Algorithms for Model Checking (2IW55)

Lecture 12

The Recursive Algorithm for Parity games

Background material:

"Recursive Solving of Parity Games Requires Exponential Time", Oliver Friedmann

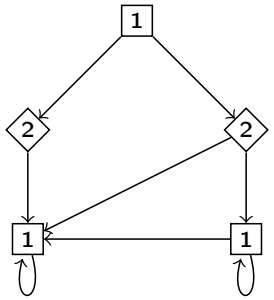
December 12, 2011

Department of Mathematics and Computer Science

TU/e Technische Universiteit Eindhoven University of Technology

Parity games (recap)

Identify graph, priorities, owners, plays, and strategies in the following parity game.



Department of Mathematics and Computer Science

TU/e Technische Universiteit Eindhoven University of Technology

Transformations on parity games

- Self-loop elimination (vs Local resolution)
- Priority compaction
- Priority propagation
- Bisimulation minimisation

Bisimulation

Definition (Bisimilarity of vertices)

Let $G = (V, E, \mathrm{p}, (V_\Diamond, V_\Box))$ be a parity game. Let R be a symmetric relation. R is a bisimulation relation if $v\,R\,v'$ implies

- $v \in V_\Diamond \Leftrightarrow v' \in V_\Diamond$
- $\mathrm{p}(v) = \mathrm{p}(v')$
- $v \to w$ implies $\exists w'$ such that $v' \to w'$ and $w\,R\,w'$

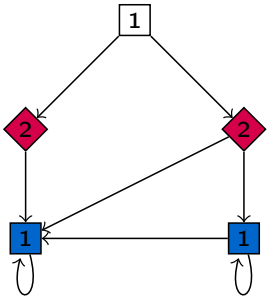
Vertices v and v' are bisimilar ($v \equiv v'$) iff there exists a bisimulation relation R such that $v\,R\,v'$.

Theorem

$v \equiv v'$ implies that v and v' are won by the same player

Bisimilarity (example)

Original



Minimal bisimilar parity game

Goal

Let $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game.

- There is a unique partition $(W_\Diamond, W_\Box)$ of V such that:
  - $\Diamond$ has winning strategy $\varrho_\Diamond$ from $W_\Diamond$, and
  - $\Box$ has winning strategy $\varrho_\Box$ from $W_\Box$.

Goal of parity game algorithms

Compute partitioning $(W_\Diamond, W_\Box)$ with strategies $\varrho_\Diamond$ and $\varrho_\Box$ of V, such that $\varrho_\Diamond$ is winning for player $\Diamond$ from $W_\Diamond$ and $\varrho_\Box$ is winning for player $\Box$ from $W_\Box$.

Notation

Let $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game.
We use the following notation:

- $\overline{\Diamond}$ is $\Box$, $\overline{\Box}$ is $\Diamond$
- $G \setminus U$ is parity game G restricted to the vertices outside U. Formally $G \setminus U = (V', E', p', (V'_\Diamond, V'_\Box))$, with
  - $V' = V \setminus U$,
  - $E' = E \cap (V \setminus U)^2$,
  - $p'(v) = p(v)$ for $v \in V \setminus U$,
  - $V'_\Diamond = V_\Diamond \setminus U$, and
  - $V'_\Box = V_\Box \setminus U$

Recursive algorithm (intuition)

- Divide and conquer
- Base: empty game
- Step: assemble winning sets/strategies from
  - winning sets/strategies of subgames
  - attractor strategy for one of players reaching set of nodes with minimal priority in the game

Attractor sets

The attractor set for $\bigcirc$ and set $U \subseteq V$ is the set of vertices such that $\bigcirc$ can force any play to reach U.

Definition

Let $U \subseteq V$. We define the attractor sets inductively as follows:

$$Attr^0_{\bigcirc}(G, U) = U$$

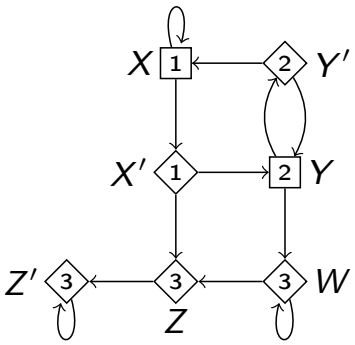
$$\begin{aligned} Attr^{k+1}_{\bigcirc}(G, U) &= Attr^k_{\bigcirc}(G, U) \\ &\cup (V_{\bigcirc} \cap \{v \in V \mid \exists v' \in V : (v, v') \in E \wedge v' \in Attr^k_{\bigcirc}(G, U)\}) \\ &\cup (V_{\overline{\bigcirc}} \cap \{v \in V \mid \forall v' \in V : (v, v') \in E \implies v' \in Attr^k_{\bigcirc}(G, U)\}) \end{aligned}$$

$$Attr_{\bigcirc}(G, U) = \bigcup_{k \in \mathbb{N}} Attr^k_{\bigcirc}(G, U)$$

Example of attractor sets

Example

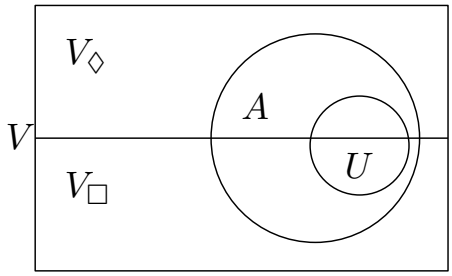
Consider parity game G:



Compute:

- $Attr_{\Diamond}(G, \{Z\}) = \{Z, X', W\}$
- $Attr_{\Box}(G, \{W\}) = \{W, Y\}$

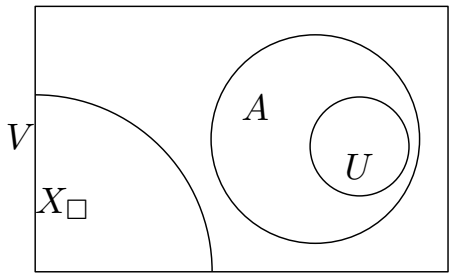
Observations



Let $U \subseteq V$. Let $A = Attr_\Diamond(G, U)$.

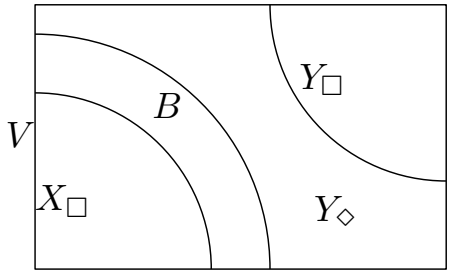
- $\Diamond$ cannot escape from $V \setminus A$. If it could, there would be an edge $(v, v') \in E$, such that $v \in V_\Diamond \setminus A$, and $v' \in A$, but then by definition also $v \in A$, which is not the case.
- $\Box$ cannot escape from A. If it could, there would be an edge $(v, v') \in E$, such that $v \in V_\Box \cap A$, and $v' \notin A$, but then by definition $v \notin A$.

Observations



Let $U \subseteq V$. Let $A = Attr_\Diamond(G, U)$.
Assume:

- $X_\Box$ is winning set for $\Box$ on $G \setminus A$;
- $B = Attr_\Box(G, X_\Box)$;
- $Y_\Diamond$ is winning set for $\Diamond$ on $G \setminus B$;
- $Y_\Box$ is winning set for $\Box$ on $G \setminus B$.



Then:

- Player $\Diamond$ can never leave B;
- Player $\Box$ can never leave $V \setminus B$;
- A winning strategy for player $\Box$ in $G \setminus (V \setminus B)$ from $V_\Box \cap B$ is also a winning strategy for player $\Box$ in G from $V_\Box \cap B$.

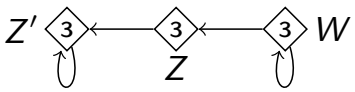
Recursive algorithm (McNaughton '93, Zielonka '98)

Recursively solve a parity game: $\mathit{Recursive}(G)$. Returns partitioning $(W_\Diamond, W_\Box)$ such that $\Diamond$ wins from $W_\Diamond$, and $\Box$ wins from $W_\Box$.

1: **if** $V_G = \emptyset$ **then**
2: $\quad W_\Diamond \leftarrow \emptyset$
3: $\quad W_\Box \leftarrow \emptyset$
4: $\quad$ **return** $(W_\Diamond, W_\Box)$
5: **end if**
6: $m \leftarrow \min\{p(v) \mid v \in V\}$
(* Paper: max *)
7: $\bigcirc \leftarrow \begin{cases} \Diamond & \text{if m is even} \\ \Box & \text{otherwise} \end{cases}$
8: $U \leftarrow \{v \in V \mid p(v) = m\}$
9: $A \leftarrow \mathit{Attr}_\bigcirc(G, U)$
10: $(X_\Diamond, X_\Box) \leftarrow \mathit{Recursive}(G \setminus A)$
12: **if** $X_{\overline{\bigcirc}} = \emptyset$ **then**
13: $\quad W_\bigcirc \leftarrow A \cup X_\bigcirc$
14: $\quad W_{\overline{\bigcirc}} \leftarrow \emptyset$
15: **else**
16: $\quad B \leftarrow \mathit{Attr}_{\overline{\bigcirc}}(G, X_{\overline{\bigcirc}})$
17: $\quad (Y_\Diamond, Y_\Box) \leftarrow \mathit{Recursive}(G \setminus B)$
18: $\quad W_\bigcirc \leftarrow Y_\bigcirc$
19: $\quad W_{\overline{\bigcirc}} \leftarrow B \cup Y_{\overline{\bigcirc}}$
20: **end if**
21: **return** $(W_\Diamond, W_\Box)$

Exercise

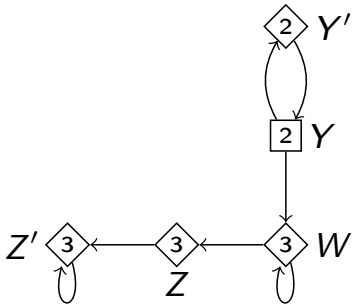
Apply the recursive algorithm to the following parity game G



$m \leftarrow 3$
$\bigcirc \leftarrow \Box$
$U \leftarrow \{v \in V \mid p(v) = 3\} = \{W, Z, Z'\}$
$A \leftarrow \mathit{Attr}_\Box(G, U) = \{W, Z, Z'\}$
$(X_\Diamond, X_\Box) \leftarrow \mathit{Recursive}(G \setminus V) = (\emptyset, \emptyset)$
if $X_\Diamond = \emptyset$ **then**
$\quad W_\Box \leftarrow A \cup X_\Box = A = \{W, Z, Z'\}$
$\quad W_\Diamond \leftarrow \emptyset$
else
$\quad \ldots$
end if
return $(W_\Diamond, W_\Box) = (\emptyset, \{W, Z, Z'\})$

Exercise

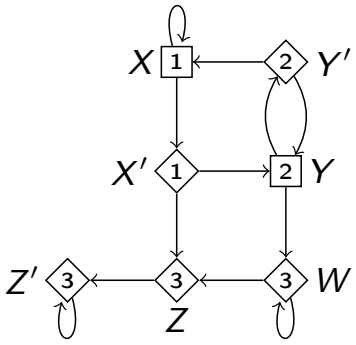
Apply the recursive algorithm to the following parity game G



6: $m \leftarrow 2$
7: $\bigcirc \leftarrow \diamond$
8: $U \leftarrow \{v \in V \mid p(v) = 2\} = \{Y, Y'\}$
9: $A \leftarrow Attr_{\diamond}(G, U) = \{Y, Y'\}$
10: $(X_{\diamond}, X_{\square}) \leftarrow Recursive(G \setminus \{Y, Y'\}) = (\emptyset, \{Z, Z', W\})$
11: **if** $X_{\square} = \emptyset$ **then**
12: . . .
14: **else**
15: $B \leftarrow Attr_{\square}(G, X_{\square}) = \{Y, Y', Z, Z', W\}$
16: $(Y_{\diamond}, Y_{\square}) \leftarrow Recursive(G \setminus V) = (\emptyset, \emptyset)$
17: $W_{\diamond} \leftarrow Y_{\diamond} = \emptyset$
18: $W_{\square} \leftarrow B \cup Y_{\square} = B = \{Y, Y', Z, Z', W\}$
19: **end if**
20: **return** $(W_{\diamond}, W_{\square}) = (\emptyset, \{Y, Y', Z, Z', W\})$

Example (*Recursive*(G))

Consider parity game G:



6: $m \leftarrow 1$
7: $\bigcirc \leftarrow \square$
8: $U \leftarrow \{v \in V \mid p(v) = 1\} = \{X, X'\}$
9: $A \leftarrow Attr_{\square}(G, U) = \{X, X'\}$
10: $(X_{\diamond}, X_{\square}) \leftarrow Recursive(G \setminus \{X, X'\}) = (\emptyset, \{Y, Y', Z, Z', W\})$
11: **if** $X_{\diamond} = \emptyset$ **then**
12: $W_{\square} \leftarrow A \cup X_{\diamond} = \{X, X', Y, Y', Z, Z', W\}$
13: $W_{\diamond} \leftarrow \emptyset$
14: **else**
15: . . .
19: **end if**
20: **return** $(W_{\diamond}, W_{\square}) = (\emptyset, \{X, X', Y, Y', Z, Z', W\})$

So, player $\square$ wins from all vertices!

Complexity

Let $G = (V, E, p, (V_\Diamond, V_\Box)$ be a parity game. $n = |V|, e = |E|, d = |\{p(v) \mid v \in V\}|$.

Worst-case running time complexity

$$\mathcal{O}(e \cdot n^d)$$

Lowerbound on worst-case:

$$\Omega(fib(n)) = \Omega((\frac{1+\sqrt{5}}{2})^n)$$

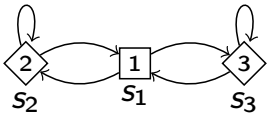
Complexity

Let $G = (V, E, p, (V_\Diamond, V_\Box)$ be a parity game; $n = |V|, e = |E|, d = |\{p(v) \mid v \in V\}|$.

- Algorithm with best known upper bound: Big step algorithm due to Schewe, with complexity
$$\mathcal{O}(d \cdot n^{d/3})$$
- Big step combines recursive algorithm with small progress measures;
- Small progress measures will be discussed next lecture

Exercise

Consider the following parity game:



- Compute the winning sets $W_\diamond, W_\square$ for players $\diamond$ and $\square$ in this parity game using the recursive algorithm.
- Translate this parity game to BES and solve the BES using Gauss elimination.