Milestones in solving games on graphs

Sasha Rubin

November 30, 2017

1 Outline

Games on graphs are a useful mathematical model of many phenomena in computer science that involve interaction between players/agents/components. Solving a game means deciding if a given player has a winning strategy. There are a number of classes of games whose complexity are in $NP \cap coNP$, but are not known to be in P, i.e., parity games, mean-payoff games, discounted payoff games, simple stochastic games. This year, there was a breakthrough in the analysis of parity games, a deceptively simply class of games with tight connections to deep results in logic and automata theory. A team from New Zealand and Singapore [Calude et. al. 2017] gave a quasi-polynomial time algorithm for solving parity games — previous algorithms were mildly subexponential. We will chart a course through the theory of games on graphs, studying determinacy, memory required for winning, and the complexity of solving games. We will start with foundational work and end with very recent results.

Participants will give a talk on a classic or current paper. There are a variety of topics to choose from: the tight connection between graph-games and logic and automata-theory, the use of graph-games in formal-methods (modeling, verification, synthesis, testing, composition, simulation), the use of graph-games in AI (automated planning, verification in robotics, general-game playing), or extensions of the basic model (multiplayer games, partial-information games, stochastic games, pushdown games, timed games).

1.1 Core Papers

- Ehrenfeucht and Mycielski (1979). Positional strategies for mean payoff games.
- McNaughton (1993). Infinite games played on finite graphs.
- Zwick and Paterson (1995). The complexity of mean payoff games.
- Dziembowski, Jurdzinski, and Walukiewicz (1997). How much memory is needed to win infinite games?
- Zielonka (1998). Infinite games on finitely coloured graphs with applications to automata on infinite trees.
- Voge and Jurdzinski (2000). A discrete strategy improvement algorithm for solving parity games.

- Chatterjee, Henzinger, and Jurdzinski (2005). Mean-payoff parity games.
- Schewe (2007). Solving Parity Games in Big Steps.
- Jurdzinski, Paterson, and Zwick (2008). A deterministic subexponential algorithm for solving parity games.
- Friedmann (2009). An exponential lower bound for the parity game strategy improvement algorithm as we know it.
- Benerecetti, Dell'Erba and Mogavero (2016). Solving parity games via priority promotion.
- Aminof and Rubin (2016). First Cycle Games.
- Calude, Jain, Khoussainov, Li, and Stephan (2017). Deciding parity games in quasipolynomial time.
- Jurdzinski and Lazic (2017). Succinct progress measures for solving parity games.

1.2 Additional reading

- Gradel, Thomas, and Wilke (Eds.) (2002). Automata, Logics, and Infinite Games: A Guide to Current Research.
- Apt and Gradel (Eds.) (2011). Lectures in Game Theory for Computer Scientists.

2 Where do games come from?

1. Where do games arise? Why analyse games?
2. Simple llustration of connection with logic.

Outline Games on graphs are a useful mathematical model of many phenomena in mathematics and computer science that involve interaction of two or more players. We will define games on graphs and mention a few scenarios where they arise, including automated planning in Artificial Intelligence, automata theory and mathematical logic. We will illustrate by showing how one can think of evaluating Boolean formulas as a game on the parse tree of the formula.

Main idea: games capture strategic interactions.

Model: two players moving a token along the edges of a graph, resulting in an infinite path called a play. Player 0 is trying to ensure the play is in a given set, and Player 1 is trying to ensure it is outside the given set.

Definition 1 (Graphs, Arenas, Games)**.**

1. *A* graph *is a structure* (V, E) *where* V *is a set of vertices and* $E \subseteq V \times V$ *a set of edges.*

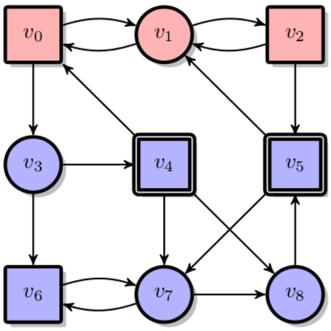
2. *An* arena *is a structure $A = (V, E, V_0, V_1)$ in which E is total, i.e., for every $v \in V$ there exists $w \in V$ such that $E(v, w)$ (we also say "no dead-ends").*

3. *A* play *is an infinite path in A.*

4. *A* game *is a structure $G = (A, Obj)$ where $Obj \subseteq V^\omega$ is an objective.*

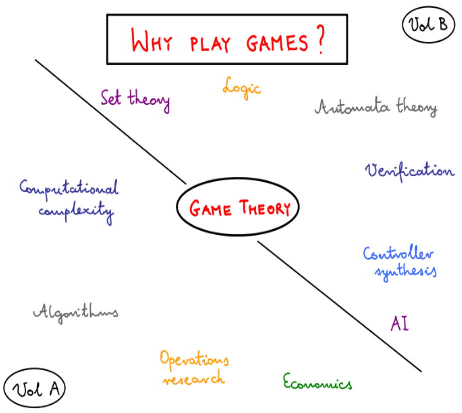
In reachability games, Player 0 is trying to force a visit to T, and Player 1 is trying to ensure the play never visits T.

Definition 2 (Reachability)**.** *Let $T \subseteq V$ and define objective* Reach(T) *to consist of all π such that some $\pi_i \in T$. In this case we say that π* reaches *T; otherwise π* avoids *T.*



Example 1. *Calculate from which vertices player Player 1 (round nodes) can force a visit to T (the double-framed vertices)?* ***Draw*** *positional strategy.*

2.1 Why play games?



Motivations/applications

- Games in logic: evaluating logic formulas
- Games humans play: the game-tree is a graph-game.
- Games from algorithms: graph reachability to sabotage games
- Games in automata: gives semantics to alternating automata
- Games in verification/AI: games model reactive systems, strategies model controllers.
- Economics/Game theory: mathematical theory of how agents, e.g., people, behave in strategic scenarios; games in extensive form; graphical games.
- Games in set theory: determinacy in games used to prove that one can reverse quantifiers in mathematical statements.
- Computational complexity: alternating TM/time/space.

Games humans play

Question 1. *How can you think of games humans play as games on graphs?*

1. Vertices correspond to "snapshots" called *configurations*.
2. Edges correspond to one-step moves in the game. Get a tree or a DAG.
3. Objectives correspond to the winning plays. E.g., *Obj* is the set of plays such that the first player to make a row of three is player 0.

To make this a reachability game, you should loop the first time someone wins and then let T be the set of winning configurations.

If a game has finitely many rounds, then T is finite. In this case, we can solve the game in LOGSPACE in the size of T. How?

Is this useful?

- tic-tac-toe (*tris*) has 9! vertices in the game-tree (approx 10^5).
- chess (*scacchi*) has at least 10^{120} vertices in the game-tree (Shannon, 1950), and about 10^{40} positions in game-graph.

Controller synthesis An open/reactive system is one with controllable and uncontrollable events. This can be thought of as a game between a controller/system player and an environment player. Controllers are strategies that say how to control the variables.

Question 2. *What are the controllable events, uncontrollable events, and objectives in the following open systems: ATM machines, Anti-lock braking system in car.*

1. ATM machine: controllable = internal state of the system, uncontrollable = what actions people will do using the ATM, objective = always give correct amount of money, ensure security, etc.
2. Anti-lock braking system in car: controllable = how much to apply breaks to each wheel, uncontrollable = sensor readings like the speed of each wheel, objective = maintain uniform wheel speed.

Positional strategies

Definition 3. *Fix a game $G = (V, V_0, V_1, E, Obj)$.*

- *A* positional strategy *for player i is a function $\sigma : V_i \to V$ such that $(v, \sigma(v)) \in E$.*
- *A play $\rho = \rho_0\rho_1 \dots$ is consistent with σ if whenever $\rho_j \in V_i$, $\rho_{j+1} = \sigma(\rho_j)$.*
- *A strategy σ_0 for player 0 is* winning *if for all plays ρ consistent with σ_0, we have that $\rho \in Obj$. A strategy σ_1 for player 1 is winning if for all plays ρ consistent with σ_1, we have that $\rho \notin Obj$.*

Example. Games for evaluating formulas

Example 2. *We consider propositional formulas without negation. e.g.,*

$$((a \wedge b) \vee c) \vee (a \wedge c)$$

and valuation $v : a \mapsto 1, b \mapsto 1, c \mapsto 0$.

Question 3.

1. *How does one evaluate the formula? Label parse tree bottom up.*
2. *How does one evaluate it top down? **Draw** winning strategy for Player 1.*

Definition 4. *Let Φ be a positive Boolean formula, and v a valuation of its variables. The* evaluation game *of ϕ, v is the reachability game*

$$G_{\phi,v} = (V, V_0, V_1, E, \textsc{Reach}(T))$$

where

- *V is the set of subformulas of ϕ,*
- *$V_0 \subseteq V$ are those subformula whose top operation is $\vee$,*
- *V_1 are the rest,*
- *T are those variables x of ϕ such that $v(x) = 1$, and*
- *E has transitions from $\varphi_1 \, op \, \varphi_2$ to both φ_1 and to φ_2 for $op \in \{\wedge, \vee\}$, and from atoms to themselves.*

Theorem 1. *Let Φ be a positive Boolean formula, and v a valuation of its variables. Then $v(\phi) = 1$ iff player 0 has a winning strategy in $G_{\phi,v}$.*

Proof. Induction on the formula.

The base case is easy.

Consider the case that $\phi = \phi_1 \vee \phi_2$ (the case $\wedge$ is similar). Then TFAE:

1. $v(\phi) = 1$
2. there exists i such that v satisfies ϕ_i
3. there exists i such such that Player 0 has a winning strategy in G_{v,ϕ_i}

4. Player 0 has a winning strategy in $G_{v,\phi}$.

Question 4. *Prove that 3 implies 4.*

We prove 3 implies 4: Let σ be a winning strategy in G_{v,φ_i}. Define a strategy σ' in $G_{v,\varphi}$ that first goes from the vertex φ to φ_i, and then does σ. To see that σ' is winning, let π be a play in $G_{v,\varphi}$. Then the suffix of π formed by removing the first vertex is a play in G_{v,ϕ_i}. By assumption this suffix reaches T.

We prove 4 implies 3: Let σ be a winning strategy in $G_{v,\varphi_1 \vee \varphi_2}$ and suppose $\sigma(\varphi_1 \vee \varphi_2) = \varphi_1$ (the case φ_2 is symmetric). Define a strategy σ' on G_{v,φ_1} to be the restriction of σ to the subformulas of φ_1. Let π be a play in G_{v,φ_1} consistent with σ'. We need to show it reaches T. But the play starting with $\varphi_1 \vee \varphi_2$ and then proceeding like π is a play of $G_{v,\varphi_1 \vee \varphi_2}$ consistent with σ, and thus by assumption reaches T. Thus π reaches T. □

Question 5. *What is the game corresponding to evaluating arbitrary proposition formulas, i.e., allowing negation as an operator?*

3 Games with qualitative objectives

Outline Games with qualitative objectives assign a winner to every play, e.g., reachability games. We will show how to solve reachability games. We will discuss how to compute winning strategies, and raise the issue of determinacy, and show how to solve games with more complex but still qualitative objectives.

Definition 5. *A* strategy *for player i is a function from $\sigma_i : V^*V_i \to V$ such that for $u \in V^*V_i$ we have that $(last(u), \sigma(u)) \in E$. Strategies σ_0, σ_1 induce a unique play, denoted $play_v(\sigma_0, \sigma_1)$. A strategy σ_0 is* winning *from v if for all strategies σ_1, the play $play_v(\sigma_0, \sigma_1) \in Obj$. Call a play π* consistent *with strategy σ_0 if for every $u \in V^*V_i$ that is a prefix of π we have that $u \cdot \sigma_0(u)$ is a prefix of π.*

Question 6. *Prove that σ_0 is a winning strategy (i.e., beats all counterstrategies) iff every play π consistent with σ_0 is in Obj.*

Definition 6. *The* winning region *$W_i(G)$ of player i is the set of vertices v such that there exists a winning strategy σ_i from v.*

Question 7. *Can a vertex be in both winning regions? Is every vertex in a winning region of one of the players? (come back to this later).*

Definition 7. Solving a game *means computing the winning regions of both players, i.e., given G and v and player i, decide if $v \in W_i(G)$.*

Theorem 2. *Solving reachability games is in* PTIME.

Proof. Fix a reachability game $G = (A, \textsc{Reach}(T))$ and $i = 0, 1$. For $X \subseteq V$, define

$$\begin{aligned}\mathrm{CPre}_i(X) = \{v \in V_i : \exists w \in X.(v,w) \in E\} \bigcup \\ \{v \in V_{1-i} : \forall w.(v,w) \in E \to w \in X\}\end{aligned}$$

and

$$\begin{aligned}\mathrm{Att}_i^0(X) &= X\\ \mathrm{Att}_i^{k+1}(X) &= \mathrm{Att}_i^k(X) \cup \mathrm{CPre}_i(\mathrm{Att}_i^k(X)) \text{ for } k > 0.\\ \mathrm{Att}_i(X) &= \cup_{k\geq 0}\mathrm{Att}_i^k.\end{aligned}$$

Question 8. *Compute $Att_i(T)$ in the example.*

Question 9. *Write the meaning of $CPre_i(X)$, $Att_i^k(X)$ and $Att_i(X)$ in words.*

- *$v \in CPre_i(X)$ iff player i can force the play from v, in one step, to visit X.*
- *$v \in Att_i^k(X)$ iff player i can force the play, in at most k steps, to visit X.*
- *$v \in Att_i(X)$ iff player i can force the play to visit X.*

Question 10. *Prove that $Att_i(T)$ is the least fix-point of the operation $X \mapsto T \cup CPre_i(X)$.*

We now prove that $\mathrm{WR}_i(G) = \mathrm{Att}_i(T)$. Let $\mathrm{rank}(v)$ be the smallest integer k such that $v \in \mathrm{Att}_i^k(T)$, and ∞ if there is no such k.

We prove that $\mathbf{Att}_i(T) \subseteq \mathbf{WR}_i(G)$. Note the following:

1. $v \in \mathrm{Att}_i^k(T)$ implies $\mathrm{rank}(v) \leq k$.
2. $\mathrm{rank}(v) = 0$ iff $v \in T$.
3. If $v \in V_i \cap \mathrm{Att}_i(T)$ and $0 < \mathrm{rank}(v)$ then some $(v, w) \in E$ has $\mathrm{rank}(w) < \mathrm{rank}(v)$.
4. If $v \in V_{1-i} \cap \mathrm{Att}_i(T)$ and $0 < \mathrm{rank}(v)$ then every $(v, w) \in E$ has $\mathrm{rank}(w) < \mathrm{rank}(v)$.

Define a positional strategy σ_i for player i as follows: $\infty > \mathrm{rank}(v) > 0$ then define $\sigma_i(v)$ to be any vertex v' with $\mathrm{rank}(v') < \mathrm{rank}(v)$, and otherwise define it arbitrarily. Then σ_i is winning from every $v \in \mathrm{Att}_i(T)$: every play π consistent with σ_i starting at v, visits T within $\mathrm{rank}(v)$-many steps. Proof is by induction on $\mathrm{rank}(v)$. If $\mathrm{rank}(v) = 0$ we are done. Suppose $\mathrm{rank}(v) > 0$ and take π. The suffix of π after one step is consistent with σ, and $\mathrm{rank}(\pi_1) < k$. By induction, this suffix visits T within $k - 1$ steps, and thus π visits T within k steps.

We prove that $\mathbf{WR}_i(G) \subseteq \mathbf{Att}_i(T)$. To see this it is sufficient to prove that $V \setminus \mathrm{Att}_i(T) \subseteq \mathrm{WR}_{1-i}(G)$ (by "squeezing").

Note the following:

1. If $v \in V_0$ and $\mathrm{rank}(v) = \infty$ then every $(v, w) \in E$ has $\mathrm{rank}(w) = \infty$.
2. If $v \in V_1$ and $\mathrm{rank}(v) = \infty$ then some $(v, w) \in E$ has $\mathrm{rank}(w) = \infty$.

Define a positional strategy σ_{1-i} for Player $1-i$ as follows: for $v \in V_{1-i}$, if $\mathrm{rank}(v) = \infty$ then it maps v to some w with $(v, w) \in E$ and $\mathrm{rank}(w) = \infty$, and otherwise it is defined arbitrarily. Then σ_{1-i} is winning from every v with $\mathrm{rank}(v) = \infty$. Indeed, let π be a play consistent with σ_{1-i} starting with v where $\mathrm{rank}(v) = \infty$. It is easy to see that $\mathrm{rank}(\pi_j) = \infty$ for all j (formally, use induction on j). But $\mathrm{rank}(v) = 0$ for $v \in T$. Thus π avoids T, so Player $1-i$ wins the play. □

We have shown more.

Corollary 1. *For reachability games:*

1. *Player i has a positional strategy that is winning from all vertices in $W_i(G)$, i.e., a* uniform *winning strategy.*

2. *The winning regions partition V. This is called* determinacy*.*

3. *The positional-winning regions partition V. This is called* positional determinacy*.*

4. *The complexity of the algorithm is $O(|V||E|)$ (prove it can be done in $O(|E|)$).*

Question 11. *Show how to solve the game in which player 0 must, in at least one step, reach T, i.e., the objective consists of all sequence $\pi = \pi_0\pi_1\ldots$ such that there exists $\pi > 0$ such that $\pi_i \in T$. Write $Att_0^+(X)$ for the attractor strategy.*

Question 12. *A safety game is one in which a player should not leave a set, i.e.,* $\textsc{Safe}(S)$ *consists of all sequences π such that $\pi_i \in S$ for all i. Show how to solve safety games, and that they are positionally determined.*

Question 13. *Let r be a regular expression over the vertices V of an arena A. Define the objective Obj_r to be the set of plays of A for which some prefix matches r.*

Show how to solve such games...

1. *convert r into a DFW $M = (Q, \iota, \delta, F)$ (it is important the automaton is deterministic, why?)*

2. *form the product arena $A \times R$, i.e., vertices $V \times Q$, edges (v, q) to (v', q') if $E(v, v')$ and $\delta(q, v) = q'$.*

3. *player i's vertices are $V_i \times Q$.*

4. *new objective Obj_M is all sequences $(v_0, q_0)(v_1, q_1)$... such that $q_n \in F$ for some n.*

Claim: player 0 has a winning strategy in (A, Obj_r) from v iff player 0 has a winning strategy in $(A \times R, Obj_M)$ from $(v, iota)$.

Moreover: "strategies transfer", and thus (A, Obj_r) is positionally determined.

What is the complexity? (I guess EXPTIME-complete... upper bound is easy (subset construction); lower-bound?)

3.1 Repeated reachability

Definition 8. *A set $X \subseteq V$ is a* trap for player i *if player $1-i$ has a strategy σ such that every play consistent with σ starting in X stays in X.*

Note that $V \setminus Att_i(T)$ is a trap for player i.

Lemma 1. *If X is a trap then the graph restricted to vertices in X is an arena (i.e., total).*

Definition 9. *The objective of a repeated-reachability game,* $\textsc{Büchi}(T)$*, consists of all sequence π such that infinitely many i have $\pi_i \in T$.*

Theorem 3. *Büchi games:*

1. *Positionally determined (i.e., the positional-winning regions partition V).*

2. *Player i has a positional strategy that is winning from all vertices in $W_i(G)$.*

3. *Solving can be done in* PTIME*, i.e., $O(|V| \times |E|)$.*

Proof. Let G be a repeated-reachability game.

- Let $V' = V \setminus \mathrm{Att}_0(T)$.
- If $V' = \emptyset$ then $\mathrm{WR}_0(G) = V$ and $\mathrm{WR}_1(G) = \emptyset$.
- Otherwise, let $V'' = V \setminus \mathrm{Att}_1(V')$.
- By recursion solve the game on arena G'' to get $\mathrm{WR}_0(G'')$ and $\mathrm{WR}_1(G'')$.
- The winning regions in G are: $\mathrm{WR}_0(G'')$ for player 0, and $\mathrm{WR}_1(G'') \cup \mathrm{Att}_1(V')$ for player 1.

Note that V'' is non-empty, and induces a sub-arena (i.e., no dead-ends) G'': edge set $(V'' \times V'') \cap E$, target set $T \cap V''$. The size of $|V''| < |V|$.

Let $T(n)$ be the maximum number of steps this algorithm takes on arenas with n vertices. Then $T(1) = 1$, and $T(n) \leq T(n-1) + O(|E|)$. Thus $T(n) = O(|V||E|)$.

We claim that $\mathbf{WR}_0(G'') = \mathbf{WR}_0(G)$ and $\mathbf{WR}_1(G'') \cup \mathbf{Att}_1(V') = \mathbf{WR}_1(G)$ By squeezing, it is enough to show $\mathrm{WR}_0(G'') \subseteq \mathrm{WR}_0(G) \subseteq \mathrm{WR}_1^c(G) \subseteq \mathrm{WR}_0(G'')$.

Clearly, $\mathrm{WR}_0(G'') \subseteq \mathrm{WR}_0(G)$. Why? Because G'' is a trap for player 1 in G, i.e., being the complement of an attractor, every edge $(v, w) \in E$ (not E''!) with $v \in V'' \cap V_1$ has $w \in V''$.

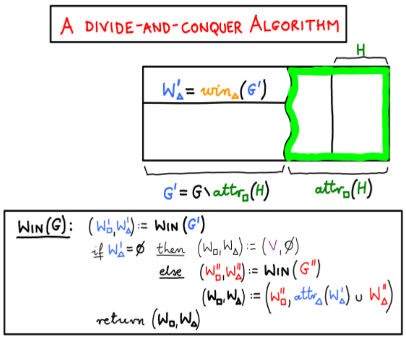
Clearly, $\mathrm{WR}_1(G'') \cup \mathrm{Att}_1(V') \subseteq \mathrm{WR}_1(G)$. Why? First, from $v \in \mathrm{Att}_1(V')$, Player 1 can ensure that from some point on T is seen only finitely often (attract to V', and then avoid T). Second, from $v \in \mathrm{WR}_1(G'')$, Player 1 plays a winning strategy for G''. If the play never leaves G'' then T is seen finitely often. If the play leaves G'' then we are in the first case $v \in \mathrm{Att}_1(V')$, and from that point on v visits T finitely often.

□

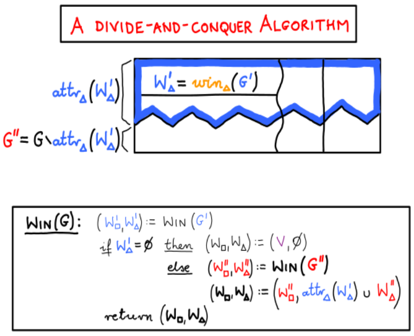
4 Parity games

A *colored-arena* is of the form (A, c) where $c : V \to \mathbb{N}$ is the coloring function. *Parity Objective* = max parity even.

Show that we can define Buchi conditions as parity conditions. Which set of colors suffice? What about co-Buchi? What is the objective if the colors are $\{1, 2, 3, 4\}$?


A DIVIDE-AND-CONQUER ALGORITHM
H
$W'_\triangle = win_\triangle(G')$
$G' = G \setminus attr_\square(H)$
$attr_\square(H)$

```
WIN(G):  $(W'_\square, W'_\triangle) := $ WIN $(G')$
         if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
                              else $(W''_\square, W''_\triangle) := $ WIN $(G'')$
                                   $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
         return $(W_\square, W_\triangle)$
```


A DIVIDE-AND-CONQUER ALGORITHM
$attr_\triangle(W'_\triangle)$
$W'_\triangle = win_\triangle(G')$
$G'' = G \setminus attr_\triangle(W'_\triangle)$

```
WIN(G):  $(W'_\square, W'_\triangle) := $ WIN $(G')$
         if $W'_\triangle = \emptyset$ then $(W_\square, W_\triangle) := (V, \emptyset)$
                              else $(W''_\square, W''_\triangle) := $ WIN $(G'')$
                                   $(W_\square, W_\triangle) := (W''_\square, attr_\triangle(W'_\triangle) \cup W''_\triangle)$
         return $(W_\square, W_\triangle)$
```

Theorem 4. *Parity games:*

1. *Positionally determined (i.e., the positional-winning regions partition V).*

2. *Player i has a positional strategy that is winning from all vertices in $W_i(G)$.*

3. *Solving can be done in* NP*and in co-*NP*, and in time $O(n^{d+O(1)})$.*

Proof. Induction on $|V|$.

Let d be highest priority, say even, and let $D = c^{-1}(d)$.

Let $A = Att_0(D)$.

1. If $A = V$ then apply attractor strategy to get to A, and from A take any step.

2. If $A \subset V$, then consider the subarena $V \setminus A$. By induction it is uniform positionally determined. Let W_i be the WRs of this subarena. There are two subcases.

   (a) $W_1 = \emptyset$, i.e., $W_0 = V \setminus A$. In this case $WR_0 = V$. Indeed, define σ_0: if in A attract to d; if in D do an arb step; if in $V \setminus A$ play given winning strategy in $V \setminus A$.

   Take player π consistent with σ_0. Two cases: π is in A infinitely often (then it sees D infinitely often); π is in $V \setminus A$ from some point on, so this suffix has max parity even. So original play has max parity even.

   (b) $W_1 \neq \emptyset$. Let $V' = Att_1(W_1)$ (in G).

   Then: $V' \subseteq WR_1$, and the following strategy is memoryless: while in W_1 play the corresonding ws (note that this is a trap for player 0, so the play does not leave W_1), and while in $V' \setminus W_1$, attract to W_1.

Let $V'' = V \setminus V'$. Since V'' is a trap for player 1, it is a subarena. Solve this recursively to get its winning regions W_0'' and W_1''.

Then: $W_0'' \subseteq WR_0$, and the strategy on W_0'' is memoryless by induction (and play stays in W_0'', so is winning in the original game).

Also: $W_1'' \subseteq WR_1$, and the following strategy is memoryless: use the strategy “while in W_1'' play the corresponding ws“. Note that if a generated play stays in W_1'' it is won by player 1, but if it ever leaves (to the choice of player 0) it can only leave to V', and thus is also won by player 1.

□

Computing winning region can be done in time $O(2^n)$; just use the recurrence $T(1) = 1$ and $T(n) \leq 2T(n-1) + O(m)$.

More careful analysis gives $O(n^{d+O(1)})$. Recurrence: $T(n, d) \leq n \times T(n, d-1) + O(n^2)$ and $T(n, 1) = O(1)$.

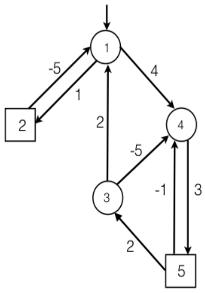
SHOW: solitaire parity games solvable in ptime.

if all nodes belong to player i, find a cycle with largest priority $i mod 2$.

idea: suppose i, d are even. check if there is a path from initial state to a node labeled d and a path from that node to itself. if yes, done. if not, remove all nodes labeled d and $d-1$, and repeat.

5 Games with quantitative objectives

Outline Games with quantitative objectives assign a real number to every play, and one player is trying to maximise this value while the other is trying to minimise it. We will discuss how to solve mean-payoff games.



edge-colored arena $(A, w : E \to [-W, +W])$.

MP-Objective is all $\pi = \pi_0\pi_1 \cdots$ such that:

$$(\lim_{n\to\infty} \inf avg_n) \geq 0$$

where $avg_n = \frac{1}{n}\sum_{i=1}^{n}(w(\pi_{i-1}, \pi_i))$.

Some calculations: consider play $[12]^\omega$. The sequence of weights is $[1, -5]^\omega$. Thus the average for n even is $\frac{-4x}{2x} = -2$, and the average for n odd is $\frac{-4x+1}{2x} \to -2$. Thus the mp is -2.

consider the play $1[453]^\omega$. The sequence of weights is $4[3,2,-5]^\omega$. The average for $n \equiv 1 \mod 3$ is $4/n$ which tends to 0. Similarly for the others. So the mp is 0.

Consider the play $[1453]^\omega$. The sequence of weights is $[4,3,2,2]^\omega$. The average for $n \equiv 0 \mod 4$ is $11/4$. Similarly the others tend to $11/4$. so the mp is $11/4 = 2.75$.

Theorem 5. *MPG are positionally determined, using uniform strategies, and solving them is in* NP *and co-*NP.

Proof is postponed till next lecture.

Meanwhile:

Proposition 1. *Solving Parity games can be reduced to mean-payoff games in polynomial time.*

Proof. WLOG, colors are in $[d-1]$. Define weight of (v,w) to be $(-1)^i n^i$ where $c(v) = i$ and $n = |V|$. In particular, supposing d is even, the weights are from the set $\{-n^{d-1}, n^{d-2}, ..., -n, 1\}$.

It is sufficient to show (*) that for every pair of memoryless strategies, the same strategy wins both games: indeed, suppose pl i has a ws in PG. Then by mem det of PG this can be taken to be a memless strategy. If pl i has no ws in MPG then by memdet pl $1-i$ does, and this can be taken to be memless. By (*) for this pair of memless strategies, the same strategy wins both games. but this is impossible.

To prove (*), fix a pair of memless strategies, and note that a cycle C is formed which governs the winner. Note that $avg(C) \geq 0$ iff $sum(C) \geq 0$. We show that the max col c on C is even iff $sum(C) \geq 0$. Let x be weight with largest absolute value.

Suppose max col on C is even. Then $x > 0$, and so

$$sum(C) \geq x - (n-1)x/n = x/n > 0$$

since there are at most $n-1$ other edges on the cycle, each of whose weight is at most x/n. The other direction is symmetric. □

6 First Cycle Games

We already saw a game that was not positionally determined. Can we characterise those games that are? We introduce FCG. To motivate them notice that if players can win positionally, then fixing positional strategies, the resulting play is a lasso, and who wins (for prefix independent games) depends only on this cycle.

Definition 10. *A* Edge-colored arena *is a tuple* $(A, \lambda : E \to \mathbb{U})$.

A cycle property Y *is a subset of* $\mathbb{U}^*$.

A first-cycle objective wrt Y, written $FC(Y)$ *consists of all plays* π *whose first cycle* $C_1(\pi)$ *is in* Y.

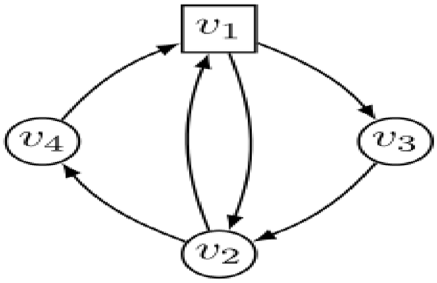
A first-cycle game (FCG) *is a game with a first-cycle objective.*

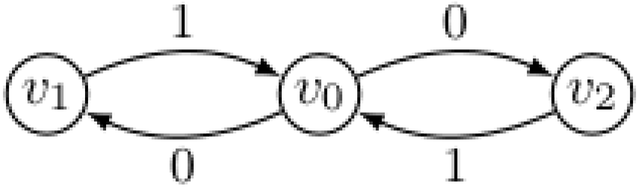
Remark: we transform vertex labeled games into edge labeled games by labeling an edge (u,v) by the label of its source v.

Finitise the following objectives: Buchi, Parity, MP.

Example 3.
- *the cycle should contain an edge from T (i.e., $u \in Y$ iff $u_i \in T$ for some i)*
- *the largest priority occuring on the cycle is even (i.e., $u \in Y$ iff $\max_i c(u_i)$ is even)*
- *the average of the weights on the cycle is non-negative (i.e., $u \in Y$ iff $avg(u) \geq 0$).*

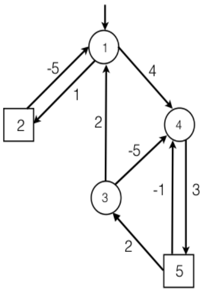
Fact 1. first-cycle games are determined (why?).
Fact 2. Not necc. positionally determined (use even-length)
Fact 3. Even if positionally determined, not necc. uniform positionally determined (see diagram with property 'ends in 1').





Definition 11. *An* edge-path *is a sequence of edges $e_1e_2\cdots$ such that $trg(e_i) = src(e_{i+1})$ for all i. An edge-path is* simple *if $i \neq j$ implies $trg(e_i) \neq trg(e_j)$. Extend λ to sequences, i.e., $\lambda(se) = \lambda(s)\lambda(e)$ for $s \in E^*, e \in E$.*

The *cycles-decomposition of an edge-path* takes as input a (usually empty) simple path s, which is treated as the initial contents of a stack, and a path π (finite or infinite). At step $j \geq 1$, the edge π_j is pushed onto the stack and if, for some k, the top k edges on the stack form a cycle, this cycle is output, then popped, and the procedure continues to step $j + 1$.



Example 4. *What are the sequence of cycles output by the cycles decomposition of*

$$(1,2)(2,1)(1,2)(2,1)(1,4)[(4,5)(5,3)(3,4)]^\omega$$

Algorithm 1 Cycles-Decomposition $CD(s,\pi)$

```
Require: $s$ is a finite (possibly empty) simple path          ▷ initial stack content
Require: $\pi$ is a finite or infinite path $\pi_1\pi_2\cdots$          ▷ the path to decompose
Require: If $s$ is non-empty then $\mathsf{trg}(s) = \mathsf{src}(\pi)$          ▷ $s\pi$ must form a path
  $step = 1$
  while $step \le |\pi|$ do                                    ▷ Start a step
      Append $\pi_{step}$ to $s$                               ▷ Push current edge into stack
      Say $s = e_1e_2\cdots e_m$
      if $\exists i : e_ie_{i+1}\cdots e_m$ is a cycle then     ▷ If stack has a cycle
          Output $e_ie_{i+1}\cdots e_m$                         ▷ output the cycle
      $s := e_1\cdots e_{i-1}$                                  ▷ Pop the cycle from the stack
      end if
      $step := step + 1$                                       ▷ advance to next input edge
  end while
```

Definition 12. *Let Y be a cycle property. For an infinite path π, let $first(\pi)$ be the first cycle output by $CD(\epsilon,\pi)$. The* first-cycle objective based on Y, written *$FC(Y)$ is the set of plays such $\lambda(first(\pi)) \in Y$.*

Define $N_z(\pi) \in \mathbb{N}\cup\{\infty\}$ to be the index of the first edge that starts with z, if one exists. Define $head_z(\pi)$ to be the prefix of π before $N_z(\pi)$, and $tail_z(\pi)$ to be the suffix of π starting at $N_z(\pi)$.

Formally, $N_z(\pi) := \infty$ if z does not occur on π, and otherwise $N_z(\pi) := \min\{j : \mathsf{src}(\pi_j) = z\}$. Also, $head_z(\pi) := \pi[1, N_z(\pi) - 1]$ and $tail_z(\pi) := \pi[N_z(\pi), |\pi|]$. By convention, if $N_z(\pi) = \infty$ then $head_z(\pi) = \pi$ and $tail_z(\pi) = \epsilon$.

We now define a game, that is very similar to the first-cycle game, except that one of the nodes z of the arena is designated as a "reset" node... the first time play sees z (if at all), the history is erased:

Definition 13. *Fix an arena A, a vertex $z \in V$, and a cycle property Y. Define the objective $FC_z(Y)$ to consist of all plays π satisfying the following property: if $head_z(\pi)$ is not a simple path then $first(\pi) \in Y$, and otherwise $first(tail_z(\pi)) \in Y$.*

Playing the game with objective $FC_z(Y)$ is like playing the first-cycle game over Y, however, if no cycle is formed before reaching z for the first time, the prefix of the play up to that point is ignored. Thus, in a sense, the game is reset. Also note that if play starts from z, then the game reduces to a first-cycle game.

Definition 14. *Fix Y. An arena is Y-*resettable *if for every $i \in \{0,1\}$, and every node z, we have that $WR^i(A, FC_z(Y)) = WR^i(A, FC(Y))$.*

Theorem 6 (Resettability implies memoryless determinacy)**.** *Suppose that every arena A is Y-resettable. Then every game $(A, FC(Y))$ is memoryless determined.*

The proof relies on the fact that we may assume that a strategy of $(A, FC_z(Y))$ makes the same move every time it reaches z.

Definition 15 (Forgetful at z from v)**.** *Fix an arena A, a vertex $v \in V$, a Player $i \in \{0,1\}$, and a vertex $z \in V_i$ belonging to Player i. We call a strategy T for Player i* forgetful at z from v *if there exists $z' \in V$ such that $(z, z') \in E$ and for all $\pi \in plays(T, v)$, and all $n \in \mathbb{N}$, if $\mathsf{src}(\pi_n) = z$ then $\mathsf{trg}(\pi_n) = z'$.*

Lemma 2 (Forgetful at z from v)**.** *Fix an arena A, a vertex $v \in V$, a Player $i \in \{0,1\}$, and a vertex $z \in V_i$ belonging to Player i. In the game $(A, FC_z(Y))$, if Player i has a strategy S that is winning from v, then Player i has a strategy T that is winning from v and that is forgetful at z from v.*

Sketch. The second time z appears on a play, the winner is already determined, and so the strategy is free to repeat the first move it made at z. On the other hand, the first time z appears on a play, the strategy can make the same move regardless of the history of the play before z, because the winning condition ignores this prefix. □

Sketch of Theorem 6. A node $z \in V$ is a *choice node* of an arena B, if there are at least two distinct vertices $v', v'' \in V$ such that $(z, v') \in E^B$ and $(z, v'') \in E^B$.

For each $i \in \{0,1\}$, we induct on the number of choice nodes of player i, i.e., the kth inductive hypothesis says that in every arena with k choice nodes of player i, if player i has a winning strategy from v, player i also has a memoryless winning strategy from v.

Base case ($k = 0$): this is immediate since there is a single strategy for player i, which is memoryless.

Inductive step ($k > 0$). Consider an arena A with $k > 0$ choice nodes for player i, and suppose player i has a winning strategy in $(A, FC(Y))$ from v. Let z be a choice node for Player i.

- By the resettability assumption applied to A, Player i has a winning strategy from v in $(A, FC_z(Y))$.
- By Lemma 2, Player i has a strategy S that is winning from v and that is also forgetful at z from v. Thus we may form a sub-arena B of A by removing all edges from z that are not taken by S. Observe that S is winning from v in $(B, FC_z(Y))$.
- Applying the resettability assumption to B, Player i has a winning strategy from v in $(B, FC(Y))$.

- But B has less choice nodes for Player i, and thus, by induction, Player i has a memoryless winning strategy from v in $(B, FC(Y))$.

This memoryless strategy is also winning from v in A (since we only removed choices of player i, which are not used in this memoryless strategy). □

Connection with infinite duration games

We now define the connection between first-cycle games and games of infinite duration (such as parity games, etc.), namely the concept of Y-greedy games. We then prove the Strategy Transfer Theorem, which says, roughly, that for every arena, the winning regions of the First-Cycle Game over Y and a Y-greedy game coincide, and that memoryless winning strategies transfer between these two games.

Definition 16 (Greedy)**.** *Fix an arena A. The* all-cycle objective based on Y, written $AC(Y)$ *is the set of plays π of A such that the labelling of every cycle output by $CD(\epsilon, \pi)$ is in Y.*

Say that a game (A, O) is Y-greedy *if*

$$AC(Y) \subseteq O \text{ and } AC(\neg Y) \subseteq \neg O.$$

The following lemma says that only finitely many edges in a path are pushed but never popped. In particular, at most $|V| - 1$ edges:

Lemma 3. *For every path π in arena A with vertex set V, there are at most $|V| - 1$ indices i such that π_i does not appear in any of the cycles in cycles(π).*

Example 5. *1. $(A, AC(Y))$ is Y-greedy.*

2. Every parity game is Y-greedy where Y says "the largest occuring color is even".

3. Every game with mean-payoff winning condition is Y-greedy where Y says "the average of the cycle is non-negative" (HW)

We state the Strategy Transfer Theorem:

Theorem 7 (Strategy Transfer)**.** *Let (A, O) be a Y-greedy game, and let $i \in \{0, 1\}$.*

1. $WR^i(A, O) = WR^i(A, FC(Y))$ (so, in particular, (A, O) is determined).

2. For every memoryless strategy S for Player i, and vertex $v \in V$: S is winning from v in the game (A, O) if and only if S is winning from v in the game $(A, FC(Y))$.

To prove the Strategy Transfer Theorem we need a lemma that states that one can pump a strategy S that is winning for the first-cycle game to get a strategy $S^{\circlearrowleft}$ that is winning for the all-cycles game.

The strategy $S^{\circlearrowleft}$ says to **follow S until a cycle is formed, remove that cycle from the history, and continue**.

Roughly, $S^{\circlearrowleft}(u) = \mathit{stack}(u)$ where $\mathit{stack}(u)$ is the stack content of CD after processing path u. The fact that every winning strategy in the first-cycle game of Y can be pumped to obtain a winning strategy in a Y-greedy game, is why we call such games "greedy".

First, we need notation to go from edge-paths to node-paths and back:

- For simple path $\pi = e_1 e_2 \cdots e_l \in E^*$ write $nodes(\pi)$ for $\mathsf{src}(e_1)\mathsf{src}(e_2)\cdots\mathsf{src}(e_l)\mathsf{trg}(e_l) \in V^*$.
- For path $u = v_1 v_2 \cdots v_l \in V^*$ with $l \geq 2$, write $edges(u)$ for $(v_1, v_2)(v_2, v_3)\cdots(v_{l-1}, v_l) \in E^+$.

Instead of writing these transformations we assume them implicitly.

Definition 17 (Pumping Strategy). *For a finite path $\pi \in E^*$, the stack content at the end of $CD(\epsilon, \pi)$ (Algorithm 1) is denoted $stack(\pi)$.*

Fix an arena A, a Player $i \in \{0, 1\}$, and a strategy S for Player i. Define the pumping strategy of S *be the strategy $S^{\circlearrowleft}$ on history $u = u_1 \ldots u_k$ ending in a node of player i, as follows:*

$$S^{\circlearrowleft}(u) = \begin{cases} S(u_1) & \text{if } k = 1 \\ S(u_k) & \text{if } k > 1, stack(u) = \epsilon \\ S(stack(u)) & \text{otherwise.} \end{cases}$$

Note that $S^{\circlearrowleft}$ is well-defined since: if $stack(u) \neq \epsilon$ then $stack(u)$ ends with $u_k \in V_i$ and so $stack(\pi)$ is in the domain of S.

Lemma 4 (equal). *If S is memoryless then $S^{\circlearrowleft} = S$.*

Proof. We need to show that $S^{\circlearrowleft}(u) = S(u)$ for all u. If $|u| = 1$ then this is immediate by construction. If $stack(u) = \epsilon$ then $S^{\circlearrowleft}(u) = S(u_k) = S(u)$ since S is memoryless. If $stack(u) \neq \epsilon$ then $last(stack(u)) = last(u)$. So $S^{\circlearrowleft}(u) = S(stack(u)) = S(u)$ since S is assumed memoryless. □

Lemma 5 (pumping). *Fix Player $i \in \{0, 1\}$ and let (A, O) be a Y-greedy game. If S is a strategy for Player i that is winning from v in $(A, FC(Y))$ then $S^{\circlearrowleft}$ is winning from v in (A, O).*

Proof. The strategy $S^{\circlearrowleft}$ says to follow S, and when a cycle is popped by CD, remove that cycle from the history and continue. Thus, for every cycle C that is popped, let l be the time at which the first edge of C is being pushed, and note that the stack up to time l followed by C is a path consistent with S whose first cycle is C.

Thus if S is a strategy of player 0 that is winning from v in the game $(A, FC(Y))$ then for every play $\pi \in plays(S^{\circlearrowleft}, v)$, every cycle in $cycles(\pi)$ is in Y. By definition of Y-greedy, this means that $S^{\circlearrowleft}$ is winning from v in the game (A, O). The case of player 1 is symmetric. □

Proof of Strategy Transfer Theorem. Let Y be a cycle property and A an arena. Suppose that (A, O) is Y-greedy. We begin by proving the first item. Use Lemma 5[pumping] to get that for $i \in \{0, 1\}$,

$$\mathrm{WR}^i(A, FC(Y)) \subseteq \mathrm{WR}^i(A, O).$$

Since first-cycle games are determined, the winning regions $\mathrm{WR}^0(A, FC(Y))$ and $\mathrm{WR}^1(A, FC(Y))$ partition V. Thus, since $\mathrm{WR}^0(A, O)$ and $\mathrm{WR}^1(A, O)$ are disjoint, the containments above are equalities, as required for item 1.

We prove the second item. Suppose S is a memoryless strategy for player 0 and recall that $S = S^{\circlearrowleft}$ (by Lemma 4[memless]).

- Suppose S is winning from v in the game $(A, FC(Y))$. Then it is winning from v in the game (A, O) (By Lemma 5).
- Suppose S is not winning from v in the game $(A, FC(Y))$. Since S is memoryless, plays of A consistent with S are exactly infinite paths in the induced sub-arena $A^{||S}$. Hence, there is a path π in the induced solitaire arena $A^{||S}$ for which the first cycle, say C, satisfies $\neg Y$. Define the infinite path π' which pumps this cycle forever. Being a path in $A^{||S}$, it is a play of A consistent with S. Moreover, π' has the property that every cycle in its cycles-decomposition (i.e., C) satisfies $\neg Y$. Since (A, O) is Y-greedy, S is not winning from v in the game (A, O).

The case that S is a strategy for player 1 is symmetric. □

Putting together we get:

Theorem 8. *If every arena is Y-resettable then every Y-greedy game is memoryless determined.*

Recipe for positional determinacy

Question 14. *How to check if every arena is Y-resettable?*

Definition 18. *Fix an arena A. Let $TAC(Y)$ consist of all plays π of A such that some suffix of π is in $AC(Y)$. An arena A is Y-*unambiguous *if there is no play of A that is in $TAC(Y) \cap TAC(\neg Y)$.*

Example 6. *If A is Y-unambigous then $(A, TAC(Y))$ is Y-greedy. Why? clearly $AC(Y) \subseteq TAC(Y)$. Also $AC(\neg Y) \subseteq TAC(\neg Y) \subseteq \neg TAC(Y)$ by assumption.*

Theorem 9 (unambigous implies resettable)**.** *Every arena that is Y-unambiguous is also Y-resettable.*

Proof. First, if A is Y-unambigous then $(A, TAC(Y))$ is Y-greedy (above). Thus by Theorem 7 the winning regions of $(A, FC(Y))$ and $(A, TAC(Y))$ coincide.

Second, we now show that the winning regions of $(A, FC_z(Y))$ and $(A, TAC(Y))$ coincide. As usual, it is sufficient to show containment, i.e., $WR^i(A, FC_z(Y)) \subseteq WR^i(A, TAC(Y))$ for $i = 0, 1$.

So, suppose player i has a winning strategy S from v in $(A, FC_z(Y))$.

It is sufficient to define a strategy T for player i such that every play consistent with T is in $TAC(Z)$ where $Z = Y$ if $i = 0$ and $Z = \neg Y$ if $i = 1$ (why? if $i = 0$ then this is what we want. if $i = 1$ then every play consistent with T is in $TAC(\neg Y) \subseteq \neg TAC(Y)$, as required).

There are two cases:

1. there is no simple path consistent with S that ends in z. Thus S is winning for $FC(Z)$, so $T = S^{\circlearrowleft}$ is winning for $AC(Z)$ and thus for $TAC(Z)$.
2. o/w let h be a simple path consistent with S ending in z. Define

$$T(u) := \begin{cases} S^{\circlearrowleft}(u) & \text{if } z \text{ does not appear on } u, \\ (S_z)^{\circlearrowleft}(tail_z(u)) & \text{otherwise.} \end{cases}$$

where $S_z(u) = S(hu)$ if u starts in z (and, otherwise arbitrarily).

In words, T behaves like the pumping strategy $S^{\circlearrowleft}$. Once (and if) z is reached, T erases all its memory and switches to $(S_z)^{\circlearrowleft}$ (which itself is winning from z in the FCG). Note that T is winning $TAC(Z)$: if z never appears then every cycle is in Z; if z does appear, then starting at that point in time, every cycle is in Z. □

Question 15. *Ok, so how to check if A is Y-unambigous?*

Definition 19. *A* winning condition *is a set $W \subseteq \mathbb{U}^\omega$. On an edge-coloured arena $(A, \lambda : E \to \mathbb{U})$ it determines an objective, i.e., $O(W)$ consisting of all plays π in A such that $\lambda(\pi) \in W$.*

Example 7.

1. *The parity winning condition over $\mathbb{U} = [N]$ consists of all sequences $\alpha \in [N]^\omega$ such that $maxinf(\alpha)$ is even.*

2. *The mean-payoff winning condition over $\mathbb{U} = [-N, N]$ consists of all sequences α such that $\lim_n \inf avg_n(\alpha) \geq 0$.*

Lemma 6. *Suppose W and $\neg W$ are prefix-independent, and consider an arena A and a cycle property Y. If $(A, O(W))$ is a Y-greedy game then A is Y-unambiguous.*

Proof. Suppose not. Take a play π of some arena such that some suffix $\pi' \in AC(Y) \subseteq O(W)$ and some suffix $\pi'' \in AC(\neg Y) \subseteq \neg O(W)$. But one of these is a suffix of the other, say π' is a suffix of π'' (the other case is symmetric). Then $\lambda(\pi') \in W$ is a suffix of $\lambda(\pi'') \notin W$. Contradiction. □

But prefix-independence is typically easy to check!

To conclude that every game with winning condition W is memless determined:

1. Check that W and $\neg W$ are prefix independent.

2. Finitise the winning condition W to get a cycle property Y.

3. Show that every game $(A, O(W))$ is Y-greedy.

Example 8. *Apply the recipe to Büchi, parity-, mean-payoff winning conditions. e.g., Let W be the buchi condition. it and its complement are prefix-independent.*

What if the winning-condition is not prefix independent?

1. Say that Y is *closed under cyclic permutations* if $ab \in Y$ implies $ba \in Y$, for all $a \in \mathbb{U}, b \in \mathbb{U}^*$.

2. Say that Y is *closed under concatenation* if $a \in Y$ and $b \in Y$ imply that $ab \in Y$, for all $a, b \in \mathbb{U}^*$.

Theorem 10. *Let Y be a cycle property. If Y is closed under cyclic permutations, and both Y and $\neg Y$ are closed under concatenation, then every arena is Y-unambigous (and therefore Y-resettable).*

To conclude that $(A, O(W))$ is uniform memless determined:

1. Finitise the winning condition W to get a cycle property Y.
2. Check that Y is closed under cyclic permutations and both Y and $\neg Y$ are closed under concatenation.
3. Check that $(A, O(W))$ is Y-greedy.

We conclude with a more sophisticated use of the recipe, applied to the initial credit problem of energy games.

Definition 20. *The* Energy winning-condition with initial credit r, *written* $W_r \subseteq \mathbb{Z}^*$, *consists of all sequences* α *such that* $r + \sum_{1 \leq i \leq n} \alpha_i \geq 0$ *for all* n.

Theorem 11. *Either there is an initial credit with which Player* 0 *(the "energy" player) wins, or for every initial credit Player* 1 *wins. In both cases, we show that the winner can use a memoryless strategy.*

Proof. Finitise "energy" as $Y \subset \mathbb{Z}^*$ which says that the sum of the numbers is non-negative. This property is clearly closed under cyclic permutations and concatentation, and its complement (the sum is negative) is also clearly closed under concatenation. Also $(A, AC(Y))$ is Y-greedy, and thus memoryless determined.

Claim: if σ is a winning strategy for player 0 in $(A, AC(Y))$ then σ is a winning strategy in the energy game on A for some initial credit r.

Indeed, let $r = -t(|V| - 1)$, where t is the minimum amongst the negative weights of the arena. Let π be consistent with σ, and consider some prefix π' of it. By Lemma 3, at most $|V| - 1$ edges are not on $cycles(\pi')$, and thus the energy level at the end of π' is at least the initial credit plus $t(|V| - 1)$. Hence, an initial credit of r suffices for π to be winning for Player 0 also in the energy game.

Claim: if σ is a winning strategy for player 1 in $(A, AC(Y))$ then σ is a winning strategy in the energy game on A for all initial credits.

Indeed, observe that a winning strategy for player 1 in $(A, AC(Y))$ is also winning $(A, FC(Y))$; and thus by pumping is also winning in $(A, EC(Y))$ where $EC(Y)$, the objective for player 0, says that some cycle should be in Y. Thus every play consistent with this strategy has all cycles not in Y, and thus in the energy game the energy along the play tends to $-\infty$, and so for every initial credit player 1 wins. □

7 Games with multiple players

Outline Games with multiple players require different types of solutions. We will define Nash equilibria on games on graphs, and show how to decide their existence (and compute an equilibrium, if it exists) for certain objectives that we have already encountered.

8 Different approaches to solving games

1. PDL satisfiability (Racer, FaCT, Pellet)
2. simulation based techniques
3. safety games, GR(1), ATL
4. Directed search (Stroeder and Pagnucco 09)
5. Planning
6. Antichain
7. Incremental
8. DES
9. parity games: progress measures, rank-based, zielonka

Reflections: 1 or 2 pars email to me or to florian on reflections on the course: what worked well for you, what didn't, level of interactive, good pace, enjoyable to come to class, connected to other material you learned, learn more than the syllabus e.g., how to formalise intuitions, etc.