

Complexities

Problem 1.
<u>In.</u> $G = \langle B, L(A)\rangle \leftarrow \omega$-regular game.
<u>Out.</u> Does Eve win?

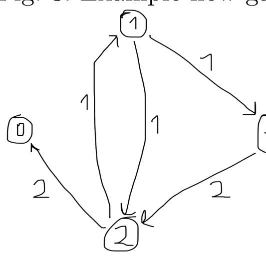
a) A is a deterministic parity automaton.
The upper bound is $NP \cap coNP$, because G is a parity game.
Algorithm: Reduce G to $G' = G \times A$ parity game, solve G'.
Lower bound: can encode parity games.

b) A is a DMA.

c) A is a non-deterministic Büchi automaton.
Algorithm: compute DPA A' such that $L(A) = L(A')$, $|a|^{|a|}$, $[0, 2|a|]$, use (a). This is in EXP.

Flow Games

Fig. 3: Example flow game

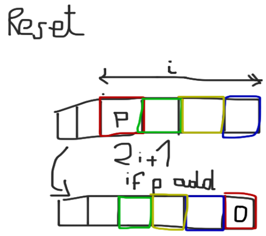


$G = (V, E, E_\diamond \dot\cup E_\Box, \lambda, e \in E)$.
$\lambda : E \cup V \to [0, d]$
Valid is $\forall_v \lambda(v) \leqslant \sum\limits_{(a,v)\in E} \lambda(a, v)$.
A player $P \in \{\diamond, \Box\}$ chooses an edge $e \in E_p$, and flips it.
Player $\diamond$ wins if a predefined edge e is flipped.
Problem 2. Show that if there is only one player $\diamond$, and every edge can be flipped at most once, then the game is NP-complete.
<u>In NP:</u> It is in NP because we can guess the witness in NP and verify in polynomial time.
<u>Complete:</u> Reduction 3-SAT[12] $\to$ Game.

Problem 3. Show that if w allow that an edge is fliped at most once, then the game is APTIME-complete, PSPACE-complete.

[12] 3-SAT: Formula $F_1 \wedge F_2 ... \wedge F_n$, s.t. $F_i = x_{i_1} \vee x_{i_2} \vee x_{i_3}$ s.t. $x_j = y_i$ (or $\neg y_i$) for $y_1 ... y_k$.

15 Lecture 8 (18 IV 2019)



Karoliina Lehtinen algorithm

16 Homework1

5 Multi-reachability games

In a reachability games there is a set of vertices which the first player wants to reach. In multi-reachability games (MRG) there is a family of sets of vertices and the first player wins if every set in the family has been visited at least once.
1. Show that MRG game is PSpace-complete.
1. <u>$\in$ PSpace</u>

2. <u>PSpace-hard</u>
I will show a reduction of QBF problem to an MRG game.
An input to the QBF problem is a formula, about which I will make two assumptions:

- It is in prenex normal form (i.e. all quantifiers preceed the portion containing an unquantified Boolean formula). Moreover, let's assume that the existental and universal quantifiers alternate – if it is not the case in the original input formula, we can introduce quantifiers with dummy variables, not used anywhere in the formula. For instance, $\exists_{x_1}\exists_{x_2}\phi(x_1, x_2) \mapsto \exists_{x_1}\forall_{y_1}\exists_{x_2}\phi(x_1, x_2)$ (y_1 is a "dummy" variable).
- The "body" of the formula is in conjunctive normal form.

Note that the above assumptions do not reduce the expressive power of input formulas. Every possible formula can be represented in the described format. QBF problem for such normalized formulas is still PSpace-complete.
Let $\forall_{x_1}\exists_{x_2}\forall_{x_3}...\exists_{x_n}(y_{1,1} \vee ... \vee y_{1,k_1}) \wedge (y_{2,1} \vee ... \vee y_{2,k_2}) \wedge ...$ be the input QBF formula, where $y_{...} \in \{x_1, ..., x_n\}$. The created multi-reachability game is $G = \langle V, E, v_I, S\rangle$, where:

- V is the set of vertices. Vertices are indexed by all variables bound by quantifiers and their negations, plus there is the initial vertex. $V = \bigcup_{1\leqslant i\leqslant n}\{v_x, v_{\neg x}\} \cup \{v_I\}$
- E is the set of edges. $E = \{(v_I, v_{x_1}), (v_I, v_{\neg x_1}\} \cup \bigcup_{2\leqslant i\leqslant n}\{(v_{x_{i-1}}, v_{x_i}), (v_{x_{i-1}}, v_{\neg x_i}), (v_{\neg x_{i-1}}, v_{x_i}), (v_{\neg x_{i-1}}, v_{\neg x_i})\}$
- S is the family of sets of vertices that the first player wants to reach. It is created directly from the CNF formula.