Algorithms for Model Checking (2IW55)

Lecture 7: Recursively Solving Parity Games

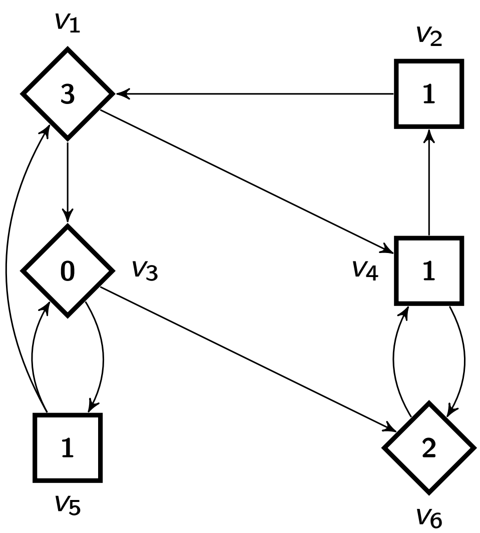
Background material:

O. Friedmann, *Recursive Solving of Parity Games Requires Exponential Time*

M. Gazda and T.A.C. Willemse, *Zielonka's Recursive Algorithm: dull, weak and solitaire games and tighter bounds*

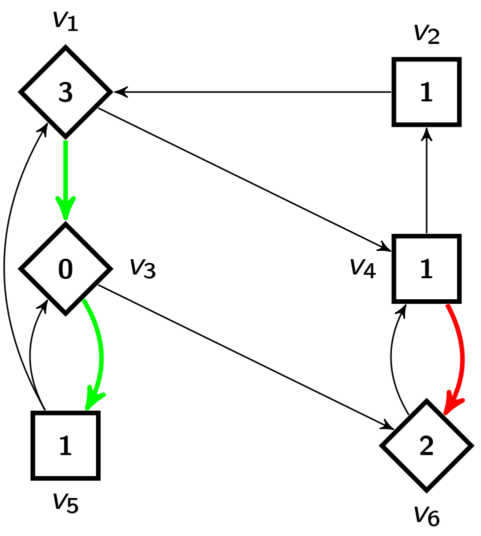
Tim Willemse
(timw@win.tue.nl)
http://www.win.tue.nl/∼timw
MF 6.073

Parity games—recap



- two players: $\Diamond$ (Even) and $\Box$ (Odd)
- every node has an owner ($V = V_\Diamond \cup V_\Box$)
- moving token indefinitely; node owner chooses the next vertex
- play = infinite path through the game
- vertices labelled with natural numbers (priorities)
- winner of a play: determined by the parity of the minimal priority occurring infinitely often ($\Diamond$ wins even parity, $\Box$ wins odd parity)

Parity games—recap



- strategy
  - winning strategy
  - memoryless strategy
- winning partition

Objective

Parity game $G = (V, E, p, (V_\Diamond, V_\Box))$.

Determinacy implies 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 $W_\Box$.

Objective of parity game algorithms

Compute partition $(W_\Diamond, W_\Box)$ with strategies $\varrho_\Diamond$ and $\varrho_\Box$ of V such that:

- $\varrho_\Diamond$ is winning for player $\Diamond$ from $W_\Diamond$
- $\varrho_\Box$ is winning for player $\Box$ from $W_\Box$.

Parity game algorithms

Deterministic algorithms for solving parity games

- Recursive *(this lecture)* . . . . McNaughton '93, Zielonka '98
- Local algorithm . . . . Stevens & Stirling '98
- Small progress measures *(next lecture)* . . . . Jurdziński, '00
- Strategy improvement . . . . Vöge & Jurziński '00
- (Deterministic) Subexponential . . . . Jurdziński, Paterson & Zwick '06
- Bigstep . . . . Schewe '07

Concepts

Parity game $G = (V, E, p, (V_\Diamond, V_\Box))$.

Notation:

- $\bigcirc$ is the 'arbitrary' player . . . . $\bigcirc \in \{\Diamond, \Box\}$
- $\overline{\bigcirc}$ is the opponent . . . . $\overline{\Diamond} = \Box$ and $\overline{\Box} = \Diamond$

Definition (Arena restriction)

The game $G \setminus U = (V', E', p', (V'_\Diamond, V'_\Box))$, for $U \subseteq V$, is the game confined to $V \setminus U$:

- $V' = V \setminus U$ and $E' = E \cap (V' \times V')$,
- $p'(v) = p(v)$ for $v \in V \setminus U$,
- $V'_\Diamond = V_\Diamond \setminus U$, and $V'_\Box = V_\Box \setminus U$

Concepts

Parity game $G = (V, E, p, (V_\Diamond, V_\Box))$.

Definition (Closed strategies)

Strategy $\varrho_\Diamond : V_\Diamond \to V$ is closed on $W \subseteq V$ if for all $v \in W$, we have:

- $v \in V_\Diamond$ implies $\varrho_\Diamond(v) \in W$, and
- $v \in V_\Box$ implies that $w \in W$ for all $(v, w) \in E$

For $\varrho_\Diamond$ closed on W, plays consistent with $\varrho_\Diamond$ and starting in W stay within W

Definition (Closed sets)

Set $W \subseteq V$ is $\Diamond$-closed if $\Diamond$ has a strategy closed on W. Likewise for $\Box$-closed.

Concepts

Parity game $G = (V, E, p, (V_\Diamond, V_\Box))$.

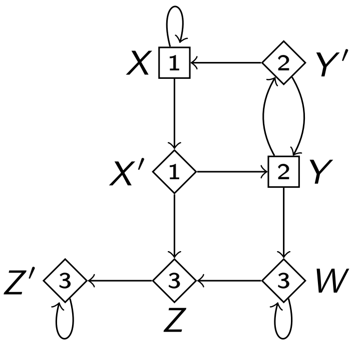
Definition (Dominion)

$D \subseteq W \subseteq W_\bigcirc$ is a dominion of $\bigcirc$, if she has a memoryless strategy ϱ that is:

- winning for $\bigcirc$ from all $v \in D$
- closed on D

Example (Dominions)

Consider parity game G:



- $\{X\}$, $\{Z', Z, W\}$ are $\Box$-dominions
- Note that $\{Z, W\}$ and $\{Y, Y'\}$ are no dominions (why?)

Parity game $G = (V, E, p, (V_\diamond, V_\Box))$.

Definition (Attractor sets)

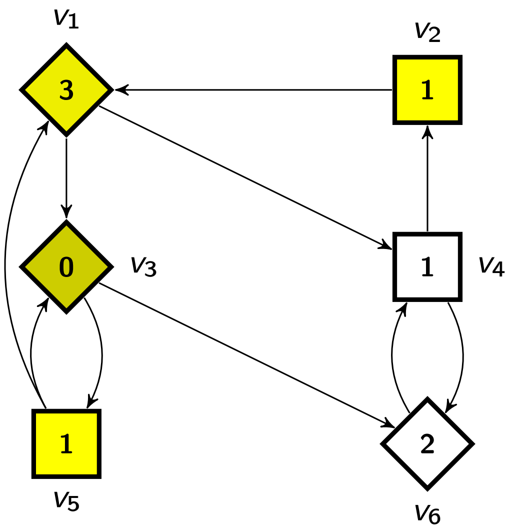
The attractor set to $U \subseteq V$ for $\bigcirc$ *(denoted* $\bigcirc$-$Attr(G, U)$*)* is the least set of vertices:

- containing U
- such that $\bigcirc$ can force any play to reach U.

Inductively: $\bigcirc\text{-}Attr(G, U) = \bigcup_{k \in \mathbb{N}} \bigcirc\text{-}Attr^k(G, U)$ where

$$\begin{aligned}
\bigcirc\text{-}Attr^0(G, U) &= U \\
\bigcirc\text{-}Attr^{k+1}(G, U) &= \bigcirc\text{-}Attr^k(G, U) \cup \\
&\{v \in V_\bigcirc \mid \exists v' \in V : (v, v') \in E \wedge v' \in \bigcirc\text{-}Attr^k(G, U)\} \cup \\
&\{v \in V_{\overline{\bigcirc}} \mid \forall v' \in V : (v, v') \in E \implies v' \in \bigcirc\text{-}Attr^k(G, U)\})
\end{aligned}$$

Example (Attractor sets)



$\bigcirc$-$Attr(G, U)$: vertices from which $\bigcirc$ can force the play to reach set U

Consider $\diamond$-$Attr(G, \{v_3\})$

$$\begin{aligned} \diamond\text{-}Attr^0(G, \{v_3\}) &= \{v_3\} \\ \diamond\text{-}Attr^1(G, \{v_3\}) &= \{v_1, v_3\} \\ \diamond\text{-}Attr^2(G, \{v_3\}) &= \{v_1, v_2, v_3, v_5\} \end{aligned}$$

Time to compute attractor: $\mathcal{O}(|V| + |E|)$

Parity game $G = (V, E, p, (V_\diamond, V_\square))$.

If U is a $\diamond$-dominion (dually for $\square$-dominion) in G then (by definition)

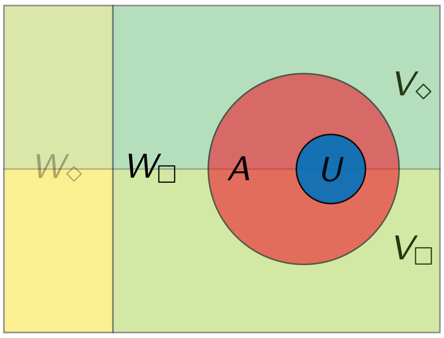
- there is a strategy ϱ such that $\diamond$ wins U
- $\diamond$ can always choose to stay in U
- $\square$ cannot leave U (it is a trap)

...but also:

- $A = \diamond$-$Attr(G, U)$ is an $\diamond$-dominion;
- $\diamond$ cannot leave $V \setminus A$
- If $(W_\diamond, W_\square)$ is solution of $G \setminus A$, then $(W_\diamond \cup A, W_\square)$ is solution of G.

Concepts

Visually:



- U is a $\diamond$-dominion
- $A = \text{-}Attr^\diamond(G, U)$
- A is a $\diamond$-dominion
- $(W_\diamond, W_\Box)$ winning sets $G \setminus A$
- $(W_\diamond \cup A, W_\Box)$ winning sets $G \setminus A$
- $\Box$ cannot leave A
- $\diamond$ can stay in A
- $\diamond$ cannot leave $V \setminus A$
- $\Box$ can avoid A from $V \setminus A$

Recursively solving parity games

Divide and conquer

- Base: trivial games with at most one priority
- Step:
  - Compute dominion
  - Solve remaining subgame
  - 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

Recursive solving parity games

Parity game $G = (V, E, p, (V_\Diamond, V_\Box))$.

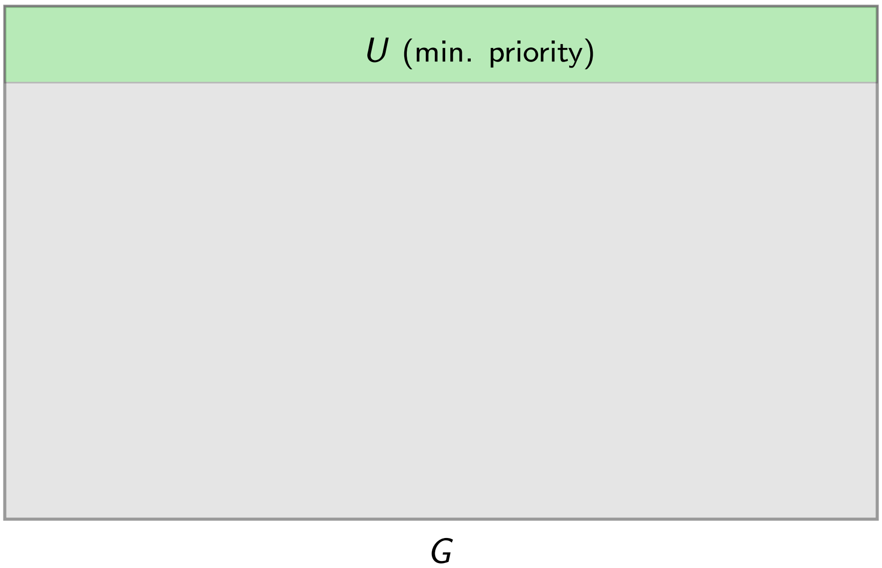
Recursive(G): recursively solve parity game G
Return: partitioning $(W_\Diamond, W_\Box)$ where $\Diamond$ wins from $W_\Diamond$, and $\Box$ wins from $W_\Box$

$m \leftarrow \min\{p(v) \mid v \in V\}$
$h \leftarrow \max\{p(v) \mid v \in V\}$
if $h = m$ or $V = \emptyset$ **then**
  if m is even or $V = \emptyset$ **then**
    return $(V, \emptyset)$
  else
    return $(\emptyset, V)$
  end if
end if
$\bigcirc \leftarrow \Diamond$ if m is even and $\Box$ otherwise
$U \leftarrow \{v \in V \mid p(v) = m\}$
$A \leftarrow \bigcirc\text{-}Attr(G, U)$
$(W'_\Diamond, W'_\Box) \leftarrow Recursive(G \setminus A)$
if $W'_{\overline{\bigcirc}} = \emptyset$ **then**
  $W_\bigcirc \leftarrow A \cup W'_\bigcirc$
  $W_{\overline{\bigcirc}} \leftarrow \emptyset$
else
  $B \leftarrow \overline{\bigcirc}\text{-}Attr(G, W'_{\overline{\bigcirc}})$
  $(W_\Diamond, W_\Box) \leftarrow Recursive(G \setminus B)$
  $W_{\overline{\bigcirc}} \leftarrow W_{\overline{\bigcirc}} \cup B$
end if
return $(W_\Diamond, W_\Box)$

Assume that the minimal priority in G is even

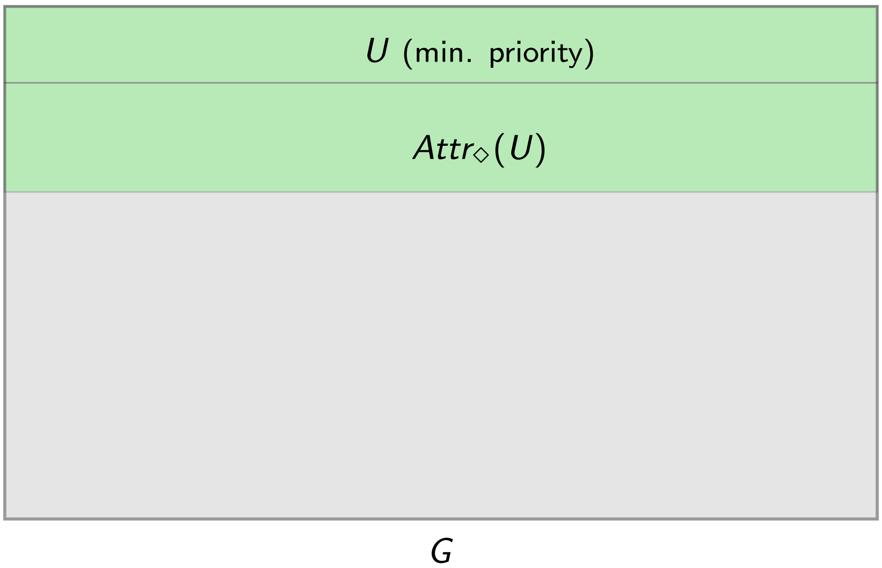
G

Assume that the minimal priority in G is even



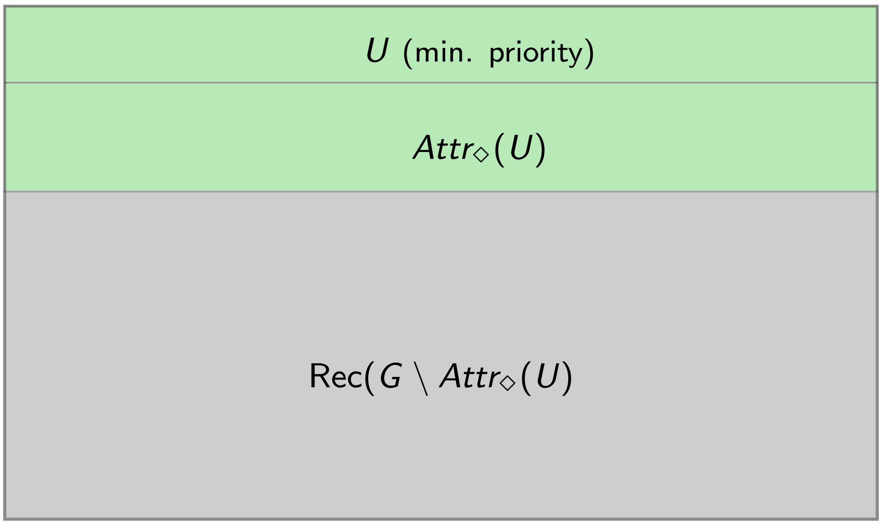
line 11

Assume that the minimal priority in G is even



line 12

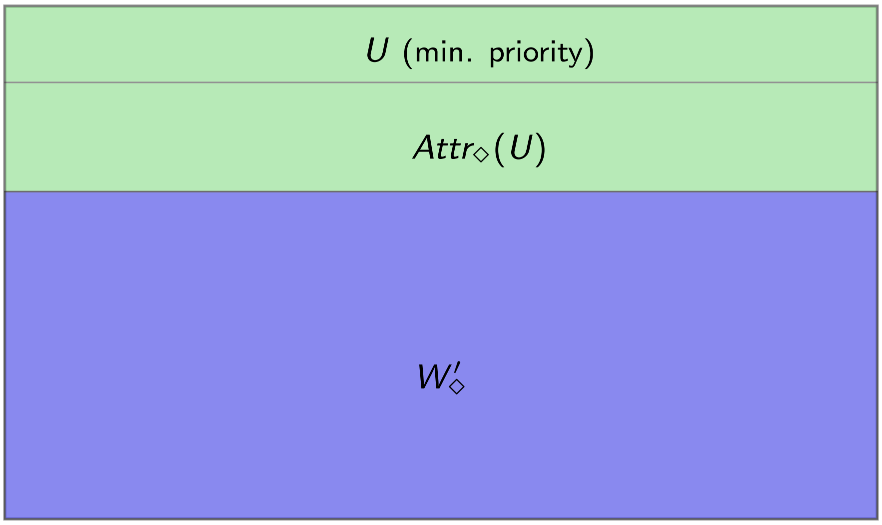
Assume that the minimal priority in G is even



G

line 13

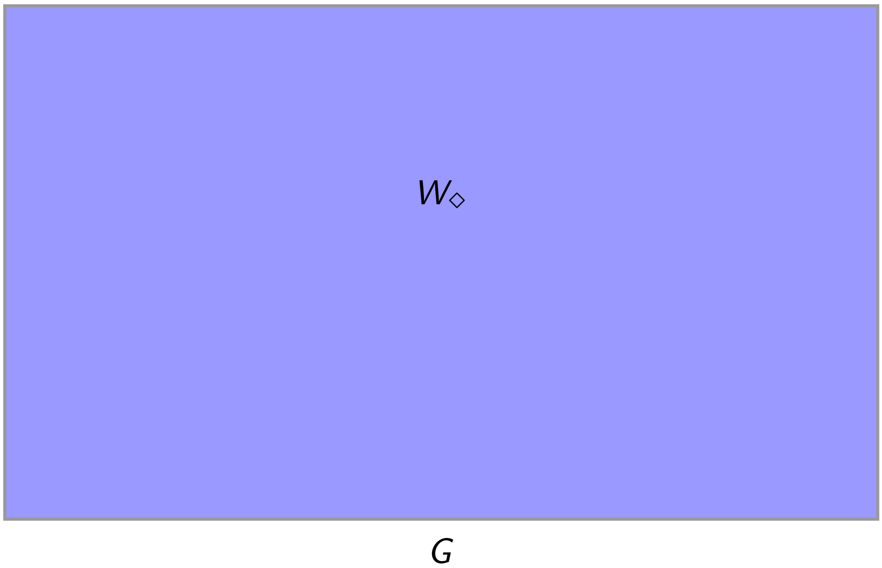
Assume that the minimal priority in G is even



G

line 14 (case $W'_\Box = \emptyset$)

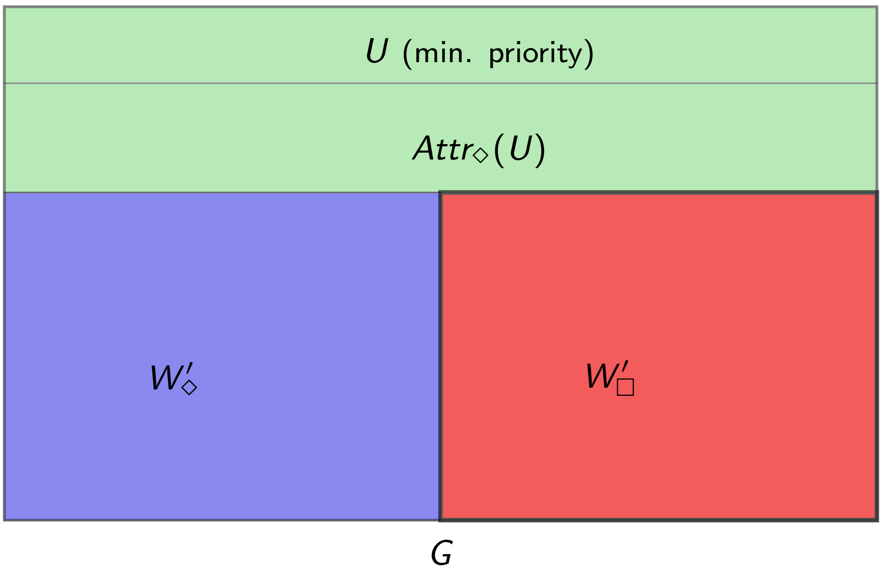
Assume that the minimal priority in G is even



line 15, 16 & 22 (case $W'_\square = \emptyset$)

TU/e Technische Universiteit Eindhoven University of Technology

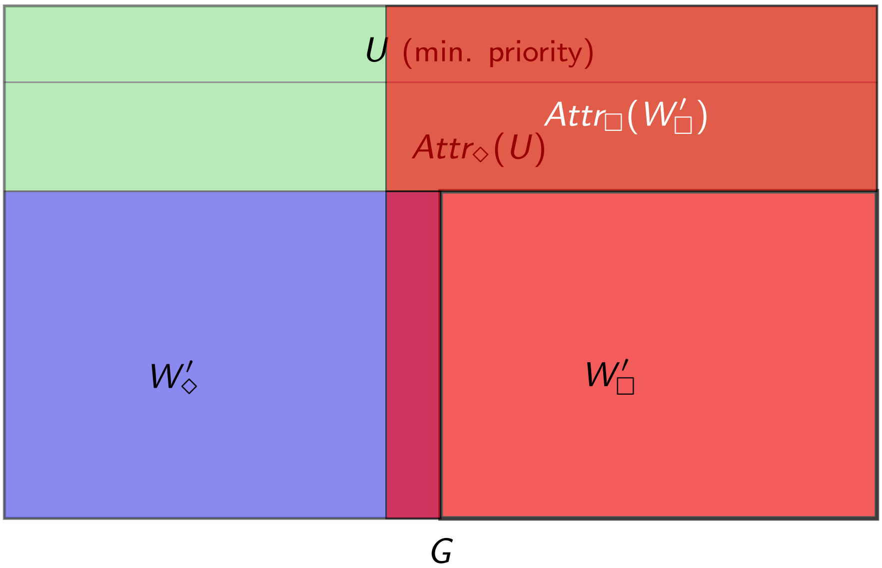
Assume that the minimal priority in G is even



line 17 (case $W'_\square \neq \emptyset$)

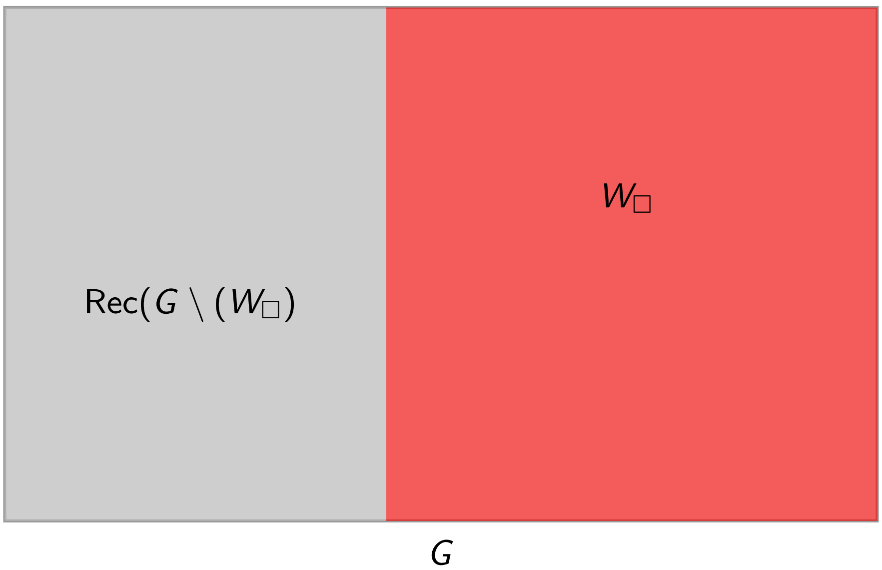
TU/e Technische Universiteit Eindhoven University of Technology

Assume that the minimal priority in G is even



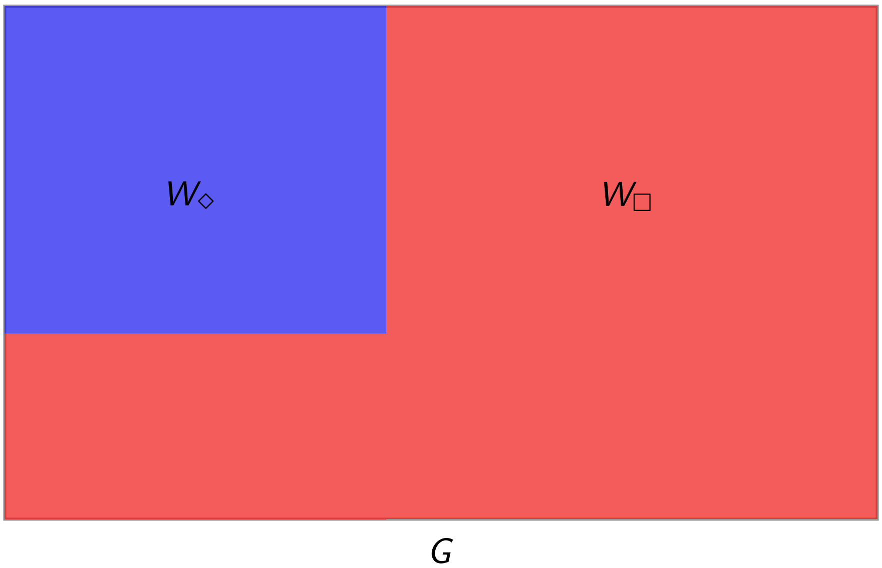
line 18

Assume that the minimal priority in G is even



line 19

Assume that the minimal priority in G is even



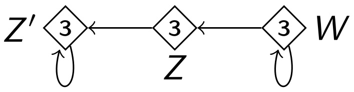
line 22

Observations

- Lines 1-9: base case, straightforward.
- Lines 10-13: try to establish a dominion. Two cases:
  - Lines 12-15: ($\bigcirc$ wins all):$\bigcirc$ wins in $G \setminus A$, then $\bigcirc$ wins all of G, since if $\overline{\bigcirc}$ visits A, then $\bigcirc$ plays towards U using attractor, visiting A infinitely often, hence m infinitely often. If A not visited, game stays in $G \setminus A$.
  - Lines 16-20: ($\overline{\bigcirc}$-dominion found): $W'_{\overline{\bigcirc}}$ is a $\overline{\bigcirc}$-dominion in $G \setminus A$. Since $\bigcirc$ cannot leave $G \setminus A$ also $W'_{\overline{\bigcirc}}$ is $\overline{\bigcirc}$-dominion in G. Then solve remaining game recursively and fix solution, compose strategies.

Exercise

Apply the recursive algorithm to the following parity game G



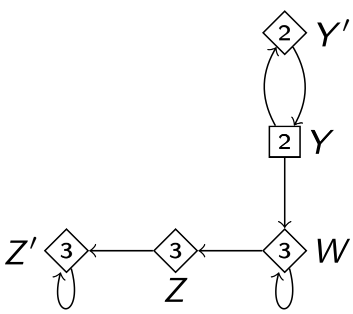
$m \leftarrow 3$

$h \leftarrow 3$

return $(\emptyset, \{W, Z, Z'\})$

Exercise

Apply the recursive algorithm to the following parity game G



$m \leftarrow 2$

$h \leftarrow 3$

...

$\bigcirc \leftarrow \diamond$

$U \leftarrow \{v \in V \mid p(v) = 2\} = \{Y, Y'\}$

$A \leftarrow \text{-}Attr^{\diamond}(G, U) = \{Y, Y'\}$

$(W'_{\diamond}, W'_{\square}) \leftarrow Recursive(G \setminus \{Y, Y'\}) = (\emptyset, \{Z, Z', W\})$

if $W'_{\square} = \emptyset$ **then**

...

else

$B \leftarrow \text{-}Attr^{\square}(G, W'_{\square}) = \{Y, Y', Z, Z', W\}$

$(W_{\diamond}, W_{\square}) \leftarrow Recursive(G \setminus B) = (\emptyset, \emptyset)$

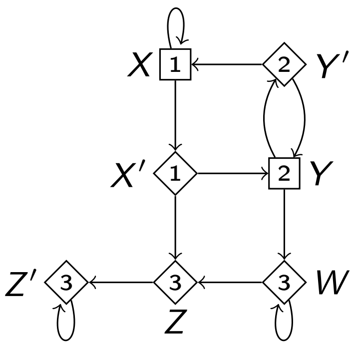
$W_{\square} \leftarrow W_{\square} \cup B = B = \{Y, Y', Z, Z', W\}$

end if

return $(W_{\diamond}, W_{\square}) = (\emptyset, \{Y, Y', Z, Z', W\})$

Exercise

Consider parity game G:




```
m ← 1
h ← 3
. . .
○ ← □
U ← {v ∈ V | p(v) = 1} = {X, X′}
A ← -Attr^□(G, U) = {X, X′}
(W′_◇, W′_□) ← Recursive(G \ {X, X′}) = (∅, {Y, Y′, Z, Z′, W})
if W′_◇ = ∅ then
    W_□ ← A ∪ W′_□ = {X, X′, Y, Y′, Z, Z′, W}
    W_◇ ← ∅
else
    . . .
end if
return (W_◇, W_□) = (∅, {X, X′, Y, Y′, Z, Z′, W})
```

So, player □ wins from all vertices!

Complexity

Parity game $G = (V, E, p, (V_\diamond, V_\square))$.

$n = |V|, m = |E|, d = |\{p(v) \mid v \in V\}|$.

- Worst-case running time complexity . . . . $\mathcal{O}(m \cdot n^d)$
- Lowerbound on worst-case (Gazda&Willemse '13) . . . . $\Omega(2^{n/3})$

Special cases (Gazda&Willemse '13):

- Basic algorithm:
  - weak games (Gazda&Willemse '13) . . . . $\mathcal{O}(d \cdot (n + m))$
  - (nested) solitaire games . . . . $\Omega(2^{n/3})$
  - dull games . . . . $\Omega(2^{n/3})$
- Optimised with SCC decomposition
  - (nested) solitaire games . . . . $\mathcal{O}(n \cdot (n + m))$
  - dull games . . . . $\mathcal{O}(n \cdot (n + m))$

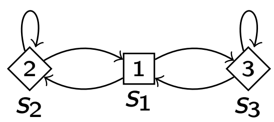
Wrap up

- Recursive algorithm:
  - Divide and conquer
  - Dominions
  - Attractor sets
  - $\mathcal{O}(m \cdot n^d)$
  - Exponential examples available
- Other algorithms:
  - Iterative (e.g. small progress measures)
  - Variations of recursive: start with other dominions

Technische Universiteit Eindhoven University of Technology

Exercise

Consider the following parity game:



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

Technische Universiteit Eindhoven University of Technology