# Algorithms for Model Checking (2IMF35)

## Lecture 7: Recursively Solving Parity Games

Background material:

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

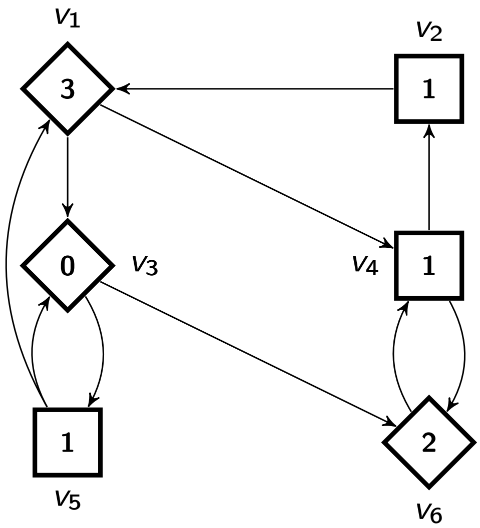
M. Gazda and T.A.C. Willemse, *Zielonka's Recursive Algorithm: dull, weak and solitaire games and tighter bounds*

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



## Parity games—recap

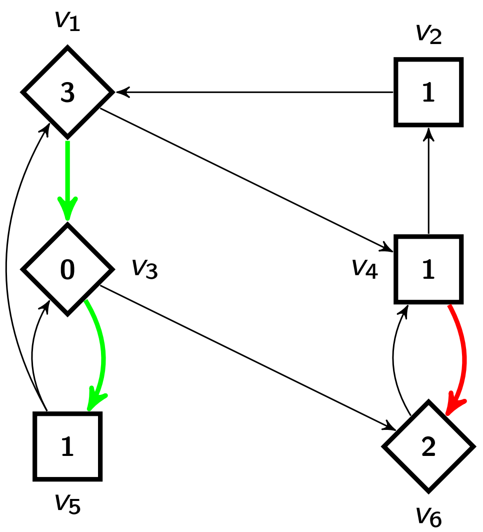





- two players: $\Diamond$ (Even) and $\Box$ (Odd)
- every node has an owner ($V = V_\Diamond \cup V_\Box$)
- moving token indefinitely; node owner chooses the next vertex
- play = infinite path through the game
- vertices labelled with natural numbers (priorities)
- winner of a play: determined by the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



TU/e Technische Universiteit Eindhoven University of Technology

# 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 $W_\Box$.



TU/e Technische Universiteit Eindhoven University of Technology

# 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 & Jurdziński '00
- (Deterministic) Subexponential . . . . . . . . . . . . . . . . . . . . Jurdziński, Paterson & Zwick '06
- Bigstep . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Schewe '07
- Priority promotion algorithms . . . . . . . . . . . . . . . Benerecetti, Dell'Erba & Mogavero '16
- Quasi-polynomial algorithm . . . . . . . . . . . Calude, Jain, Khoussainov, Li & Stephan '16



# Concepts



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

**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.



TU/e Technische Universiteit Eindhoven University of Technology

## Concepts



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

**Definition ($\diamond$-Dominion)**

$D \subseteq W_\diamond$ is a dominion of $\diamond$, if she has a memoryless strategy $\varrho$ that is:

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

Likewise for an $\Box$-dominion.



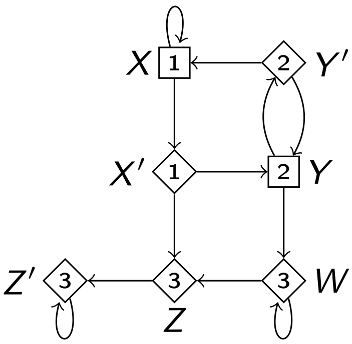
TU/e Technische Universiteit Eindhoven University of Technology

# Concepts



## Example (Dominions)

Consider parity game $G$:




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



# Concepts



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

## Definition (◇-Attractor sets)

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

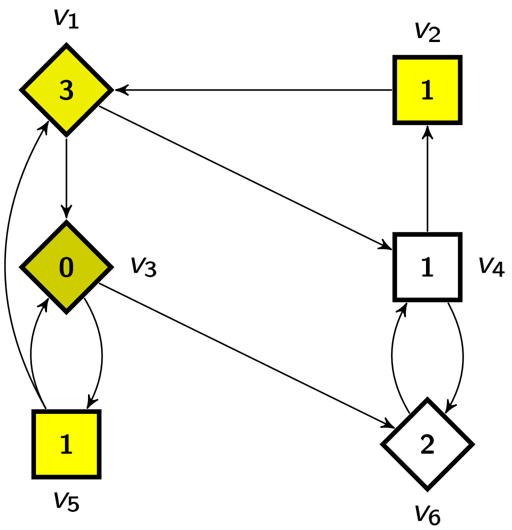
- containing $U$
- such that ◇ can force any play to reach $U$.

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

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

Likewise for an □-attractor.

### Example (Attractor sets)




$\Diamond$-$Attr(G, U)$: vertices from which $\Diamond$ 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: $\mathcal{O}(|V| + |E|)$; can be made to run in $\mathcal{O}(|E|)$





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

If $U$ is an $\Diamond$-dominion (dually for $\Box$-dominion) in $G$ then (by definition)

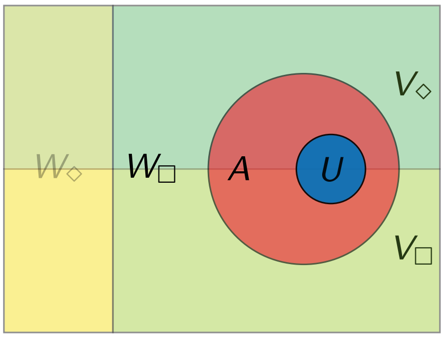
- there is a strategy $\varrho$ such that $\Diamond$ wins $U$
- $\Diamond$ can always choose to stay in $U$
- $\Box$ cannot leave $U$ (it is a trap)

...but also:

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

# Concepts

Visually:




- $U$ is a $\diamond$-dominion
- $A = \diamond\text{-}Attr(G, U)$
- $A$ is an $\diamond$-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

# Recursively 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 $W_\Box$

```
1:  m ← min{p(v) | v ∈ V}
2:  h ← max{p(v) | v ∈ V}
3:  if h = m or V = ∅ then
4:     if m is even or V = ∅ then
5:        return (V, ∅)
6:     else
7:        return (∅, V)
8:     end if
9:  end if
```

```
10: ○ ← ◇ if m is even and □ otherwise
11: U ← {v ∈ V | p(v) = m}
12: A ← ○-Attr(G, U)
13: (W'◇, W'□) ← Recursive(G \ A)
14: if W'_(○̄) = ∅ then
15:    W○ ← A ∪ W'○
16:    W_(○̄) ← ∅
17: else
18:    B ← ○̄-Attr(G, W'_(○̄))
19:    (W◇, W□) ← Recursive(G \ B)
20:    W_(○̄) ← W_(○̄) ∪ B
21: end if
22: return (W◇, W□)
```



# Observations



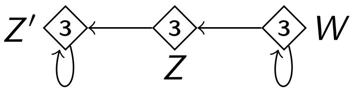
- Lines 1-9: base case, straightforward.
- Lines 10-13: try to establish a dominion. Two cases:
  - Lines 12-15: ($\bigcirc$ wins all): $\bigcirc$ wins in $G \setminus A$, then $\bigcirc$ wins all of $G$, since if $\overline{\bigcirc}$ visits $A$, then $\bigcirc$ plays towards $U$ using attractor, visiting $A$ 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** $(\emptyset, \{W, Z, Z'\})$

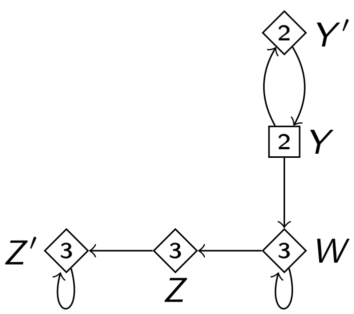





# Exercise



Apply the recursive algorithm to the following parity game $G$

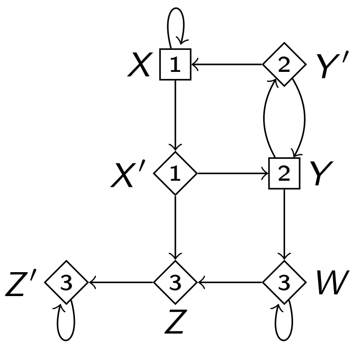



$m \leftarrow 2$
$h \leftarrow 3$
...
$\bigcirc \leftarrow \Diamond$
$U \leftarrow \{v \in V \mid p(v) = 2\} = \{Y, Y'\}$
$A \leftarrow$ -$Attr^{\Diamond}(G, U) = \{Y, Y'\}$
$(W'_{\Diamond}, W'_{\Box}) \leftarrow Recursive(G \setminus \{Y, Y'\}) = (\emptyset, \{Z, Z', W\})$
**if** $W'_{\Box} = \emptyset$ **then**
    ...
**else**
    $B \leftarrow$ -$Attr^{\Box}(G, W'_{\Box}) = \{Y, Y', Z, Z', W\}$
    $(W_{\Diamond}, W_{\Box}) \leftarrow Recursive(G \setminus B) = (\emptyset, \emptyset)$
    $W_{\Box} \leftarrow W_{\Box} \cup B = B = \{Y, Y', Z, Z', W\}$
**end if**
**return** $(W_{\Diamond}, W_{\Box}) = (\emptyset, \{Y, Y', Z, Z', W\})$

## Exercise



Consider parity game $G$:




$m \leftarrow 1$
$h \leftarrow 3$
...
$\bigcirc \leftarrow \square$
$U \leftarrow \{v \in V \mid p(v) = 1\} = \{X, X'\}$
$A \leftarrow$ -$Attr^{\square}(G, U) = \{X, X'\}$
$(W'_{\diamond}, W'_{\square}) \leftarrow Recursive(G \setminus \{X, X'\}) = (\emptyset, \{Y, Y', Z, Z', W\})$
**if** $W'_{\diamond} = \emptyset$ **then**
    $W_{\square} \leftarrow A \cup W'_{\square} = \{X, X', Y, Y', Z, Z', W\}$
    $W_{\diamond} \leftarrow \emptyset$
**else**
    ...
**end if**
**return** $(W_{\diamond}, W_{\square}) = (\emptyset, \{X, X', Y, Y', Z, Z', W\})$

So, player $\square$ 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

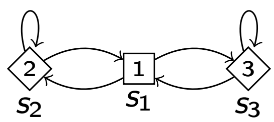




# Exercise



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.