2. Parity Games

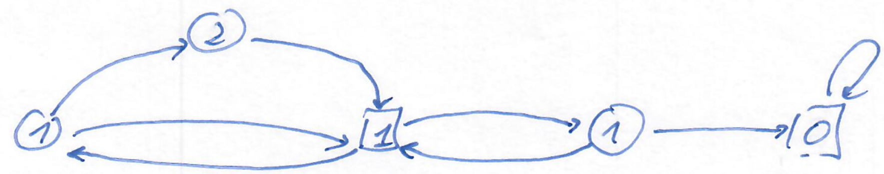
Goal:
- Introduce parity games
- State the positional determinacy result
- Show that parity games can be reduced to exponentially larger safety games.

Definition:

A parity game is a tuple $\mathcal{G} = (V_0, V_1, E, \Omega)$,

where
- $V_0 \cap V_1 = \emptyset$,
- $E \subseteq V \times V$ with $V = V_0 \cup V_1$, and
- $\Omega : V \to \{0, \dots, M\}$, $M \in \mathbb{N}$, is a priority function.

Example: $\mathcal{G}$



$\bigcirc := V_0$

$\square := V_1$.

The semantics is defined in terms of plays, played between Player 0 and Player 1.
When the play is in position q, the owner of q ($q \in V_0$ or $q \in V_1$) decides about the next position to move to.
This yields an infinite path through the arena, the play.

Definition:

Let $\mathcal{G} = (V_0, V_1, E, \Omega)$ with $V = V_0 \cup V_1$.
- A play is an infinite sequence $v_0, v_1, \dots \in V^\omega$.

- The <u>play is won</u> by Player 0,
  if
  the highest priority that occurs infinitely often in $\Omega(v_0), \Omega(v_1), \ldots$ is even.

  Otherwise, the play is won by Player 1.

In game theory, one is not interested in winning single plays. One is interested in having a strategy to win, no matter how well the opponent plays.

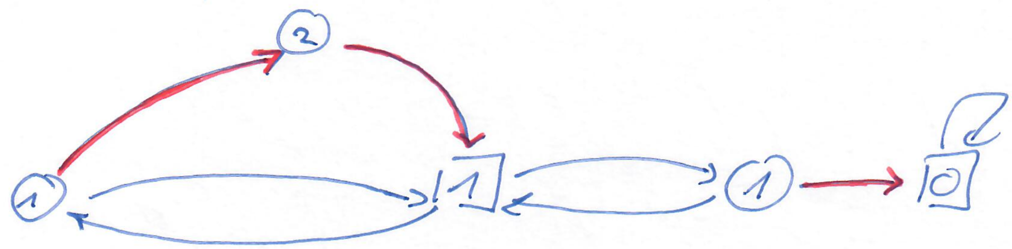
<u>Definition:</u>

- A <u>strategy for Player i</u> is a function
  $$\sigma : V^* V_i \longrightarrow V$$
  that takes the prefix of a play ending in a position of Player i, and returns the next position to move to.
- A strategy is <u>positional</u>, if it can be written as
  $$\sigma : V_i \longrightarrow V,$$
  i.e. the next position only depends on the current position, not on the whole prefix of the play.
- A play $v_0 v_1 \ldots$ <u>conforms to</u> a strategy σ, if for all $v_i \in V_i$, we have
  $$v_{i+1} = \sigma(v_0 \ldots v_i).$$
- A <u>strategy</u> σ is <u>winning from position</u> v, if Player i wins every play starting in position v

that conforms to σ.

- Player i wins the parity game from position v, if there is a winning strategy from position v.
  The winning region $W_i \subseteq V$ is the set of all positions from which Player i wins the parity game.

Fact: $W_0 \cap W_1 = \emptyset$.



$\rightarrow$:= a positional strategy σ for Player 0.
It is winning from every position $v \in V$.

There are several important facts to know about parity games.
Here are two.

2.1 Parity Games are in $NP \cap coNP$

Goal: Give an upper bound on the complexity of this problem:

PG:
Given: Parity game $\mathcal{G}$, position v in $\mathcal{G}$, Player i.
Question: Is $v \in W_i$?

The upper bound relies on the following, which may be considered the main result about parity games.

Theorem (Emerson & Jutla, 1985, FOCS 1991):

Consider parity game $\mathcal{G}$ with positions V.

(1) Determinacy: $W_0 \cup W_1 = V$.

(2) Positional winning strategies:

Player i wins $\mathcal{G}$ from $v \in V$ iff she has a positional strategy to do so.

In fact, there is a positional strategy that is winning for all positions in W_i.

We will not prove this result but derive consequences from it.

Definition:

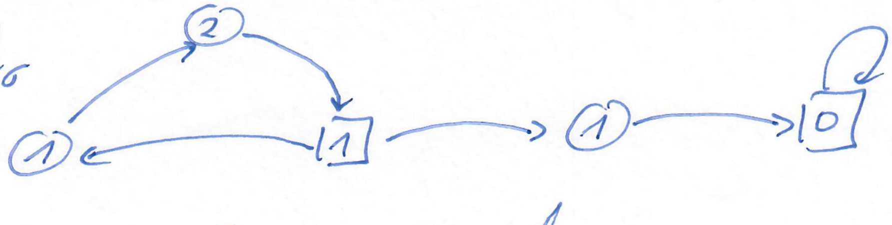
Consider a parity game $\mathcal{G} = (V_0, V_1, E, \Omega)$ and a positional strategy $\sigma : V_i \to V$.

We obtain the subgame

$$\mathcal{G}_\sigma := (V_0, V_1, E \setminus \{(v_i, v) \mid v_i \in V_i, v \neq \sigma(v_i)\}, \Omega)$$

by removing the edges from positions in V_i that do not conform to the strategy.

In the example: $\mathcal{G}_\sigma$



Lemma 1: The plays in $\mathcal{G}_\sigma$ are precisely the plays in $\mathcal{G}$ conform to σ.

Corollary (of Lemma 1):

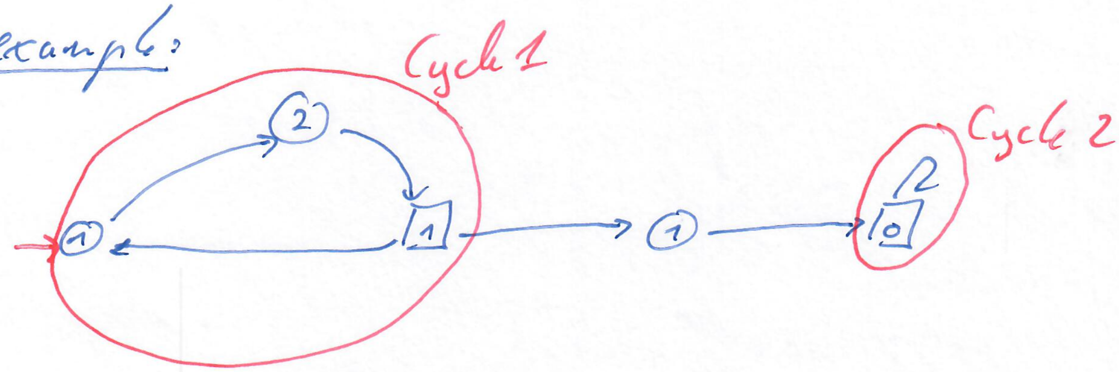
Strategy σ is winning from v for Player i

iff (Def.) all plays in $\mathcal{G}$ from v conform to σ are won by Player i.

iff (Lemma 1) all plays in $\mathcal{G}_\sigma$ from v are won by Player i.

iff (Def.) all cycles reachable from v have highest highest even.

In the example:



From the position marked →

two cycles are reachable.

Cycle 1 has highest priority 2

Cycle 2 has highest priority 0.

Since both priorities are even, position → is winning for Player 0.

With this corollary, we are ready to give an algorithm for problem PG.

Theorem: $PG \in NP \cap coNP$.

Proof: • Given G, Player i, and v,
we guess a positional strategy σ
in time linear in G.
• Then we compute G_σ, also in linear time.
• We can also verify in linear time
that the least priority on every cycle
reachable from v in G_σ is even/odd,
depending on the Player i.

• This shows $PG \in NP$.
• We obtain $PG \in coNP$ by determinacy,
i.e. if $v \notin W_i$, then $v \in W_{1-i}$
and the same algorithm works for the opponent. $\square$

2.2 Counter Reduction

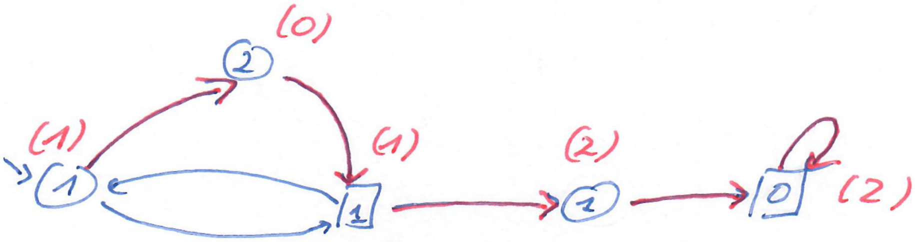
due to Bernet, Janin, Walukiewicz 2002,
better presentation Bernet, Doyen 2008.

Goal: Show that parity games can be reduced
to exponentially larger safety games (avoid a set of bad positions)

Idea: • Count the occurrences of odd positions
• If the counter exceeds the number of positions: Cycle.
• Play a safety game that keeps
the counter below the threshold.

In the example:

There is only one odd position,
track its counter as (i):



$\rightarrow$:= a concrete play
with the counter values
annotated to the positions.

Correctness:

Why should such a reduction work?
By the existence of positional winning strategies.

Definition:

An infinite sequence of priorities $p_1 p_2 \ldots \in \mathbb{N}^\omega$
is parity-n-fair, if
every infix $p_i \ldots p_j$
that contains an odd priority, say r, more than n times
also contains a priority $> r$.

Proposition:

Consider parity game $\mathcal{G}$ and position v.
Player 0 wins $\mathcal{G}$ from position v iff
she has a strategy σ so that
every play from v consistent with σ is parity-n-fair.

Proof:

"⇐" Parity-n-fair plays satisfy the parity condition.

"⇒" Consider a positional strategy σ that is winning from v.

Inspect the cycles in G_σ reachable from v. $\square$

Idea of the counter reduction — more technically:

- Build a monitoring device that checks whether the current play is parity-n-fair.
- Leave the safe states if this is not the case.

Definition:

- Define a vector of counters, one for each odd priority,

$$\bar{c} := (c_1, \dots, c_m).$$

- When entering a position with priority r, compute

$$\oplus_r(\bar{c}) = \bar{d}$$

with

$$d_i := c_i, \quad \text{if } i > r$$

$$d_i := 0, \quad \text{if } r > i$$

$$d_i := \begin{cases} c_i + 1, & \text{if } i = r \text{ and } c_i + 1 \leq n \\ \text{overflow}, & \text{if } i = r \text{ and } c_i + 1 > n. \end{cases}$$

Note: There are $O(n^{d/2})$ vectors of counters, where d is the number of priorities in G.