Where are parity games used?

- Model Checking
- Equivalence Checking
- Satisfiability/Validity of modal logic
- Synthesis
Parity Games

$V$ A set of vertices.

$\rightarrow \subseteq V \times V$ An edge relation.

$\Omega : V \rightarrow \mathbb{N}$ A priority mapping.

$\Diamond, \Box$ Two players (even, odd).

$(V^{\Diamond}, V^{\Box})$ A partition of $V$.

\begin{align*}
0 & \xrightarrow{w} 0 \\
0 & \xrightarrow{s} 0 \\
0 & \xrightarrow{t} 0 \\
0 & \xrightarrow{u} 1 \\
0 & \xrightarrow{v} 1
\end{align*}

$\Diamond \equiv \text{even}$

$\Box \equiv \text{odd}$
Parity Games

\[ V \subseteq V \times V \] A set of vertices.

\[ \Omega : V \rightarrow \mathbb{N} \] An edge relation.

\[ \square, \Diamond \] A priority mapping.

\[ (V_\Diamond, V_\square) \] Two players (even, odd).

\[ (V_\Diamond, V_\square) \] A partition of \( V \).
Parity Games

- $V$ (a set of vertices)
- $\rightarrow \subseteq V \times V$ (an edge relation)
- $\Omega : V \rightarrow \mathbb{N}$ (a priority mapping)
- $\Diamond, \Box$ (two players, even, odd)
- $(V_\Diamond, V_\Box)$ (a partition of $V$)
Parity Games

- $V$: A set of vertices.
- $\rightarrow \subseteq V \times V$: An edge relation.
- $\Omega: V \rightarrow \mathbb{N}$: A priority mapping.
- $\Diamond, \Box$: Two players (even, odd).
- $(V^\Diamond, V^\Box)$: A partition of $V$. 

![Parity Game Diagram]

- $\Diamond = $ even
- $\Box = $ odd

Open Universiteit

www.ou.nl
Parity Games

- $V$ \quad A set of vertices.
- $\rightarrow \subseteq V \times V$ \quad An edge relation.
- $\Omega : V \rightarrow \mathbb{N}$ \quad A priority mapping.
- $\Diamond, \Box$ \quad Two players (even, odd).
- $(V_\Diamond, V_\Box)$ \quad A partition of $V$.

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]

\[\]
Parity Games

$V$ A set of vertices.
$\rightarrow \subseteq V \times V$ An edge relation.
$\Omega : V \rightarrow \mathbb{N}$ A priority mapping.
$\Diamond$, $\Box$ Two players (even, odd).
$(V_\Diamond, V_\Box)$ A partition of $V$.

$\Rightarrow$ Winner?
$\Rightarrow$ Optimal strategies?
Parity Games

\[ V \subseteq V \times V \] 
A set of vertices.

\[ \Omega : V \rightarrow \mathbb{N} \] 
An edge relation.

Two players (even, odd).

\[ (V_\Diamond, V_\Box) \] 
A priority mapping.

Winner?

\[ \Diamond = \text{even} \]
\[ \Box = \text{odd} \]
Parity Games

$V$ A set of vertices.
$ightarrow \subseteq V \times V$ An edge relation.
$\Omega : V \rightarrow \mathbb{N}$ A priority mapping.
$\Diamond, \Box$ Two players (even, odd).
$(V\Diamond, V\Box)$ A partition of $V$.

**Winner?**

**Optimal strategies?**
Parity Games

- $V$: A set of vertices.
- $\rightarrow \subseteq V \times V$: An edge relation.
- $\Omega: V \rightarrow \mathbb{N}$: A priority mapping.
- $\Diamond, \Box$: Two players (even, odd).
- $(V_{\Diamond}, V_{\Box})$: A partition of $V$.

Winner?

Optimal strategies?
Parity Games

\[ V \subseteq V \times V \]  
An edge relation.

\[ \Omega : V \rightarrow \mathbb{N} \]  
A priority mapping.

\[ \Diamond, \Box \]  
Two players (even, odd).

\( (V\Diamond, V\Box) \)  
A partition of \( V \).

\[ t \]

\[ w \]

\[ s \]

\[ u \]

\[ v \]

\[ \Diamond = \text{even} \]
\[ \Box = \text{odd} \]

▶ Winner?
▶ Optimal strategies?
Winning Parity Games

Memoryless determinacy

- Partition \((W_\Diamond, W_\Box)\) of \(V\)
- Player \(\bigcirc\) has memoryless winning strategy from \(W_\bigcirc\), for \(\bigcirc \in \{\Diamond, \Box\}\)
Solving Parity Games

Solving a parity game:

- Determine partition \((W_\Diamond, W_\Box)\)

Complexity:

- Problem is in \(NP \cap \text{co-} NP\)
- Is it in \(P\)?
Solving Parity Games

Solving a parity game:

▶ Determine partition \((W_\Diamond, W_\Box)\)

Complexity:

▶ Problem is in \(NP \cap co-NP\)
▶ Is it in \(P\)? Open!
Why benchmark?

Complexity + applications ⇒ active research

Algorithms for:
  ▶ solving
  ▶ simplifying
  ▶ reducing

parity games

How to compare new algorithms to existing ones?
Existing practice

- Only *theoretical* analysis (big-O)
- Class of games that meets *upper bound*
- Random games
- *(Very) small set of* games

Results from different papers *not comparable*
Requirements on Benchmarks

- Cover **broad** range of games:
  - Different **problems**
  - Different **structural properties**
- Games from the **literature**
Contributions

- Set of parity games
- List of structural properties
- Analysis of games w.r.t. these properties
Set of parity games

- Model checking:
  - Communication protocols (C)ABP, BRP, SWP
  - Cache coherence protocol
  - Two-player board games
  - Industrial IEEE-1394 link-layer, truck lift
  - Elevator, Hanoi towers

- Equivalence checking: strong-, weak-, branching bisimulation of communication protocols

- Validity/satisfiability of LTL, CTL, CTL*, PDL and \( \mu \)-calculus (using MLSolver)

- Random games (using PGSolver)

- Hard cases (using PGSolver)
Structural properties

- Some properties known to affect complexity of solving:
  - Number of vertices and edges ("size")
  - Number of priorities
  - Width measures (tree-width, DAG-width, etc.)
  - SCCs
- New: alternation depth (inspired by modal equation systems)
- And some more...
Alternation depth

- Describe complexity more accurately
- Similar to ideas in [Emerson & Lee 1986] for \(\mu\)-calculus

Three steps (let \(C \in \text{sccs}(G)\))

1. Nesting depth of \(v\) in \(C\) is \(#\text{alternations}\) between even and odd priorities on paths of descending priorities in \(C\)
2. Nesting depth of \(C\) is \(\max\{\text{nestingdepth}(v) \mid v \in C\}\)
3. Alternation depth of a parity game is the maximal nesting depth of its SCCs
Analysis of games w.r.t. structural properties

Vertices vs. edges

- Model checking
- Equivalence
- Mlsolver
- Special cases
- Random
Analysis of games w.r.t. structural properties

Alternation depth

- model checking
- equivalence
- mlsolver
- special cases
- random

![Graph showing alternation depth vs vertices](image)
Analysis of games w.r.t. structural properties

Diameter

Graph showing the relationship between Diameter and BFS Height on a log-log scale. Different markers represent different categories:
- x: model checking
- o: equivalence
- +: mlsolver
- □: special cases
- △: random

Diameter values range from 10^0 to 10^4, and BFS Height values range from 10^1 to 10^4.
Applications

- Used to assess parity game reductions in [Cranen, K & Willemse 2011, 2012]
- Subset of generation process used for benchmarks in [K, Wesselink & Willemse, 2014]
- Confirmed observation from [Friedmann & Lange 2009]: recursive algorithm beats sophisticated algorithms (unpublished)
Summary

I presented:

- A set of **parity games**
- Structural **properties** of parity games
- An **analysis** of the games w.r.t. these properties
Open issues

▶ Use structural properties to optimise/design algorithms
▶ Perform large-scale comparison of different algorithms
▶ Extend set of games with other encodings/more examples
▶ Design algorithms for computing more complex structural properties
Please contribute your own games!

jeroenkeiren.nl
github.com/jkeiren/paritygame-generator