About Games and Trees

Barbara Jobstmann

Verimag/CNRS (Grenoble, France)

November 27, 2009

Recap

Winning conditions are defined over Occ and Inf.

$\mathrm{Occ}(\rho)$	$\mathrm{Inf}(\rho)$
Reachability/Guarantee game	Büchi game
Safety game	co-Büchi game
Weak-parity game	Parity game
Obligation/Staiger-Wagner game	Muller game
LTL games	

Recap

How did we solve those games?

Game	Solution
Reachability games	Attractor + Attractor Strategy
Safety games	like Reachability games
Büchi games	Recurrence set + Extended Attractor Strategy
co-Büchi games	like Büchi games
Weak-parity games	Alternation between $Attr_0$ and $Attr_1$
Obligation games	Reduction to Weak-parity games with record sets
Parity games	Recursive algorithm, Progress-Measure algorithm, and Strategy Improvement algorithm

Muller, Rabin, and Streett Games

Muller Games

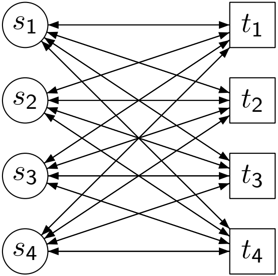
Given a game graph $G = (S, S_0, E)$ and a Muller condition $\mathcal{F} \subseteq \mathcal{P}(S)$, then a play ρ is winning for Player 0 if exists $F \in \mathcal{F}$ s.t.

$$\mathrm{Inf}(\rho) = F.$$

Recall, in Staiger-Wagner games, we had $\mathrm{Occ}(\rho) = F$.

Example

Player 0 wins iff the number of states in $S_0 = \{s_1, s_2, s_3, s_4\}$ visited infinitely often is equal to the lowest index of the states in $S_1 = \{t_1, t_2, t_3, t_4\}$ visited infinitely often.



Winning condition in Muller form: $F \in \mathcal{F}$ iff $\mathsf{min}_i(t_i \in F) = |F \cap S_0|$.

Record the Past

For simplicity, we record only the s-states.

Visited letter	Record set
s_1	s_1
s_3	$s_1 s_3$
s_3	$s_1 s_3$
s_4	$s_1 s_3 s_4$
s_2	$s_1 s_2 s_3 s_4$
s_4	$s_1 s_2 s_3 s_4$
s_3	-”-
s_4	-”-
s_4	-”-

Latest Appearance Record

Visited letter	Record set	LAR
s_1	s_1	$s_1 s_2 s_3 s_4 (1)$
s_3	$s_1 s_3$	$s_3 s_1 s_2 s_4 (3)$
s_3	$s_1 s_3$	$s_3 s_1 s_2 s_4 (1)$
s_4	$s_1 s_3 s_4$	$s_4 s_3 s_1 s_2 (4)$
s_2	$s_1 s_2 s_3 s_4$	$s_2 s_4 s_3 s_1 (4)$
s_4	$s_1 s_2 s_3 s_4$	$s_4 s_2 s_3 s_4 (2)$
s_3	-"-	..
s_4	-"-	..
s_4	-"-	..

Example

Assume the states s_3 and s_4 are repeated infinitely often. Then:

- the states s_1 and s_2 eventually arrive at the last two positions and are not touched any more, so finally the hit appears at most on positions 1 and 2
- position 2 is hit again and again; if only position 1 is hit from some point onwards, only the same letter would be chosen from there onwards (and not two states s_3 and s_4 as assumed)

Example

LAR-strategy for Player 0:

During play update and use the LAR as follows:

- shift the letter of the current state to the front
- record the position from where the current letter was taken
- move to the state whose index is the current hit position

This is a finite-state winning strategy with $n! \cdot n$ memory states if n letter states and n number states occur in the game graph.

From Muller to Parity Games

Theorem

For a game (G, ϕ) with $G = (S, S_0, E)$ and Muller winning condition ϕ (using the set $\mathcal{F} \subseteq 2^S$), there is a game (G', ϕ') with $G' = (S', S'_0, E')$ and parity winning condition ϕ' such that $(G, \phi) \leq (G', \phi')$

Proof.

Assume $= \{1, \ldots n\}$. Define $S' := LAR(S)$

LAR(S) is the set of pairs $((i_1, \ldots i_n), h)$ consisting of a permutation of $1, \ldots n$ and a number $h \in \{1, \ldots n\}$.

Construction

Initialisation: For $i \in S$ set

$$g(i) = ((i, i+1, \ldots, n, 1, \ldots, i-1), 1)$$

Definition of E': Introduce an edges from $((i_1 \ldots i_n), h)$ to $((i_m i_1 \ldots i_{m-1} i_{m+1} \ldots i_n), m)$ if $(i_1, i_m) \in E$

Construction

Initialisation: For $i \in S$ set

$$g(i) = ((i, i+1, \ldots, n, 1, \ldots, i-1), 1)$$

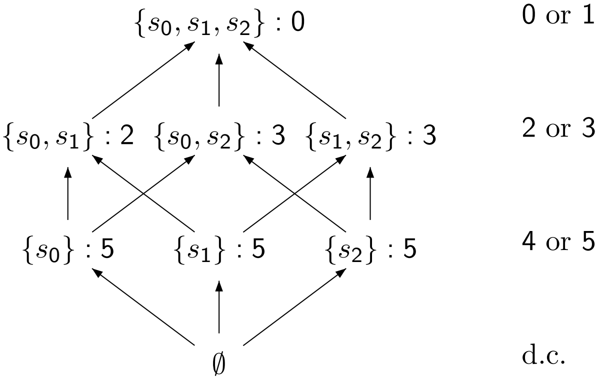
Definition of E': Introduce an edges from $((i_1 \ldots i_n), h)$ to $((i_m i_1 \ldots i_{m-1} i_{m+1} \ldots i_n), m)$ if $(i_1, i_m) \in E$

How should we assign the priorities?

Record Sets and Priorities

Recall, priorities in the reduction of Staiger-Wagner to Weak-Parity.

$F = \{\{s_0, s1\}, \{s0, s1, s2\}\}$.



Construction(2)

Now, we are only interested in states visited infinitely often. The hit value tells as how many states are visited infinitely often.

E.g., if s_0 and s_1 are visited infinitely often, we see from some point on only the LARs: $(s_0 s_1 \ldots, 1)$,$(s_0 s_1 \ldots, 2)$, $(s_1 s_0 \ldots, 1)$, $(s_1 s_0 \ldots, 2)$.

If $\mathcal{F} = \{\{s_0, s_1\}\}$, then we want plays that visit only $(s_0 s_1 \ldots, 1)$ or $(s_1 s_0 \ldots, 1)$ from some point on to be losing. So, the priorities signed to $(s_0 s_1 \ldots, 2)$ or $(s_0 s_1 \ldots, 2)$ need to override the priorities of $(s_0 s_1 \ldots, 1)$ or $(s_1 s_0 \ldots, 1)$.

Construction(2)

Now, we are only interested in states visited infinitely often. The hit value tells as how many states are visited infinitely often.

E.g., if s_0 and s_1 are visited infinitely often, we see from some point on only the LARs: $(s_0 s_1 \ldots, 1)$,$(s_0 s_1 \ldots, 2)$, $(s_1 s_0 \ldots, 1)$, $(s_1 s_0 \ldots, 2)$.

If $\mathcal{F} = \{\{s_0, s_1\}\}$, then we want plays that visit only $(s_0 s_1 \ldots, 1)$ or $(s_1 s_0 \ldots, 1)$ from some point on to be losing. So, the priorities signed to $(s_0 s_1 \ldots, 2)$ or $(s_0 s_1 \ldots, 2)$ need to override the priorities of $(s_0 s_1 \ldots, 1)$ or $(s_1 s_0 \ldots, 1)$.

Priorities p: LAR(S) $\rightarrow \{1, \ldots 2n\}$

$$p((i_1 \ldots i_n, h)) = 2n - \begin{cases} 2h - 1 & \text{if } \{i_1 \ldots i_h\} \notin \mathcal{F} \\ 2h & \text{if } \{i_1 \ldots i_h\} \in \mathcal{F} \end{cases}$$

Proof of Correctness

Lemma

Given a play ρ *in* (G, ϕ) *and its counterpart* ρ' *in* (G', ϕ')*, then* $\mathit{Inf}(\rho) = F$ *with* $|F| = m$ *iff*

1. *in* ρ' *the hit value is* $> m$ *only finitely often*
2. *in* ρ' *the hit-segment is equal to* F *infinitely often*

Proof (forward).

Let $\mathrm{Inf}(\rho) = F$ and $|F| = m$. Choose k and $k' > k$ s.t. forall $j > k$ $\rho(j) \in F$ and $\{\rho(k), \ldots, \rho(k'-1)\} = F$.

By construction of ρ', the F-states $F = \{i_1, \ldots, i_m\}$ are at the beginning of $\rho'(k')$ and for every $k'' > k'$ the hit is always $\leq m$ (1).

Proof of Correctness

Proof (forward cont.)

For the hit equal to m the hit-segment must be the set F. So, for (2) it suffices to show that the hit is infinitely often equal to m. Assume the hit is only finitely often equal to m, then eventually the LAR-entries $i_m, i_{m+1}, \ldots, i_n$ are not changed anymore (and so, these states are not visited anymore). Then,$|Inf(\rho)| < m$, which contradcits $\text{Inf}(\rho) = F$ with $|F| = m$.

Proof (backwards).

Assume (1) and (2) holds. It follows from (1), that the LAR-entries $i_{m+1}, \ldots, i_n$ in ρ' are fixed from some point j_0 onwards. So, the states $i_{m+1}, \ldots, i_n$ are not visited anymore after j_0. From, (2) it follows that $i_{m+1}, \ldots, i_n$ are not in F (i.e., $\text{Inf}(\rho) \subseteq F$).

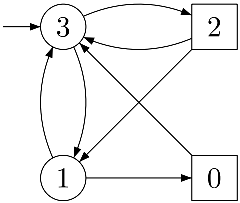
Proof of Correctness

Proof (backwards cont.)

For $F \subseteq \text{Inf}(\rho)$, assume $q \in F$ but $q \notin \text{Inf}(\rho)$.

Since $q \in F$ and hit-segment $= F$ infinitely often (2), we know that $q \in$ hit-segment infinitely often. Furthermore, since $|\text{hit-segment}| \leq m$ from some point on (1), it follows that from some point on the index i of q in the hit segment is $\leq m$. From $q \notin \text{Inf}(\rho)$ it follows that from some point onwards q can only stay in the same position in the LAR or go to the right and its final position i is $> m$. Contradiction.

Example



$$\rho \in \mathrm{Win} \leftrightarrow \{0, 2\} \subseteq \mathrm{Inf}(\rho)$$

$$\mathcal{F} = \{\{0, 2\}, \{0, 1, 2\}, \{0, 1, 2, 3\}\}$$

Summary

We can solve Muller games by reduction to parity games using the Last Appearence Record construction.

Summary

We can solve Muller games by reduction to parity games using the Last Appearence Record construction.

Finally, Rabin and Streett games can be viewed as Muller games.

Rabin and Streett Games

Given a game graph $G = (S, S_0, E)$ and a Rabin/Streett condition $\{(F_1, E_1), \ldots, (F_k, E_k)\}$, then a play ρ is winning for Player 0 for the

- Rabin condition if there exists (F_i, E_i) s.t. $\text{Inf}(\rho) \cap F_i \neq \emptyset \wedge \text{Inf}(\rho) \cap E_i = \emptyset$
- Streett condition if forall (F_i, E_i), we have that $\text{Inf}(\rho) \cap F_i \neq \emptyset \rightarrow \text{Inf}(\rho) \cap E_i \neq \emptyset$

Rabin and Streett to Muller Games

Simple reduction:

Given a Rabin (or Streett) game $(G, \mathcal{F})$ with $G = (S, S_0, E)$ and $\mathcal{F} = \{(F_1, E_1), \ldots, (F_k, E_k)\}$, there exists an equivalent Muller game $(G', \mathcal{F}')$ with $G' = G$ and

$$\mathcal{F}' = \{F \in 2^S \mid \exists i \in \{1, \ldots, k\} : F \cap F_i \neq \emptyset \wedge F \cap E_i = \emptyset\} \text{ (Rabin)}$$

$$\mathcal{F}' = \{F \in 2^S \mid \forall i \in \{1, \ldots, k\} : F \cap F_i \neq \emptyset \rightarrow F \cap E_i \neq \emptyset\} \text{ (Streett)}$$

Some interesting facts about Rabin/Streett games:

- In a Rabin game one of the players (Player 0) has a memoryless strategy.
- There is a special record set called Index Appearance record (IAR) optimized for Streett games. It records permutation and satisfaction of Streett-pair indices (not states).

Back to Tree Automata

Muller tree automaton

Recall, a Muller tree automaton over Σ is $A = (S, s_0, T, \mathcal{F})$, where

- S is a finite set of states,
- $s_0 \in S$ is an initial state,
- $T : S \times \Sigma \to 2^{S \times S}$ is a transition function
- $\mathcal{F} \subseteq 2^S$ is the set of accepting sets.

Given an input tree t, a run π of A over t is accepting iff for every path σ in t:

$$\mathrm{Inf}(\pi_{|\sigma}) \in \mathcal{F}$$

Parity tree automaton

A Parity tree automaton over Σ is $A = (S, s_0, T, p)$, where

- S is a finite set of states,
- $s_0 \in S$ is an initial state,
- $T : S \times \Sigma \to 2^{S \times S}$ is a transition function
- $p : S \to \{0, \ldots k\}$ is a priority function.

Given an input tree t, a run π of A over t is accepting iff for every path σ in t:

$$\min_{s \in \mathrm{Inf}(\pi_{|\sigma})} p(s) \text{ is even}$$

Example

A parity tree automaton over $\Sigma = \{a, b\}$ that recognizes all binary trees

$$\mathcal{T} = \{t \in \mathcal{T}^\omega(\Sigma) \mid \text{ each path through } t \text{ has only finitely many } b\}$$

- $S = \{q_a, q_b\}$
- $I = \{q_a, q_b\}$
- $T(q_a, a) = \{(q_a, q_a)\}$, $T(q_b, a) = \{(q_a, q_a)\}$
  $T(q_a, b) = \{(q_b, q_b)\}$, $T(q_b, b) = \{(q_b, q_b)\}$
- $p(q_a) = 2$, $p(q_b) = 1$

Tree Automata and Games

With any parity tree automaton $A = (S, s_0, T, p)$ over Σ and any input tree $t \in \mathcal{T}^\omega(\Sigma)$, we can associate a parity game between

- Player Automaton and
- Player Pathfinder

with proceeds as follows:

- First, Automaton picks a transition in T (from s_0) which matches the labels of the root of t
- Then Pathfinder decides on a direction (left or right) to proceed to a son of the root
- Then Automaton chooses again a transition for this node (and compatible with the first transition)
- Then Pathfinder reacts again by branching left or right...

Tree Automata and Games

Such a play give a sequence of transitions (and hence a sequence of states in S) built up along a path chosen by Pathfinder.
Automaton wins the play iff the sequence of states satisfies the parity condition.

Given a parity tree automaton $A = (S, s_0, T, p)$ over Σ and an input tree t, the *game graph* $G_{A,t} = (S_0 \cup S_1, S_0, E)$ is defined by

- $S_0 = \{(w, t(w), s) \mid w \in \{0,1\}^*, t(w) \in \Sigma, s \in S_0\}$,
- $S_1 = \{(w, t(w), \tau) \mid w \in \{0,1\}^*, t(w) \in \Sigma, \tau \in T\}$,

and the edges relation E is such that successive game positions are compatible with the transitions in A on t.
The priority of a triple $u = (w, t(w), s)$ or $(w, t(w), (s, t(w), s', s''))$ is the priority $p(s)$. (Standard initial position: $(\epsilon, t(\epsilon), s_0)$)

Tree Automata and Games

Lemma

The tree automaton A accepts an input tree t iff in the parity game over $G_{A,t}$ there is a winning strategy for player Automaton from the initial position $(\epsilon, t(\epsilon), s_0)$.

Proof.

A successful run of A on t yields a winning strategy for Automaton in the parity game over $G_{A,t}$: Along each path the suitable choice of transitions is fixed by the run.
Conversely, a winning strategy for Automaton over $G_{A,t}$ clearly provides a method to build up a successful run of A on t. Just apply this winning strategy along arbitrary paths.

Emptiness of Parity Tree Automata

Lemma

For each parity tree automata $A = (S, s_0, T, p)$ *over* Σ*, there exists an input-free tree automaton* A' *such that* $\mathcal{L}(A) \neq \emptyset$ *iff* A' *admits a successful run.*

Idea: build an automaton A' that guesses an input tree t

Proof.

Given $A = (S, s_0, T, p)$ over Σ, we construct $A' = (S \times \Sigma, s_0 \times \Sigma, T', p')$ that nondeterministically guesses an input tree t in the second component of its states.

$T' = \{((s, a), (s', x), (s'', y)) \mid (s, a, s', s'') \in T$ and $\exists p, p', r, r' : (s', x, p, p')$ and $(s'', y, r, r') \in T\}$ and $p'(s, a) = p(s)$ for all states (s, a). The behavior of A' and A on the guessed input t is identical.

Emptiness of Parity Tree Automata

For every input-free tree automaton $A = (S, s_0, T, p)$, we can associate a simpler parity game $((S_0 \cup S_1, S_0, E), p')$

- $S_0 = S$ and
- $S_1 = T = S \times S \times S$
- $\forall s \in S, (s, s', s'') \in T$, we have $(s, (s, s', s'')) \in E$ and $\forall (s, s', s'') \in T$ we have $((s, s', s''), s')$ and $((s, s', s''), s'') \in E$
- $p'((s, s', s'')) = p(s)$ and $p'(s) = p(s)$

Clearly, every strategy for Player 0 corresponds to a run and vice versa. So, every winning strategy corresponds to a successful run (vv)

Theorem

For parity tree automata it is deciable wheater their recognized language is empty or not.

Example

Consider the input-free tree automaton $A = (S, s_0, T, p)$ with $S = \{s_0, s_a, s_b, s_d\}$ and $T = \{(s_0, s_a, s_d), (s_0, s_d, s_b), (s_a, s_a, s_0), (s_a, s_d, s_a), (s_d, s_d, s_b), (s_b, s_b, s_d)\}$.

Parity $\leftrightarrow$ Muller

Theorem

1. *For any parity tree automaton one can construct an equivalent Muller tree automaton.*
2. *For any Muller tree automaton one can construct an equivalent parity tree automaton.*

Proof 2.

Given a parity tree automaton $A = (S, s_0, T, p)$ keep states and transitions and define $\mathcal{F}$ as follows:

$$\mathcal{F} = \{F \in 2^S \mid \min_{s \in F} p(s) \text{ is even}\}$$

Parity $\leftrightarrow$ Muller

Proof 1.

Copy the simulation of Muller games by parity games. Given a Muller tree automaton with state set S use for the parity tree automaton the state set $\mathrm{LAR}(S)$ and define the transition according to the LAR update rule.

Allow transition

$$((s_1 \dots s_n, i), a, (s'_1 \dots s'_n, j), (s''_1 \dots s''_n, k))$$

for transition (s_1, a, s'_1, s''_1) of the Muller automaton, where

- $(s'_1 \dots s'_n, j)$ is the LAR update for a visit to s'_1 and
- $(s''_1 \dots s''_n, k)$ is the LAR update for a visit to s''_1.

Define priorities as in the simulation of Muller games by parity games.

Summary: Tree Automaton

- Tree Automata can be viewed as games between Automaton and Pathfinder
- Parity and Muller tree automata can be reduced to each other
- (Same holds for Rabin/Streett, Parity, and Muller tree automata)
- Radu showed closure properties of Muller tree automaton (union, intersection, projection)
- Missing: complementation

Complementation of Parity Tree Automaton

We will show basic idea.

- To complement a given automaton A means to construct an automaton B s.t.

$$t \notin A \leftrightarrow t \in B$$

- Due to the run lemma, complementation means to conclude from the non-existence of a winning strategy of Player Automaton in $G_{A,t}$ that there exists a winning strategy of Automaton in $G_{B,t}$.

Proof has two steps:

1. use determinacy of parity games to show that if Automaton has no winning strategy over $G_{A,t}$, then Pathfinder has a winning strategy over $G_{A,t}$ (from $(\epsilon, t(\epsilon), s_0)$)
2. Convert Pathfinder's strategy into an Automaton strategy.

Complementation of Parity Tree Automaton

Theorem

For any parity tree automaton A over Σ, one can construct a Muller tree automaton (and therefore a parity tree automaton) B over Σ that recognizes $\mathcal{T}^\omega(\Sigma) \setminus \mathcal{L}(A)$

Proof.

From Step 1 (determinacy of parity games), we know there exists a (memoryless) winning strategy $f : S_1 \rightarrow \{0, 1\}$ for Player Pathfinder.

$$f : \{0, 1\}^* \times \Sigma \times T \rightarrow \{0, 1\}$$

decompose f into a family of strategies parameterized by $w \in \{0, 1\}^*$

$$f_w : \Sigma \times T \rightarrow \{0, 1\}$$

Complementation of Parity Tree Automaton

Let I be the set of all possible local instructions $i : \Sigma \times T \to \{0, 1\}$. Then, f can be represented as I-labeled binary tree s with $s(w) = f_w$.

Let $s \cdot t$ be the corresponding $(I \times \Sigma)$-labeled tree

$$s \cdot t(w) = (s(w), t(w)) \text{ for } w \in \{0, 1\}^*.$$

Since f exists, we know there is an I-labeled tree s s.t. for all sequences $\tau_0 \tau_1 \ldots$ of transitions chosen by Automaton and for all paths (in path for the unique) $\pi \in \{0, 1\}^*$, the generated state sequence violates the parity condition.
Intuitively, f tells the "new" automaton for every tree $t \notin \mathcal{L}(A)$ which path to track for a given transition sequences in order to reject/accept the tree t.

Complementation of Parity Tree Automaton

So, we know:

1. There exists an I-labeled tree s such that $s \cdot t$ satisfies
   2. for all $\pi \in \{0, 1\}^\omega$
      3. for all $\tau_0\tau_1 \dots \in T^\omega$
         4. if the sequence $s_{|\pi}$ of local instructions applied to the sequence of tree labels $t_{|\pi}$ and the sequence $\tau_0\tau_1 \dots$ produces the path π, then the state sequence determined by $\tau_0\tau_1 \dots$ violates the parity condition.

Complementation of Parity Tree Automaton

- Condition 4 is a property of ω-words over $I \times \Sigma \times T \times \{0, 1\}$, which can be checked by a Muller word automaton M_4.
- Condition 3 is a property of ω-words over $I \times \Sigma \times \{0, 1\}$ checked by M_3, which results from M_4 by universally quantifying T (negate, project, negate).
- Condition 2 is a property of $(I \times \Sigma)$-labeled trees, which can be checked by a Muller tree automaton M_2 that simulates M_3 along each path.
- Condition 1, apply nondeterminism, a Muller tree automaton B can be built by guessing a tree s on the input tree t and running M_2 on $s \cdot t$.