

Logic, games and synthesis – ESSV, Univ. Bordeaux 1, 2012/13

CONTENTS

- Anca Muscholl: 2-player games on graphs, logic and automata on infinite trees, controller synthesis, distributed games.
- Hugo Gimbert: Markov chains, stochastic games, probabilistic automata.

- Two-player games can model **open systems**, i.e., where a system faces a (hostile) environment. The question is whether the system has a strategy for satisfying a given specification, **no matter** what the environment does. **Synthesizing** a strategy (or a program) amounts to construct correct programs.
- Two-player games are connected to logics (\forall/\exists). They can be used for showing (in)expressivity results (see first-order logic and Ehrenfeucht-Fraïssé games).
- More sophisticated types of games arise in specific applications, like in stochastic, timed or **distributed systems**.

- Two-player games can model **open systems**, i.e., where a system faces a (hostile) environment. The question is whether the system has a strategy for satisfying a given specification, **no matter** what the environment does. **Synthesizing** a strategy (or a program) amounts to construct correct programs.
- Two-player games are connected to logics (\forall/\exists). They can be used for showing (in)expressivity results (see first-order logic and Ehrenfeucht-Fraïssé games).
- More sophisticated types of games arise in specific applications, like in stochastic, timed or **distributed systems**.

SOME VOCABULARY

- Games are played on **arenas**, which can be finite or infinite (words, trees, graphs, ...).
- Solving games means finding winning positions as well as **winning strategies** for the players.

NOTATIONS

- Σ finite alphabet, Σ^* set of finite words over Σ ,
- Σ^ω set of infinite words over Σ (mappings $\mathbb{N} \rightarrow \Sigma$)

GALE-STEWART GAMES

- 2 players: 0, 1
- Every player has his own alphabet: Σ_0, Σ_1 .
- A Gale-Stewart game is given by a **winning condition**

$$\text{Win} \subseteq (\Sigma_0 \Sigma_1)^\omega$$

- **Play**: infinite word $w \in (\Sigma_0 \Sigma_1)^\omega$
- Play w is **winning** for player 0 if $w \in \text{Win}$. Otherwise it is winning for player 1.
- A **strategy** for player 0 is a mapping $\sigma_0 : \Sigma_1^* \rightarrow \Sigma_0$ (or $\sigma : (\Sigma_0 \Sigma_1)^* \rightarrow \Sigma_0$).
Extend σ to $\bar{\sigma} : \Sigma_1^\omega \rightarrow \Sigma_0^\omega$ by

$$\bar{\sigma}(b_1 b_2 \dots) = a_0 a_1 \dots \text{ if } a_n = \sigma(b_1 \dots b_n), \quad n \geq 0$$

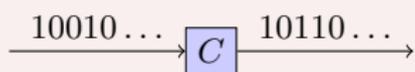
PLAYING THE GAME

- If player 1 plays $y = b_1 b_2 \dots \in \Sigma_1^\omega$ and player 0 applies σ by playing $\bar{\sigma}(y) = a_0 a_1 \dots$:
resulting play is $a_0 b_1 a_1 b_2 \dots$, denoted as $\bar{\sigma}(y) \mid y$.
- Player 0 wins the game with strategy σ if **for every** $y \in \Sigma_1^\omega$ we have $\bar{\sigma}(y) \mid y \in \text{Win}$.
- We define in the same way strategies for player 1. The winning condition for player one is $\text{Win}^{co} = \Sigma^\omega \setminus \text{Win}$.
- The game is called **determined** if either player 0 or player 1 has a winning strategy.
- **Martin's theorem** says that Gale-Stewart games are determined when Win is a Borel set.

REGULAR GAMES

- **ω -regular**: defined by **automata** (Büchi, Muller, parity, ...), or **logics** (monadic second-order), or ω -expressions, ...
- Regular (infinite) games: winning conditions are ω -regular. In particular, they are Borel. Regular games are therefore determined.

CHURCH'S PROBLEM (1957)

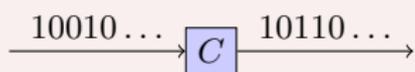


Church's problem:

given an input/output relation $R \subseteq (\{0, 1\}^2)^\omega$, construct a circuit C implementing R .

- Example 1: if the current input is 1, then the corresponding output is 1; if the input is 0 then the output is the parity of the number of 0's seen so far. A finite memory circuit C (automaton) can do this job.
- Example 2: if the current input is 1, then the output is 0; and if there are infinitely many inputs equal to 0, then infinitely many outputs are equal to 0.
The circuit that outputs only 0's does the job.
- Example 3: Infinitely many 1's in the input if and only if finitely many 1's in the output.
No circuit can do this (why?)

CHURCH'S PROBLEM (1957)



Church's problem:

given an input/output relation $R \subseteq (\{0, 1\}^2)^\omega$, construct a circuit C implementing R .

- Example 1: if the current input is 1, then the corresponding output is 1; if the input is 0 then the output is the parity of the number of 0's seen so far. A finite memory circuit C (automaton) can do this job.
- Example 2: if the current input is 1, then the output is 0; and if there are infinitely many inputs equal to 0, then infinitely many outputs are equal to 0.
The circuit that outputs only 0's does the job.
- Example 3: Infinitely many 1's in the input if and only if finitely many 1's in the output.
No circuit can do this (why?)

REMARK

Church's problem is about non-terminating computations. The specifications are MSOL (monadic second-order logic). Church's problem was solved by Büchi-Landweber in 1969.

ARENAS

$$\mathcal{A} = (V_0, V_1, E)$$

- Two players P_0 et P_1 . The set of vertices is partitioned into two disjoint subsets: V_0 belongs to P_0 and V_1 to P_1 .
- Set of edges $E \subseteq V \times V$, $V = V_0 \cup V_1$.

PLAYS

A **play** π over \mathcal{A} is a **maximal** path in \mathcal{A} :

- either an infinite path $\pi = v_0, v_1, \dots$
- or a finite path $\pi = v_0, v_1, \dots, v_n$ such that v_n has no successor.

STRATEGIES

- A strategy for P_0 is a mapping $\sigma : V^*V_0 \rightarrow V$
s.t. $\sigma(\pi v) \in vE := \{w \in V \mid (v, w) \in E\}$, with $\pi \in V^*$, $v \in V$.
- A play $\pi = v_0, v_1, \dots$ is consistent with the strategy σ if
 $v_{i+1} = \sigma(v_0 \cdots v_i)$ for all i s.t. $v_i \in V_0$.
- Different types of strategies (non-randomized): positional, finite-memory,
...

GAME

A game is given by a pair $(\mathcal{A}, \text{Win})$, where \mathcal{A} is a graph arena and $\text{Win} \subseteq V^\omega$ is the winning condition.

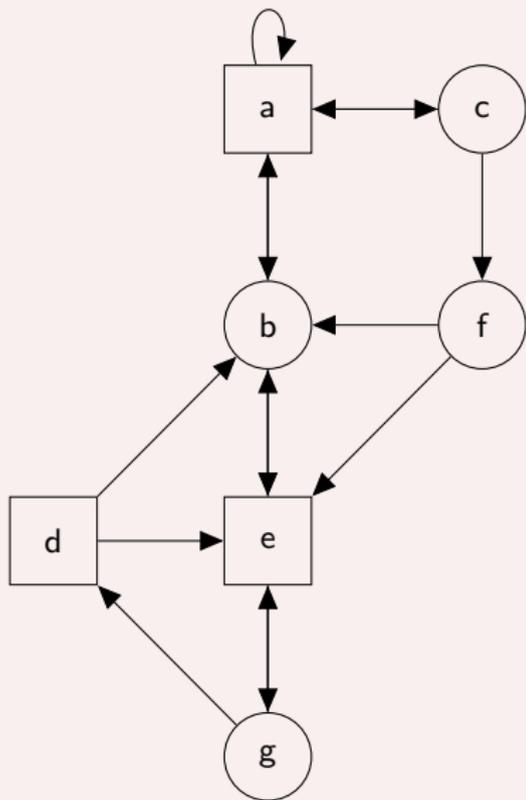
Some examples $(G, B \subseteq V, \mathcal{F} \subseteq \mathcal{P}(V))$:

- **reachability games**: $\text{Win} = V^*GV^\omega$
- **safety games**: $\text{Win} = V^\omega \setminus V^*BV^\omega$
- **obligation games**: the set of states visited during a play should belong to \mathcal{F}
- **Büchi games**: $\text{Win} = (V^*G)^\omega$, **Muller games**, **parity games**...

GALE-STEWART GAMES AND GRAPH GAMES

- Let \mathcal{A} be a finite arena such that $E \subseteq V_0 \times V_1 \cup V_1 \times V_0$ (**turn-based**) and no vertex is a dead end. Such a graph game corresponds to a Gale-Stewart game where $\Sigma_0 = \{(u, v) \in V_0 \times V_1 \mid (u, v) \in E\}$ and $\Sigma_1 = \{(u, v) \in V_1 \times V_0 \mid (u, v) \in E\}$. A regular winning condition can in particular enforce that every play corresponds to a path in \mathcal{A} .
- A Gale-Stewart game (Σ_0, Σ_1) basically corresponds to a graph game with vertices $V_0 = \{start\} \cup \Sigma_0 \times \Sigma_1 \times \{0\}$, $V_1 = \Sigma_0 \cup \Sigma_0 \times \Sigma_1 \times \{1\}$. Plays are of the form $start, a_0, (a_0, b_0), (a_1, b_0), (a_1, b_1), \dots$ and P_0 wins if $a_0b_0a_1b_1 \dots \in \text{Win}$.
If Win is regular we can translate the Gale-Stewart game into an equivalent graph game with winning condition of type: Muller, Rabin, parity, ...

EXAMPLES: P_0 CIRCLE, P_1 SQUARE



Reachability game:

$F = \{a\}$: $W_0 = \{a, b, c, f\}$

$F = \{e\}$: $W_0 = V \setminus \{a\}$

Büchi games: same W_0 as above

ATTRACTORS

$$\begin{aligned} \text{Attr}_0^0(F) &= F \\ \text{Attr}_0^{n+1}(F) &= \text{Attr}_0^n(F) \cup \\ &\quad \{v \in V_0 \mid \exists w \in \text{Attr}_0^n(F), (v, w) \in E\} \\ &\quad \{v \in V_1 \mid \forall w \text{ t.q. } (v, w) \in E : w \in \text{Attr}_0^n(F)\} \end{aligned}$$

$$\text{Attr}_0^0(F) \subseteq \text{Attr}_0^1(F) \subseteq \dots \subseteq \text{Attr}_0^{|V|}(F)$$

$\text{Attr}_0^i(F)$ is the set of vertices from which P_0 can reach F after at most i moves. $W_0 = \text{Attr}_0^{|V|}(F)$ is the **winning region** of P_0 (smallest fixpoint), and $V \setminus \text{Attr}_0^{|V|}(F)$ is the winning region of P_1 .

STRATEGIES

Reachability games have **positional** winning strategies (for both players), which can be computed in polynomial time.

Positional strategy (for P_0): mapping $\sigma : V_0 \rightarrow V$.

STRATEGIES

Büchi games have **positional** winning strategies (for both players), which can be computed in polynomial time.

ALGORITHM

- $\text{Attr}_0^+(F)$: set of states from which P_0 can reach F in **at least** one move. We can compute $\text{Attr}_0^+(F)$, as well as a positional strategy, in polynomial time.
- $X^{(i)}$: set of states from which P_0 can go through F **at least** i times (without counting the starting state).

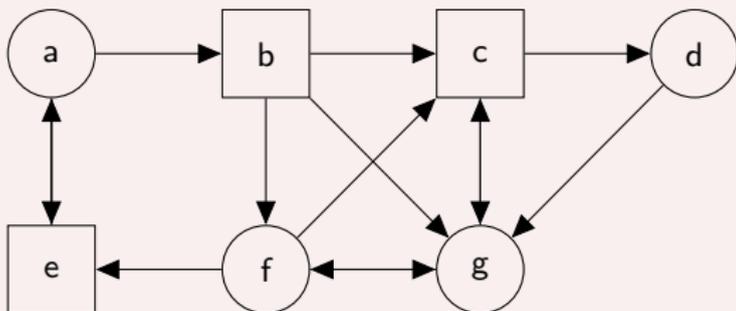
$$X^{(0)} = V, \quad X^{(i+1)} = \text{Attr}_0^+(X^{(i)} \cap F)$$

$$W_0 = \bigcap_{i \geq 1} X^{(i)}$$

Notice that $X^{(i+1)} \subseteq X^{(i)}$. So there exists k such that $W_0 = X^{(k)}$ and W_0 is a largest fixpoint.

- Strategy for P_0 in the Büchi game: positional strategy used by P_0 from $X^{(k)} = \text{Attr}_0^+(X^{(k)} \cap F)$.

NODE COLORING $\chi : V \rightarrow \{0, \dots, k\}$. WINNING CONDITION: MAXIMAL VISITED COLOR IS EVEN.



$\chi(a) = \chi(e) = 0$, $\chi(b) = \chi(f) = 1$, $\chi(c) = \chi(g) = 2$, $\chi(d) = 3$.

- $d \in W_1$ and $\text{Attr}_1(d) = \{d, c, b\} \subseteq W_1$
- $V' = V \setminus \{b, c, d\}$ is a trap for P_1 , defining a subarena for P_0 . So in V' , P_0 can avoid color 3, and $\text{Attr}_0(g) \upharpoonright_{V'} = \{g, f\} \subseteq W_0$
- $V'' = V' \setminus \{f, g\}$ is a trap for P_0 , so subarena for P_1 .
 $W_1 \upharpoonright_{V''} = \emptyset$, $W_0 \upharpoonright_{V''} = \{a, e\}$

Conclusion: $W_0 = \{a, e, f, g\}$, $W_1 = \{b, c, d\}$. Positional winning strategies for both players (attractor strategies).

- The complement of an attractor C for P_i is a **trap** pour P_i : P_i has no outgoing transition from C , so P_{1-i} can force the play to stay in $V \setminus C$.
- A trap C for P_i is a **subarena** for P_{1-i} : every winning strategy for P_{1-i} in C is winning for P_{1-i} in V .

SOLUTION

$$V^{(k)} := \chi^{-1}(k).$$

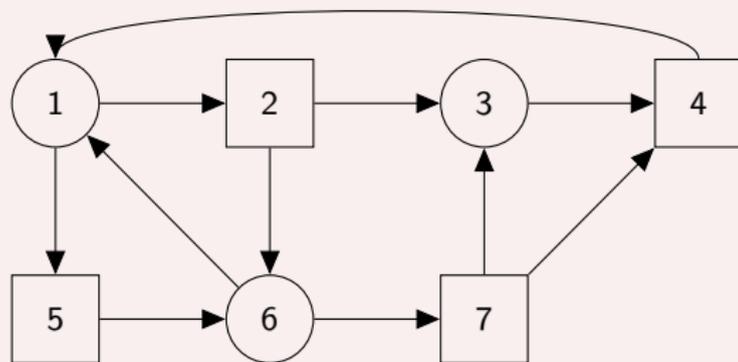
- 1 Suppose that the maximal color k is odd (otherwise exchange P_0, P_1). Compute $\text{Attr}_1(V^{(k)})$ (a subset of W_1).
- 2 $V' := V \setminus \text{Attr}_1(V^{(k)})$ is subarena for P_0 and the maximal color in V' is at most $k - 1$. Compute inductively the winning region W_0 of P_0 in V' . Every vertex in W_0 is winning for P_0 in V (see above).
- 3 Every vertex in $V' \setminus W_0$ is winning for P_1 in V : P_1 applies his winning strategy from V' while staying in V' , and the attractor strategy as soon as the game enters $\text{Attr}_1(V^{(k)})$.
- 4 Strategies for both players are positional.

WINNING CONDITION

Win is given as a set $\mathcal{F} \subseteq \mathcal{P}(V)$ of sets of states. Player P_0 wins a play π if the set $\text{Occ}(\pi)$ of states visited in π belongs to \mathcal{F} .

Obligation games are thus Boolean combinations of reachability games.

EXAMPLE: CONJUNCTION OF REACHABILITY CONDITIONS



Play $\pi \in \text{Win}$ iff $\{2, 7\} \subseteq \text{Occ}(\pi)$. Player P_0 needs 1 bit of memory in state 1: he needs to visit alternatively states 2 and 5.

SOLUTION

Reduce the obligation game to a simpler (and larger) weak parity game.

- Recall: P_0 wins in a weak parity game if the maximal visited color is even.
- General idea of the **reduction**: start with an obligation game $G = (\mathcal{A}, \mathcal{F})$ and define a weak parity game $G' = (\mathcal{A}', \chi)$ such that:
 - 1 $\mathcal{A}' = (V'_0, V'_1, E')$, $V'_i = V_i \times M$, with M finite (memory).
 - 2 Transitions: $(\langle v, m \rangle, \langle w, m' \rangle) \in E'$ if $(v, w) \in E$ and $m' = f(m, v, w)$ ($f =$ memory update function).
 - 3 With each play π in G starting in v , and each initial value $m_0 \in M$, we associate a unique play π' in G' from (v, m_0) . In addition, π and π' are won by the same player.
- Since weak parity games have positional strategies, such a reduction then provides finite memory strategies in the obligation game.

SOLUTION

Reduce the obligation game to a simpler (and larger) weak parity game.

- Recall: P_0 wins in a weak parity game if the maximal visited color is even.
- General idea of the **reduction**: start with an obligation game $G = (\mathcal{A}, \mathcal{F})$ and define a weak parity game $G' = (\mathcal{A}', \chi)$ such that:
 - $\mathcal{A}' = (V'_0, V'_1, E')$, $V'_i = V_i \times M$, with M finite (memory).
 - Transitions: $(\langle v, m \rangle, \langle w, m' \rangle) \in E'$ if $(v, w) \in E$ and $m' = f(m, v, w)$ ($f =$ memory update function).
 - With each play π in G starting in v , and each initial value $m_0 \in M$, we associate a unique play π' in G' from (v, m_0) . In addition, π and π' are won by the same player.
- Since weak parity games have positional strategies, such a reduction then provides finite memory strategies in the obligation game.

REDUCTION

- $M = 2^V$, $E' = \{(v, R) \rightarrow (w, R \cup \{v\}) \mid (v, w) \in E\}$.
- Coloring: $\chi(v, R) = 2|R|$ if $R \in \mathcal{F}$; $\chi(v, R) = 2|R| - 1$ if $R \notin \mathcal{F}$.
- For every play $\pi' = (v_0, R_0), (v_1, R_1), \dots$ in G' we have $R_0 \subseteq R_1 \subseteq R_2 \dots$, so for some $i \geq 0$ we have $R_j = R_i$ for all $j \geq i$. For $R_0 = \emptyset$, the set R_i equals $\text{Occ}(\pi)$, where $\pi = v_0, v_1, \dots$

Let $\text{Inf}(v_0v_1\cdots) = \{v \in V \mid \exists^\infty i : v = v_i\}$. For simplicity, we assume that there are no dead-ends in (V, E) .

PARITY GAMES

Node coloring $\chi : V \rightarrow \{1, \dots, k\}$.

$$\text{Win} = \{\pi \mid \max(\chi(\text{Inf}(\pi))) \text{ is even}\}.$$

MULLER GAMES

Let $\mathcal{F} \subseteq \mathcal{P}(V)$.

$$\text{Win} = \{\pi \mid \text{Inf}(\pi) \in \mathcal{F}\}$$

REM.

Every game $(\mathcal{A}, \text{Win})$ with $\text{Win} \subseteq V^\omega$ regular reduces to a Muller game. We will show how to reduce Muller games to parity games, then how to solve parity games.

(DZIEMBOWSKI, JURDZINSKI, WALUKIEWICZ)

- $V_0 = \{A, B, C, D\}$, $V_1 = \{1, 2, 3, 4\}$
- $E = V_0 \times V_1 \cup V_1 \times V_0$
- Winning condition for *player 1*:

$$|\text{Inf}(\pi) \cap V_0| = \max(|\text{Inf}(\pi) \cap V_1|)$$

How can P_1 win? solution: Last Appearance Record (LAR).

(DZIEMBOWSKI, JURDZINSKI, WALUKIEWICZ)

- $V_0 = \{A, B, C, D\}$, $V_1 = \{1, 2, 3, 4\}$
- $E = V_0 \times V_1 \cup V_1 \times V_0$
- Winning condition for *player 1*:

$$|\text{Inf}(\pi) \cap V_0| = \max(|\text{Inf}(\pi) \cap V_1|)$$

How can P_1 win? solution: Last Appearance Record (LAR).

LAR

The order of visited V_0 states is recorded: the last visited state is moved to the front of LAR, and the position where it occurred previously (if any) is underlined – the hit position.

Example:

$$\epsilon \xrightarrow{A} A \xrightarrow{D} DA \xrightarrow{C} CDA \xrightarrow{C} \underline{C}DA \xrightarrow{D} D\underline{C}A \xrightarrow{D} \underline{D}CA \xrightarrow{C} C\underline{D}A \dots$$

If $\text{Inf}(\pi) \cap V_0 = \{C, D\}$ then infinitely often, the hit position in the LAR is 2 (we start to count from 1). P_1 's (winning) strategy is to play the hit position of the LAR.

REDUCTION

- Let (G, \mathcal{F}) be a Muller game with $G = (V_0, V_1, E)$ and $\mathcal{F} \subseteq 2^V$.
- LAR: pairs $\langle v_1 \cdots v_r, h \rangle$, where $0 \leq h \leq r$ and v_1, \dots, v_r are distinct states from V .

Let Q be the set of LAR's. The initial LAR is $q_0 = \langle \epsilon, 0 \rangle$, and the transition function $\delta : Q \times V \rightarrow Q$ is given by:

$$\delta(\langle v_1 \cdots v_r, h \rangle, v) = \begin{cases} \langle v v_1 \cdots v_r, 0 \rangle & \text{if } v \neq v_p \text{ for all } p \\ \langle v v_1 \cdots v_{p-1} v_{p+1} \cdots v_r, p \rangle & \text{if } v = v_p \end{cases}$$

Hit set of $\langle v_1 \cdots v_r, h \rangle$: $\{v_1, \dots, v_h\}$.

- For an infinite play π , let $\pi = \pi_0 \pi'$ be such that $\text{Inf}(\pi)$ is the set of states occurring in π' . If h is the maximal hit position repeated infinitely often, then $\text{Inf}(\pi)$ equals the hit set for the LAR's with hit position h in π' .

$$\chi(\langle v_1 \cdots v_r, h \rangle) = \begin{cases} 2h & \text{if } \{v_1, \dots, v_h\} \in \mathcal{F} \\ 2h - 1 & \text{else} \end{cases}$$

$\text{Inf}(\pi) \in \mathcal{F}$ iff the maximal LAR-color repeated infinitely often is even.
Thus, Muller games can be simulated by parity games (with $n! \cdot n$ additional memory).

INDUCTIVE SOLUTION

- Let $G = (V_0, V_1, E)$ and $\chi : V \rightarrow \{0, \dots, k\}$ be a parity game.
- Induction over $|V|$. Suppose w.l.o.g. that the maximal color k is even and let v_0 be some state with $\chi(v) = k$.
- Let $A_0 = \text{Attr}_0(v_0)$, so $V \setminus A_0$ is a subgame for P_1 . By induction, $V \setminus A_0$ is partitioned into W'_0 and W'_1 , and P_i has a positional winning strategy on W'_i .
- 2 cases (and exactly one applies):
 - ① From v_0 , P_0 can get into $A_0 \cup W'_0$ in one step.
 - ② From v_0 , P_1 can get into W'_1 in one step.
- Case 1: $W_0 = W'_0 \cup A_0$, $W_1 = W'_1$.
- Case 2: $v_0 \in \text{Attr}_1(W'_1)$. Let $A_1 = \text{Attr}_1(W'_1 \cup \{v_0\})$.
 $V \setminus A_1$ is a subgame for P_1 . Let W''_i be the winning region of P_i in this subgame (with positional strategies).
 $W_0 = W''_0$ and $W_1 = W''_1 \cup A_1$.

INDUCTIVE SOLUTION

- Let $G = (V_0, V_1, E)$ and $\chi : V \rightarrow \{0, \dots, k\}$ be a parity game.
- Induction over $|V|$. Suppose w.l.o.g. that the maximal color k is even and let v_0 be some state with $\chi(v) = k$.
- Let $A_0 = \text{Attr}_0(v_0)$, so $V \setminus A_0$ is a subgame for P_1 . By induction, $V \setminus A_0$ is partitioned into W'_0 and W'_1 , and P_i has a positional winning strategy on W'_i .
- 2 cases (and exactly one applies):
 - ① From v_0 , P_0 can get into $A_0 \cup W'_0$ in one step.
 - ② From v_0 , P_1 can get into W'_1 in one step.
- Case 1: $W_0 = W'_0 \cup A_0$, $W_1 = W'_1$.
- Case 2: $v_0 \in \text{Attr}_1(W'_1)$. Let $A_1 = \text{Attr}_1(W'_1 \cup \{v_0\})$.
 $V \setminus A_1$ is a subgame for P_1 . Let W''_i be the winning region of P_i in this subgame (with positional strategies).
 $W_0 = W''_0$ and $W_1 = W''_1 \cup A_1$.

COMPLEXITY

The above algorithm has exponential running time. Since strategies are positional, computing the winning regions is in $\text{NP} \cap \text{co-NP}$.

Applying parity games: automata over infinite trees

Ch. 8 in *Automata, Logics and Infinite Games* (Eds. Grädel,
W. Thomas, Th. Wilke)

STRUCTURES

- The infinite binary tree is the set $\{0, 1\}^*$ of all words over $\{0, 1\}$: root ϵ , children of node u : $u0, u1$.
- A Σ -labeled infinite binary tree is a mapping $T : \{0, 1\}^* \rightarrow \Sigma$.

AUTOMATA

On infinite trees: top-down automata $\mathcal{A} = \langle S, \Sigma, \delta, s_0, \text{Acc} \rangle$, where $\delta \subseteq S \times \Sigma \times S \times S$.

Run $\rho : \{0, 1\}^* \rightarrow S$ on tree T :

- $\rho(\epsilon) = s_0$,
- $(\rho(u), T(u), \rho(u0), \rho(u1)) \in \delta$ for all $u \in \{0, 1\}^*$.

A run ρ is accepted if *all* infinite paths of ρ satisfy Acc (Acc can be Büchi, Muller, parity etc.).

An infinite tree T is accepted by \mathcal{A} if there is at least one accepting run ρ on T . We denote by $\mathcal{L}(\mathcal{A})$ the set of accepted trees.

BÜCHI TREE AUTOMATA

Büchi tree automata are strictly weaker than parity tree automata.

Example: “each path of the tree has finitely many a -nodes” cannot be accepted by a Büchi automaton, even non-deterministic (why?).

Parity tree automaton:

- $S = \{s_a, s_b\}$, $\Sigma = \{a, b\}$,
- $\delta = \{s_a\} \times \{a\} \times S^2 \cup \{s_b\} \times \{b\} \times S^2$.
- $\chi(s_b) = 0$, $\chi(s_a) = 1$.

EXPRESSIVE POWER

Parity, Muller, Rabin, Streett automata all recognize the same languages. The deterministic versions are strictly weaker (why?).

CLOSURE OPERATIONS

Parity tree automata are closed under union, intersection and projection.

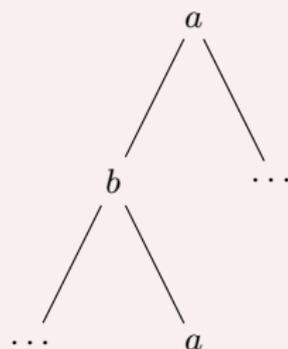
COMPLEMENTATION (RABIN 1972)

Parity tree automata can be effectively complemented.

COMPLEMENTATION VIA GAMES (GUREVICH/HARRINGTON 1982)

Two players, **Automaton** and **Pathfinder**. Player **Automaton** chooses a transition, player **Pathfinder** chooses a direction (left/right) to proceed. The “outcome” is an infinite sequence from $(\Sigma \times S)^\omega$. **Automaton** wins the game if each outcome satisfies the parity condition.

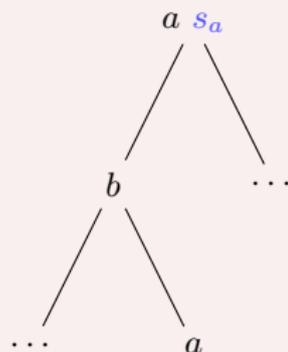
EXAMPLE



COMPLEMENTATION VIA GAMES (GUREVICH/HARRINGTON 1982)

Two players, **Automaton** and **Pathfinder**. Player **Automaton** chooses a transition, player **Pathfinder** chooses a direction (left/right) to proceed. The “outcome” is an infinite sequence from $(\Sigma \times S)^\omega$. **Automaton** wins the game if each outcome satisfies the parity condition.

EXAMPLE



COMPLEMENTATION GAME (1/3)

- The above game (on tree T) is a *parity game* \mathcal{G}_T on an infinite graph:
 - Automaton positions: $V_0 = \{(u, s) \mid u \in \{0, 1\}^*, s \in S\}$,
 - Pathfinder positions: $V_1 = \{(u, \tau) \mid u \in \{0, 1\}^*, \tau \in S \times T(u) \times S^2 \cap \delta\}$.

From position (u, s) , **Automaton** can choose a transition $\tau = (s, T(u), s_0, s_1)$ and move to (u, τ) . From position (u, τ) , **Pathfinder** can choose $i \in \{0, 1\}$ and move to (ui, s_i) . Color (u, s) and $(u, (s, a, s_0, s_1))$ by $\chi(s)$.

COMPLEMENTATION GAME (1/3)

- The above game (on tree T) is a *parity game* \mathcal{G}_T on an infinite graph:
 - Automaton positions: $V_0 = \{(u, s) \mid u \in \{0, 1\}^*, s \in S\}$,
 - Pathfinder positions: $V_1 = \{(u, \tau) \mid u \in \{0, 1\}^*, \tau \in S \times T(u) \times S^2 \cap \delta\}$.

From position (u, s) , **Automaton** can choose a transition $\tau = (s, T(u), s_0, s_1)$ and move to (u, τ) . From position (u, τ) , **Pathfinder** can choose $i \in \{0, 1\}$ and move to (ui, s_i) . Color (u, s) and $(u, (s, a, s_0, s_1))$ by $\chi(s)$.

- **Automaton** has a winning strategy in \mathcal{G}_T (from (ϵ, s_{in})) iff $T \in \mathcal{L}(\mathcal{A})$.

COMPLEMENTATION GAME (1/3)

- The above game (on tree T) is a *parity game* \mathcal{G}_T on an infinite graph:
 - Automaton positions: $V_0 = \{(u, s) \mid u \in \{0, 1\}^*, s \in S\}$,
 - Pathfinder positions: $V_1 = \{(u, \tau) \mid u \in \{0, 1\}^*, \tau \in S \times T(u) \times S^2 \cap \delta\}$.

From position (u, s) , **Automaton** can choose a transition $\tau = (s, T(u), s_0, s_1)$ and move to (u, τ) . From position (u, τ) , **Pathfinder** can choose $i \in \{0, 1\}$ and move to (ui, s_i) . Color (u, s) and $(u, (s, a, s_0, s_1))$ by $\chi(s)$.

- **Automaton** has a winning strategy in \mathcal{G}_T (from (ϵ, s_{in})) iff $T \in \mathcal{L}(\mathcal{A})$.
- The game \mathcal{G}_T is determined. Hence, **Pathfinder** has a winning strategy in \mathcal{G}_T (from (ϵ, s_{in})) iff $T \notin \mathcal{L}(\mathcal{A})$.

COMPLEMENTATION GAME (1/3)

- The above game (on tree T) is a *parity game* \mathcal{G}_T on an infinite graph:
 - Automaton positions: $V_0 = \{(u, s) \mid u \in \{0, 1\}^*, s \in S\}$,
 - Pathfinder positions: $V_1 = \{(u, \tau) \mid u \in \{0, 1\}^*, \tau \in S \times T(u) \times S^2 \cap \delta\}$.

From position (u, s) , **Automaton** can choose a transition $\tau = (s, T(u), s_0, s_1)$ and move to (u, τ) . From position (u, τ) , **Pathfinder** can choose $i \in \{0, 1\}$ and move to (ui, s_i) . Color (u, s) and $(u, (s, a, s_0, s_1))$ by $\chi(s)$.

- **Automaton** has a winning strategy in \mathcal{G}_T (from (ϵ, s_{in})) iff $T \in \mathcal{L}(\mathcal{A})$.
- The game \mathcal{G}_T is determined. Hence, **Pathfinder** has a winning strategy in \mathcal{G}_T (from (ϵ, s_{in})) iff $T \notin \mathcal{L}(\mathcal{A})$.
- Turn **Pathfinder's** winning, memoryless strategy into a parity tree automaton (accepting the complementary tree language):
Memoryless strategy of **Pathfinder** = mapping $\sigma : \{0, 1\}^* \times \delta \rightarrow \{0, 1\}$.
Equivalently, $\sigma : \{0, 1\}^* \rightarrow (\delta \rightarrow \{0, 1\})$.

COMPLEMENTATION GAME (2/3)

Strategy of **Pathfinder**: $\sigma : \{0, 1\}^* \rightarrow (\delta \rightarrow \{0, 1\})$.

- Trees labeled by $\delta \rightarrow \{0, 1\}$ are called *strategy trees*. A strategy tree is *winning* for T if it corresponds to a winning strategy of **Pathfinder** in \mathcal{G}_T .

COMPLEMENTATION GAME (2/3)

Strategy of **Pathfinder**: $\sigma : \{0, 1\}^* \rightarrow (\delta \rightarrow \{0, 1\})$.

- Trees labeled by $\delta \rightarrow \{0, 1\}$ are called *strategy trees*. A strategy tree is *winning* for T if it corresponds to a winning strategy of **Pathfinder** in \mathcal{G}_T .
- There exists a winning tree \mathcal{S} iff $T \notin \mathcal{L}(\mathcal{A})$.

COMPLEMENTATION GAME (2/3)

Strategy of **Pathfinder**: $\sigma : \{0, 1\}^* \rightarrow (\delta \rightarrow \{0, 1\})$.

- Trees labeled by $\delta \rightarrow \{0, 1\}$ are called *strategy trees*. A strategy tree is *winning* for T if it corresponds to a winning strategy of **Pathfinder** in \mathcal{G}_T .
- There exists a winning tree \mathcal{S} iff $T \notin \mathcal{L}(\mathcal{A})$.
- Given the tree T and a strategy tree \mathcal{S} we consider the “product” tree $T \times \mathcal{S}$. Then \mathcal{S} is winning iff **every path** of $T \times \mathcal{S}$ **violates** the parity condition of \mathcal{A} .

COMPLEMENTATION GAME (2/3)

Strategy of **Pathfinder**: $\sigma : \{0, 1\}^* \rightarrow (\delta \rightarrow \{0, 1\})$.

- Trees labeled by $\delta \rightarrow \{0, 1\}$ are called *strategy trees*. A strategy tree is *winning* for T if it corresponds to a winning strategy of **Pathfinder** in \mathcal{G}_T .
- There exists a winning tree \mathcal{S} iff $T \notin \mathcal{L}(\mathcal{A})$.
- Given the tree T and a strategy tree \mathcal{S} we consider the “product” tree $T \times \mathcal{S}$. Then \mathcal{S} is winning iff **every path** of $T \times \mathcal{S}$ **violates** the parity condition of \mathcal{A} .
- Paths in $T \times \mathcal{S}$:

$$(a_0, f_0, i_0), (a_1, f_1, i_1), \dots$$

$$a_j \in \Sigma, f_j : \delta \rightarrow \{0, 1\}, i_j \in \{0, 1\}.$$

- A *non-deterministic* parity ω -automaton \mathcal{C} with state set S and coloring χ can check that a path in $T \times \mathcal{S}$ **satisfies** the parity condition (non-determinism: guess the transitions of \mathcal{A}).

COMPLEMENTATION GAME (3/3)

Let $\mathcal{D} = \langle S', \Sigma', \delta', s'_{\text{in}}, \chi' \rangle$ be a **deterministic** parity automaton accepting $(\Sigma')^\omega \setminus \mathcal{L}(\mathcal{C})$, with $\Sigma' = \Sigma \times \{0, 1\}^\delta \times \{0, 1\}$.

- Define a tree automaton $\mathcal{B} = \langle S', \Sigma, \Delta', s'_{\text{in}}, \chi' \rangle$ that runs \mathcal{D} on every path of T , guessing Σ' at the same time:

$$(s', a, s'_0, s'_1) \in \Delta' \quad \text{iff} \quad s_0 = \delta'(s', (a, f, 0)) \text{ and } s_1 = \delta'(s', (a, f, 1)) \\ \text{for some } f : \delta \rightarrow \{0, 1\}$$

COMPLEMENTATION GAME (3/3)

Let $\mathcal{D} = \langle S', \Sigma', \delta', s'_{\text{in}}, \chi' \rangle$ be a **deterministic** parity automaton accepting $(\Sigma')^\omega \setminus \mathcal{L}(\mathcal{C})$, with $\Sigma' = \Sigma \times \{0, 1\}^\delta \times \{0, 1\}$.

- Define a tree automaton $\mathcal{B} = \langle S', \Sigma, \Delta', s'_{\text{in}}, \chi' \rangle$ that runs \mathcal{D} on every path of T , guessing Σ' at the same time:

$$(s', a, s'_0, s'_1) \in \Delta' \quad \text{iff} \quad s_0 = \delta'(s', (a, f, 0)) \text{ and } s_1 = \delta'(s', (a, f, 1)) \\ \text{for some } f : \delta \rightarrow \{0, 1\}$$

- \mathcal{B} accepts the complement of \mathcal{A} .

Games & complementing Büchi ω -automata

Ch. 4 in *Automata, Logics and Infinite Games* (Eds. Grädel,
W. Thomas, Th. Wilke)

METHODS

- Büchi (1962): combinatorial arguments
- McNaughton (1966), Safra (1988): determinization
- Klarlund (1991): progress measures
- Kupferman/Vardi (1997): alternating automata

COMPLEXITY (UPPER/LOWER BOUNDS)

- Büchi: doubly exponential upper bound
- Safra: upper bound $2^{\Theta(n \log(n))}$
- M. Michel (1988): lower bound $n!$
- Kupferman/Vardi (1997, . . .): upper bound $(c_1 \cdot n)^n$ ($c < 1$)
- Yan (2008): lower bound $(c_2 \cdot n)^n$
- Schewe (2009): matching upper bound

RUN GRAPHS

- Büchi automaton $\mathcal{A} = \langle S, \Sigma, s_{in}, \delta, F \rangle$, ω -word w
- Run graph $G(w)$ of \mathcal{A} on $w = a_0 a_1 \dots$:

$$G(w) = \langle V, E, C \rangle$$

- vertices $V \subseteq S \times \mathbb{N}$, where

$$\begin{aligned} (s, 0) &\in V \text{ iff } s = s_{in} \\ (s', n + 1) &\in V \text{ iff } s' \in \delta(s, a_n) \text{ for some } (s, n) \in V \end{aligned}$$

- edge set:

$$(s, n) \rightarrow (s', n + 1) \text{ if } s' \in \delta(s, a_n)$$

- marked vertices:

$$C = \{(s, n) \in V \mid s \in F\}$$

- $w \notin L(\mathcal{A})$ iff no infinite path in $G(w)$ has infinitely many marked vertices (we say that $G(w)$ is **finitely marked**).

IDEA

- Partition the vertex set V of $G(w)$ in a (infinite) sequence of sets V_0, V_1, \dots :
 - $V_0 \subseteq V$ contains all vertices from which we can reach only finitely many vertices,
 - $V_1 \subseteq V \setminus V_0$ contains all vertices from which we cannot (non-trivially) reach any marked vertex,
 - $V_2 \subseteq V \setminus (V_0 \cup V_1)$ contains all vertices from which we can reach only finitely many vertices,
 - ...
- We show that $w \notin L(\mathcal{A})$ iff there exists some $0 \leq k \leq 2|S|$ such that $V_k = \emptyset$ for all $k > k_0$.
- Büchi automaton for $\Sigma^\omega \setminus L(\mathcal{A})$ guesses the existence of a partition.

PROGRESS MEASURES

Let $m \in \mathbb{N}$ and $G = \langle V, E, C \rangle$ be a DAG with $C \subseteq V$.

A **progress measure** of size m for G is a mapping $\mu : V \rightarrow \{1, \dots, 2m + 1\}$ satisfying:

- $\mu(u) \geq \mu(v)$ for all $(u, v) \in E$,
- if $\mu(u) = \mu(v)$ and $(u, v) \in E$, then either $\mu(u)$ is odd or $v \notin C$,
- there is no infinite path in G with constant μ -value odd.

NOTATIONS

$G = \langle V, E, C \rangle$ marked graph. For $v \in V$ let $R(v) = \{w \in V \mid v \xrightarrow{*} w\}$ denote the set of vertices reachable from v .

- **Finite reach:**

$$B(G) = \{v \in V \mid R(v) \text{ is finite}\}$$

- **Unmarked reach:**

$$U(G) = \{v \in V \mid C \cap (R(v) \setminus \{v\}) = \emptyset\}$$

- If G is layered DAG (edges from layer n go to layer $n + 1$):

width(G) = maximal number occurring infinitely often as size of a layer.

PARTITION OF $G(w)$

Define sequence of subgraphs $(G_n)_n$ and sets of vertices $(V_n)_n$:

- $G_0 = G(w)$ and $V_0 = B(G_0)$,
- $G_{2n+1} = G_{2n} \setminus V_{2n}$ and $V_{2n+1} = U(G_{2n+1})$,
- $G_{2n+2} = G_{2n+1} \setminus V_{2n+1}$ and $V_{2n+2} = B(G_{2n+1})$.

We have: $V \supseteq V_0 \supseteq V_1 \supseteq \dots$

PROGRESS MEASURE FOR $G(w)$

- If $G(w)$ is finitely marked and $G_{2n+1} \neq \emptyset$:

$$\text{width}(G_{2n+2}) < \text{width}(G_{2n+1}).$$

By construction $U(G_{2n+1}) \neq \emptyset$. Moreover, if $v \in U(G_{2n+1})$ then using $G_{2n+1} = G_{2n} \setminus V_{2n}$ we obtain that there is an infinite path starting in v and lying in G_{2n+1} . Since this path is not preserved in G_{2n+2} , we obtain the claim.

- Let $n = |S|$ ($= \text{width}(G_0)$). Then $G_{2n+1} = \emptyset$.
- $G(w) = \langle V, E, C \rangle$. Recall: $w \notin L(\mathcal{A})$ iff $G(w)$ finitely marked. Define

$$\mu(v) = k + 1 \text{ with } k = \min\{n \mid v \in G_n \setminus G_{n+1}\}$$

Then μ is a progress measure for $G(w)$ of size $m = |S|$.

COMPLEMENTATION

A Büchi automaton of size $2^{O(n+m \log(n))}$ can check the existence of a progress measure of size m .

$\mathcal{B}^+(X)$: SET OF POSITIVE FORMULAS OVER X

- contains X and constants 0,1,
- built using \vee, \wedge .

Model M for $\phi \in \mathcal{B}^+(X)$: subset of X s.t.. ϕ evaluates to 1 if $x \in M$ is replaced by 1, and $x \notin M$ by 0. Set of minimal models of ϕ : $\text{Mod}(\phi)$.

WEAK ALTERNATING PARITY AUTOMATA (WAPA)

$\mathcal{A} = \langle S, \Sigma, \delta, s_{in}, \chi \rangle$ with $\chi : S \rightarrow \{1, \dots, d\}$ and $\delta : S \times \Sigma \rightarrow \mathcal{B}^+(S)$.

RUNS OF WAPA

Runs of WAPA \mathcal{A} on $w = a_0 a_1 \dots$ are DAGs G with vertex set $V \subseteq S \times \mathbb{N}$, such that:

- $(s_{in}, 0) \in V$,
- $(s, m) \rightarrow (t, n)$ implies $n = m + 1$,
- for every $(s, n) \in V$: $\{s' \in S \mid (s, n) \rightarrow (s', n + 1)\} \in \text{Mod}(\delta(s, a_n))$.

An infinite path of G is accepting if the minimal color **seen** is even. A finite path is accepting if the last node (s, n) is such that $\delta(s, a_n) = 1$. A run DAG is accepting if all paths are accepting. An ω -word w is accepted if there is some accepting run DAG on w .

WEAK PARITY GAME

Given a WAPA \mathcal{A} and $w = a_0 a_1 \dots$ we define a weak parity game $\mathcal{G}(\mathcal{A}, w) = \langle V_0, V_1, E, \chi \rangle$:

- $V_0 = S \times \mathbb{N}$, $V_1 = S \times \mathcal{P}(S) \times \mathbb{N}$
- $(s, m) \rightarrow (t, M, n)$ if $n = m + 1$ and $M \in \text{Mod}(\delta(s, a_m))$
- $(s, M, m) \rightarrow (t, n)$ if $m = n$ and $t \in M$
- $\chi(s, n) = \chi(s, M, n)$ is the color of s in \mathcal{A} .

WAPA & WEAK PARITY GAME

$w \in L(\mathcal{A})$ iff P_0 wins the weak parity game $\mathcal{G}(\mathcal{A}, w)$.

DUAL WAPA

- Dual formula $\widehat{\phi}$ obtained from $\phi \in \mathcal{B}^+(X)$ by exchanging 0, 1 and \vee, \wedge .
- Dual WAPA $\widehat{\mathcal{A}} = \langle S, \Sigma, \widehat{\delta}, s_{in}, \widehat{\chi} \rangle$, $\widehat{\delta}(s, a) = \widehat{\delta(s, a)}$, $\widehat{\chi}(s) = \chi(s) + 1$.

By determinacy of weak parity games: P_0 wins $\mathcal{G}(\mathcal{A}, w)$ iff P_1 wins $\mathcal{G}(\widehat{\mathcal{A}}, w)$.

BÜCHI \rightarrow WAPA

From a Büchi automaton \mathcal{B} with n states construct a WAPA \mathcal{A} with $O(n^2)$ states:

- $\mathcal{B} = \langle S, \Sigma, \delta, s_{in}, F \rangle$, $\mathcal{A} = \{S', \Sigma, \delta', s'_{in}, \chi\}$
- $S' = S \times \{0, \dots, 2|S|\}$, $\chi(s, i) = i$,
- $s'_{in} = (s_{in}, 2|S|)$,

$$\delta'((s, i), a) = \begin{cases} \bigvee_{s' \in \delta(s, a)} (s', 0) & \text{if } i = 0 \\ \bigvee_{s' \in \delta(s, a)} ((s', i) \wedge (s', i - 1)) & \text{if } i > 0 \text{ and } i \text{ even} \\ \bigvee_{s' \in \delta(s, a)} (s', i) & \text{if } s \notin F \text{ and } i \text{ odd} \\ \bigvee_{s' \in \delta(s, a)} (s', i - 1) & \text{if } s \in F \text{ and } i \text{ odd} \end{cases}$$

Show that $L(\mathcal{B}) = L(\mathcal{A})$.

WAPA \rightarrow BÜCHI

From a WAPA with n states one can construct an equivalent Büchi automaton with $2^{O(n^2)}$ states.