# Deciding Parity Games in Quasipolynomial Time

**Cristian S. Calude (U. Auckland)**

**Sanjay Jain (Nat. U. of Singapore)**

**Bakhadyr Khoussainov (Auckland)**

**Wei Li (Nat. U. of Singapore)**

**Frank Stephan (Nat. U. of Singapore)**

# Infinite games played on Finite Graph

- Given a directed graph $(\mathbf{V}, \mathbf{E})$, where each node has at least one successor
- Start node $\mathbf{s}$, Anke starts the play
- A function $\mathbf{F}$ mapping subsets of $\mathbf{V}$ to the player $\mathbf{Anke}/\mathbf{Boris}$ who wins
- Players move alternately a marker through the graph along the edges of the graph forever
- Let $\mathbf{U}$ be the set of infinitely often visited nodes in the play
- The winner of the play is then given by $\mathbf{F}(\mathbf{U})$
- The winner of the game is one which has a winning strategy.

# Parity and Muller Games

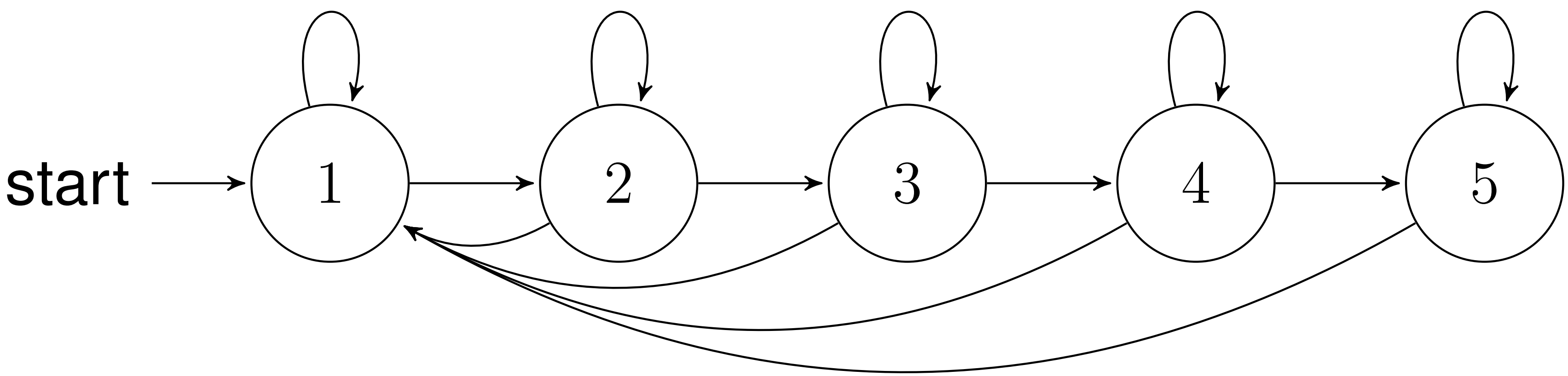
**Parity Game**: Each node has a value (natural number) and $\mathbf{F}(\mathbf{U})$ depends only on $\max\{\mathbf{val}(\mathbf{u}) : \mathbf{u} \in \mathbf{U}\}$

One can assume that $\mathbf{F}(\mathbf{U})$ depends on parity of $\max\{\mathbf{val}(\mathbf{u}) : \mathbf{u} \in \mathbf{U}\}$ being odd/even

**Coloured Muller game:** Each node has some colours and $\mathbf{F}$ does not depend on $\mathbf{U}$ directly but on $\bigcup_{\mathbf{u} \in \mathbf{U}} \mathbf{Colour}(\mathbf{u})$. This permits a more compact representation in the case that only few colours are used.

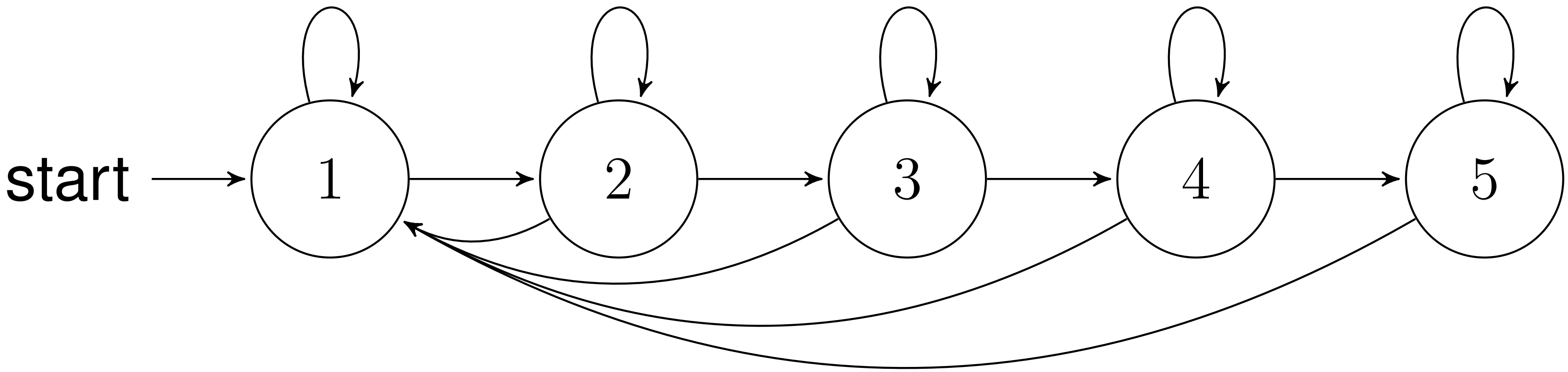
# Parity Games

A parity game $(\mathbf{V}, \mathbf{E}, \mathbf{s}, \mathbf{val})$ has a function $\mathbf{val} : \mathbf{V} \rightarrow \mathbb{N}$.
Anke wins a play $\mathbf{v_0}, \mathbf{v_1}, \mathbf{v_2}, \ldots$ iff $\mathbf{limsup\ val(v_k)}$ is odd.




Example of parity game, node $\mathbf{q}$ is labeled with $\mathbf{val(q)}$.
Play $\mathbf{1-2-3-4-1-2-3-4-1-2-3-4-1-2-\ldots}$
is won by Boris.

# Anke's Winning Strategy




| Node | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Anke's Move | 1 | 3 | 3 | 5 | 5 |

If $\mathbf{maxval}$ is odd and one can always go from $\mathbf{n}$ to $\mathbf{n}$ and to $\mathbf{min\{maxval, n+1\}}$ and to $\mathbf{1}$ then Anke has a winning strategy.

# Parity and Muller Games

**Observation**

A parity game with $\mathbf{n}$ nodes and values from $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{m}\}$ can be translated into an isomorphic Muller game with $\mathbf{n}$ nodes and $\mathbf{m}$ colours where node $\mathbf{u}$ has colour $\{\mathbf{val}(\mathbf{u})\}$. Anke wins play iff there is a $\mathbf{k}$ such that the union of colours contains $\mathbf{2k+1}$, but no greater colour.

# Parity vs Muller Games

**Theorem** [Björklund, Sandberg and Vorobyov 2003]
Every coloured Muller game with $\mathbf{n}$ nodes and $\mathbf{m}$ colours can be translated into parity game with $\mathbf{m! \cdot n}$ nodes and $\mathbf{2m}$ values and the same winner in time polynomial in the size of the target game.

**Theorem** [Hunter 2007]
A Muller game is also a parity game if $\mathbf{F}$ (for the Muller game) satisfies that whenever $\mathbf{F(U) = F(U')}$ then $\mathbf{F(U \cup U') = F(U)}$.

# Memoryless Strategies

- A **strategy** is a function which tells a player how to move after a certain sequence of moves has occurred;
- A strategy is called a **winning strategy** iff a player wins whenever following the strategy's advice;
- A strategy is **memoryless** if it only depends on the current position;
- A player who has a winning strategy is called **the winner of a game**.

# Memoryless Strategies

**Theorem** [Zielonka 1998]
Player Anke has a memoryless winning strategy in a Muller game $(V, E, s, F)$ if (a) she has a winning strategy and (b) for all $U, U' \subseteq V$ with $F(U) = \mathrm{Boris}$ and $F(U') = \mathrm{Boris}$ it holds that $F(U \cup U') = \mathrm{Boris}$.

**Corollary** [Allen and Jutla 1991, McNaughton 1993, Mostowski 1991]
The winner of a parity game can use a memoryless winning strategy.

# Complexity of Parity Game

- $\mathbf{NP} \cap \mathbf{coNP}$
- $\mathbf{UP} \cap \mathbf{coUP}$ (Jurdzinski 1998)
  Here $\mathbf{UP}$ is the class of all problems $\mathbf{L}$ which have unique certificates, that is, there exists a nondeterministic machine $\mathbf{N}$ accepting $\mathbf{L}$ such that for all $\mathbf{x} \in \mathbf{L}$, there is exactly one path of $\mathbf{N(x)}$ which accepts.
- Walukiewicz 2001: investigated relationship between games and model-checking
- Bernet, Janin and Walukiewicz 2002: investigated relationship between parity games and safety games

# Complexity of Parity Game

The following work provided algorithms to determine the winner of parity games; the complexity is measured by the number $\mathbf{n}$ of nodes and $\mathbf{m}$ of values.

- McNaughton 1993: $\mathbf{O((kn)^{m+1})}$ for some $\mathbf{k}$.
- Browne, Clarke, Jah, Long and Marrero 1997: $\mathbf{O(n^2 \cdot (2n/m)^{(m+3)/2})}$.
- Jurdinski, Patterson and Zwick 2006/2008: $\mathbf{n^{k \cdot \sqrt{n}}}$ for some $\mathbf{k}$.
- Schewe 2007/2016: $\mathbf{n^2 \cdot (k \cdot n \cdot m^{-2})^{m/3}}$ for some $\mathbf{k}$.

# Complexity of Parity Game

- Calude, Jain, Khoussainov, Li and Stephan 2017: $\mathbf{O(n^{\log(m)+6})}$; if $\mathbf{m} \leq \log(\mathbf{n})$ then $\mathbf{O(n^5)}$.
- Jurdziński and Lazić 2017 and Fearnly et. al 2017: Simultaneously quasi linear space and quasi polynomial Time Algorithm.
- **Open**: Can parity games be decided in polynomial time?

# Complexity of Parity Games

**Theorem**
There exists an alternating polylogarithmic space algorithm deciding which player has a winning strategy in a given parity game. When the game has $n$ nodes and the values of the nodes are in the set $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{m}\}$, then the algorithm runs in $\mathbf{O}(\log(\mathbf{n}) \cdot \log(\mathbf{m}))$ alternating space.

Thus, parity games can be solved in $\mathbf{O}(\mathbf{n}^{\mathbf{c}\log\mathbf{m}})$ time using Chandra, Kozen and Stockmeyer simulation of Alternating Turing Machines.

# Idea

- Consider the possible plays as infinite paths in a tree, where the root is the starting node, and children denote the next move
- At any node, the play so far is the path from the root to the node
- Try to determine win/loss at the nodes, and then use an alternating Turing machine
- At any node, wish to track if the play has gone through a loop, (same node with same player's move) with the highest valued node in the loop being of Anke (Boris)'s parity

# Idea

- The naive method is to archive the last 2n+1 nodes visited
- The naive method uses too much space
- Maintain only some partial info; however this information is enough to eventually determine a win for Anke (Boris) if she (he) has a memoryless winning strategy; this may delay detection of win i.e., require more depth than the naive method, but space usage is reduced
- The partial info is called winning statistics

# Winning Statistics

For Anke:

*i*-**sequence** is a sequence of $\mathbf{2^i}$ nodes (not necessarily consecutive, but in order) visited say $\mathbf{a_1, a_2, \ldots, a_{2^i}}$, in the play so far such that for each $\mathbf{k} \in \{\mathbf{1, 2, \ldots, 2^i - 1}\}$, the maximum value of the nodes visited between $\mathbf{a_k}$ and $\mathbf{a_{k+1}}$ (both inclusive) is of Anke's parity.

Aim of Anke: To find a sequence as above of length at least $\mathbf{2n+1}$, that is *i*-**sequence** with $\mathbf{2^i} \geq \mathbf{2n+1}$.

$\mathbf{i} = \lceil \log \mathbf{n} \rceil + \mathbf{2}$ suffices.

Such a sequence is built by combining smaller sequences over time in the winning statistics.

# Winning Statistics

**Winning statistics**:

- $(\mathbf{b_0}, \mathbf{b_1}, \ldots, \mathbf{b_{\lceil \log n \rceil + 2}})$, where each $\mathbf{b_i}$ is in $\{\mathbf{0}, \mathbf{1}, \ldots, \mathbf{m}\}$
- $\mathbf{b_i} > \mathbf{0}$ indicates that some $i$-**sequence** is being tracked
- $\mathbf{b_i} = \mathbf{0}$ indicates that no $i$-**sequence** is being tracked
- Note that, for any $\mathbf{i}$, there will be at most one $i$-**sequence** which is tracked.
- Property $\mathbf{b_i}$ (for $\mathbf{b_i} > \mathbf{0}$): an $i$-**sequence** is being tracked, and the largest value of a node visited at the end or after this $i$-**sequence** is $\mathbf{b_i}$.
- Winning statistics indicates a win (for Anke) if $\mathbf{b_{\lceil \log n \rceil + 2}} > \mathbf{0}$.

# Winning Statistics

- If a player plays a memoryless winning strategy then its winning statistics will eventually indicate a win (mature) while the opponent's winning statistics will never do so.
- The winning statistics can be kept small ($\mathbf{O}(\log \mathbf{n} \log \mathbf{m})$).
- The winning statistics permit to translate the parity game into a quasipolynomially sized reachability game where Anke has to reach a state where her winning statistics indicate a win; if she fails to do so, Boris wins.
- The reachability game can be solved in time linear in the number of its edges (well-known fact).
- Closer examination of special cases in order to obtain that parity games are fixed parameter tractable and to obtain furthermore some bounds for Muller games.

# Winning Statistics: Invariants

- Only $\mathbf{b_i}$ with $\mathbf{0 \leq i \leq \lceil \log(n) \rceil + 2}$ are considered and each such $\mathbf{b_i}$ is either zero or a value of a node which occurs in the play so far. Let $\mathbf{k = \lceil \log(n) \rceil + 2}$.
- An entry $\mathbf{b_i}$ refers to an $i$-**sequence** which occurred in the play so far iff $\mathbf{b_i > 0}$.
- If $\mathbf{b_i, b_j}$ are both non-zero and $\mathbf{i < j}$ then $\mathbf{b_i \leq b_j}$.
- If $\mathbf{b_i, b_j}$ are both non-zero and $\mathbf{i < j}$, then in the play of the game, the $i$-**sequence** starts only after a node with value $\mathbf{b_j}$ was visited at or after the end of the $j$-**sequence**.

# Update Algorithm for Winning Statistics

**Initialisation**: All $b_i$ are set to $0$.

**Update rule**: For each node visited with value $b$, choose the largest $i$ which satisfies one of the following:

- (1a) $b$ and $b_0, b_1, \ldots, b_{i-1}$ have Anke's parity and $b_i$ does not;
- (1b) $0 < b_i < b$.

If found, then let $b_i = b$ and $b_j = 0$ for all $j < i$, else no change.

**Winning Condition**: $b_{\lceil \log n \rceil + 2} > 0$.

# Example

Example for how $i$**-sequences** and $\mathbf{b_i}$s work.
The example play used here is not using memoryless strategy.
Consider a graph with nodes $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{7}\}$, with $\mathbf{value(v) = v}$, in which there is a directed edge from every node to every other node including itself.
The example play used is:

```
1 6 7 5 1 4 5 3 2
```

| Move | $b_4$,...,$b_0$ | $i$-sequences in play so far | rule |
|---|---|---|---|
| 1 | 0,0,0,0,1 | 0:1 | 1.(a) |
| 6 | 0,0,0,0,6 | 0:1 6 | 1.(b) |
| 7 | 0,0,0,0,7 | 1 6 0:7 | 1.(a) |
| 5 | 0,0,0,5,0 | 1 6 1:7 1:5 | 1.(a) |
| 1 | 0,0,0,5,1 | 1 6 1:7 1:5 0:1 | 1.(a) |
| 4 | 0,0,0,5,4 | 1 6 1:7 1:5 0:1 4 | 1.(b) |
| 5 | 0,0,0,5,5 | 1 6 1:7 1:5 1 4 0:5 | 1.(a) |
| 3 | 0,0,3,0,0 | 1 6 2:7 2:5 1 4 2:5 2:3 | 1.(a) |
| 2 | 0,0,3,0,0 | 1 6 2:7 2:5 1 4 2:5 2:3 2 | |

# Verification

Suppose Anke has a memoryless winning strategy.

**Claim**: If a player is declared a winner by the algorithm in a play, then the play contains a loop with its maximum valued node being a node of the player.

As $\mathbf{2^k > 2n}$, there exists a node $v$ which has appeared twice with the same player having the next move. The maximum valued node between these two occurences in the play has the value of X's parity, where X was declared a winner.

# Verification Continued

**Claim**: If the winner of the game plays a memoryless winning strategy then the opponent is never declared a winner.

Suppose Boris is declared a winner. Then by the earlier claim, there exists a loop with maximum valued node between these two occurences in the play having Boris' parity.
But then Boris could repeat this loop forever (as Anke is playing memoryless strategy) contradicting the assumption that Anke has a memoryless winning strategy.

# Verification Continued

**Claim**: If a player follows a memoryless winning strategy then it is eventually declared a winner.

Let $\mathbf{b_k(t)}$ denote the value of $\mathbf{b_k}$ at the end of $\mathbf{t}$-th step (after the updates for the $\mathbf{t}$-th move). Let

$$\mathbf{B_c(t) = \{i : b_i(t) \text{ has Anke's parity and } b_i(t) \geq c\}}$$

$$\mathbf{count(c,t) = \sum_{i \in B_c(t)} 2^i}$$

# Verification Continued

If at steps $\mathbf{t}, \mathbf{t}'$, with $\mathbf{t} < \mathbf{t}'$, a move to a node with value $\mathbf{c}$ are made and in between these times no moves to a node with value $\geq \mathbf{c}$ is made, then $\mathbf{count}(\mathbf{c}, \mathbf{t}) < \mathbf{count}(\mathbf{c}, \mathbf{t}')$.

To see this, let $\mathbf{i}$ be the largest index such that $\mathbf{b_i}$ is updated in some step $\mathbf{t}''$ with $\mathbf{t} < \mathbf{t}'' \leq \mathbf{t}'$. Thus, (i) $\mathbf{b_i}(\mathbf{t}) < \mathbf{c}$ or $\mathbf{b_i}(\mathbf{t})$ is of Boris' parity, and (ii) $\mathbf{0} < \mathbf{b_i}(\mathbf{t}'') \leq \mathbf{c}$. Hence, if not earlier, $\mathbf{b_i}(\mathbf{t}')$ would be made equal to $\mathbf{c}$.
Hence,

$$\mathbf{count}(\mathbf{c}, \mathbf{t}') - \mathbf{count}(\mathbf{c}, \mathbf{t}) \geq \mathbf{2^i} - \sum_{\mathbf{j} \in \mathbf{B_c}(\mathbf{t}) : \mathbf{j} < \mathbf{i}} \mathbf{2^j} \geq \mathbf{1}$$

# Verification Continued

If the lim sup value of the play is of Anke's parity, say $\mathbf{c}$, then, $\mathbf{count}(\mathbf{c}, \mathbf{t})$ will eventually keep going up, and thus imply $\mathbf{b}_{\lceil \log \mathbf{n} \rceil + 2} > \mathbf{0}$.

If Boris has a memoryless winning strategy, this cannot happen, as $\mathbf{b}_{\mathbf{k}} \geq \mathbf{1}$ indicates that the game goes through a loop with the maximum value being Anke's parity.

# Reachability Games

- A directed graph $(\mathbf{V}', \mathbf{E}')$.
- A set $\mathbf{T}$ of target nodes.
- A start node, and a starting player.
- Anke wins a play if the play goes through a node in $\mathbf{T}$
- If the play goes on forever, then Boris wins.

# Reachability Game: Linear Time

**Theorem** [Well-Known]
The winner of a reachability game on a graph $(V, E)$ with set $T$ to be reached from $s$ can be found in time $O(|V| + |E|)$.

**Assumption:** The node says whether Anke or Boris moves or node is in $T$; the algorithm marks winning nodes for Anke.

**Algorithm**
For all $v \in T$ and all $v \in V - T$ where Anke moves let $q_v = 1$; for all $v \in V - T$ where Boris moves let $q_v$ be the number of successors.

Call $A(v)$ below for all $v \in T$.

$A(v)$: If $q_v = 0$ then return with no activity;
If $q_v = 1$ then let $q_v = 0$ and call $A(w)$ for all $w$ with $v$ being a successor of $w$ and return;
If $q_v > 1$ then let $q_v = q_v - 1$ and return.

Anke wins iff $q_s = 0$ after running the algorithm.

# Reduction to Reachability Games

Parity games can be reduced to reachability game as follows. Reachability game consists of:

- All nodes $(\mathbf{v}, \mathbf{p}, \mathbf{w})$ with node $\mathbf{v}$ of parity game, player $\mathbf{p}$ to move and current winning statistics $\mathbf{w}$ of Anke.
- Move from $(\mathbf{v}, \mathbf{p}, \mathbf{w})$ to $(\mathbf{v}', \mathbf{p}', \mathbf{w}')$ is possible iff there is an edge from $\mathbf{v}$ to $\mathbf{v}'$ in parity game, $\mathbf{p} \neq \mathbf{p}'$ and on move to $\mathbf{v}'$, winning statistics $\mathbf{w}$ of Anke are updated to $\mathbf{w}'$ and $\mathbf{w}$ is not already won for Anke.
- $\mathbf{T}$ consists of all the nodes in which the winnings statistics of Anke show a win for Anke.
- Starting node can be chosen appropriately based on starting node of the parity game.

# Main result

The winning statistics can be written down in $\lceil \log(\mathbf{n}) + \mathbf{3} \rceil$ binary numbers of $\lceil \log(\mathbf{m}) + \mathbf{1} \rceil$ bits,
the player to move needs one bit and
the position needs $\lceil \log(\mathbf{n}) \rceil$ bits.

This gives an overall number of nodes in the reachability game being bounded by $\mathbf{O}(\mathbf{n}^{\log(\mathbf{m})+\mathbf{5}})$ nodes with each node having up to $\mathbf{n}$ outgoing edges.
Thus, the overall size of the reachability graph is at most $\mathbf{O}(\mathbf{n}^{\log \mathbf{m}+\mathbf{6}})$.

**Theorem**
Parity games can be solved in time $\mathbf{O}(\mathbf{n}^{\log \mathbf{m}+\mathbf{6}})$.

# Fixed Parameter Tractability

A problem with parameters $n, m, \ldots$ is fixed parameter tractable in $m$, if its complexity can be expressed in the form $O(g(m) + poly(n, \ldots))$, where $g(m)$ does not depend on $n, \ldots$, and the degree of $n, \ldots$ do not depend on $m$.

# Fixed Parameter Tractability

**Theorem**
If $\mathbf{m} \leq \log(\mathbf{n})$ then the winner of a parity game with $\mathbf{n}$ nodes and $\mathbf{m}$ values can be found in time $\mathbf{O}(\mathbf{n}^5)$.

We need to count the number of possible values of $\mathbf{b_i}$'s.
Let $\mathbf{b'_0} = \mathbf{max}(\mathbf{b_0}, \mathbf{1})$; $\mathbf{b'_{i+1}} = \mathbf{max}(\mathbf{b'_i}, \mathbf{b_{i+1}})$.
Let $\mathbf{b''_i} = \mathbf{b'_i} + \mathbf{i}$.
Now, $\mathbf{b''_i}$ are strictly increasing functions from $\{\mathbf{0}, \mathbf{1}, \ldots, \lceil \log \mathbf{n} \rceil + \mathbf{2}\}$ to $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{2}\lceil \log \mathbf{n} \rceil + \mathbf{2}\}$, and thus there are at most $\mathbf{O}(\mathbf{n}^2)$ many possibilities for them.
$\mathbf{b'_i}$ can be recovered from $\mathbf{b''_i}$, and then $\mathbf{b_i}$ can be recovered by knowing which of the $\mathbf{b_i}$'s are $\mathbf{0}$.

Thus, the overall number of possibilities for $\mathbf{b_i}$ is bounded by $\mathbf{O(n^3)}$.
This gives that in the corresponding reachability game, there are at most $\mathbf{O(n^4)}$ nodes and at most $\mathbf{O(n^5)}$ edges.

**Corollary**
Parity games are fixed parameter tractable when parameterised by the number of values.

**Remark**
For constant $\mathbf{m}$, the number of possible values of the winning statistics can be given by $\mathbf{O(n^2)}$ and the number of nodes in the reachability game by $\mathbf{O(n^3)}$ and the overall running time is $\mathbf{O(n^4)}$. Furthermore, in the case of graphs of constant out-degree, say out-degree $\mathbf{2}$, the overall running time is $\mathbf{O(n^3)}$.

# More careful bounds

The winning statistics of $\lceil \log(\mathbf{n}) + \mathbf{3} \rceil$ numbers $\mathbf{b_0, b_1, \ldots, b_k}$ from $\mathbf{0, \ldots, m}$ can be coded by $\mathbf{\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_k}$ with

- $\mathbf{\hat{b}_0 = b_0}$
- if $\mathbf{b_{i+1} = 0}$ then $\mathbf{\hat{b}_{i+1} = \hat{b}_i + 1}$ else $\mathbf{\hat{b}_{i+1} = \hat{b}_i + 2 + \max\{b_{i+1} - b_j : j \leq i\}}$.

So $\mathbf{\hat{b}_k \leq b_k + 2 \cdot k}$.

For $\mathbf{h} = \lceil \mathbf{m} / \log(\mathbf{n}) \rceil$, one has that the $\hat{\mathbf{b}}_\mathbf{0}, \ldots, \hat{\mathbf{b}}_\mathbf{k}$ select $\lceil \log(\mathbf{n}) + \mathbf{3} \rceil$ numbers out of $\lceil \log(\mathbf{n}) + \mathbf{3} \rceil \cdot (\mathbf{h} + \mathbf{2})$.
Number of ways to do this, for all $\mathbf{h} \in \mathbb{N}$, can be given by $\mathbf{O}(\mathbf{h}^\mathbf{4} \cdot \mathbf{n}^{\mathbf{1.45} + \log(\mathbf{h}+\mathbf{2})})$.

So the number of nodes in the reachability game is $\mathbf{O}(\mathbf{h}^\mathbf{4} \cdot \mathbf{n}^{\mathbf{2.45} + \log(\mathbf{h}+\mathbf{2})})$ and it has $\mathbf{O}(\mathbf{h}^\mathbf{4} \cdot \mathbf{n}^{\mathbf{3.45} + \log(\mathbf{h}+\mathbf{2})})$ edges.

Time $\mathbf{O}(\lceil \mathbf{m} / \log(\mathbf{n}) \rceil^\mathbf{4} \cdot \mathbf{n}^{\mathbf{3.45} + \log(\lceil \mathbf{m} / \log(\mathbf{n}) \rceil + \mathbf{2})})$ is sufficient to determine the winner of a parity game.

# Bounds for Coloured Muller Games

The reduction of coloured Muller games to parity games by Björklund, Sandberg and Vorobyov from 2003 gives the following application; note that for almost all $(m, n)$, $2m \leq \log(m! \cdot n)$.

**Theorem**
A Muller game with $m$ colours and $n$ nodes can be solved in time $O((m^m \cdot n)^5)$.

**Theorem** [Björklund, Sandberg and Vorobyov 2003]
Parity games are FPT iff Muller games are FPT.

# Bounds for Coloured Muller Games

However, there is a limitation.

**Theorem**
If $\mathbf{W[1] \neq FPT}$ then coloured Muller games with $\mathbf{m}$ colours and $\mathbf{n}$ nodes are not solvable in $\mathbf{2^{o(m \cdot \log(m))} \cdot n^{O(1)}}$ time.

**Theorem**
Solving memoryless coloured Muller games with four colours is NP-hard.

# Multi-Dimensional Parity Games

**Definition** A $\mathbf{k}$-dimensional parity game has for every node a $\mathbf{k}$-dimensional vector of values from $\{\mathbf{1}, \mathbf{2}, \ldots, \mathbf{m}\}$. A play is win for Anke iff for all coordinates $\tilde{\mathbf{k}}$, the limit superior of the $\tilde{\mathbf{k}}$-th value of the vectors is an odd number; a play is a win for Boris iff for some coordinate the limit superior of the values in that coordinate is even. Note that $\mathbf{k}, \mathbf{m} \geq \mathbf{2}$.

**Theorem**
The $\mathbf{k}$-dimensional parity games with $\mathbf{m}$ values per dimension and $\mathbf{n}$ nodes can be solved in time $\mathbf{O}((\mathbf{2}^{\mathbf{k} \cdot \log(\mathbf{k}) \cdot \mathbf{m}} \cdot \mathbf{n})^{\mathbf{5.45}})$.

**Theorem**
The $\mathbf{k}$-dimensional parity games with $\mathbf{m}$ values per dimension and $\mathbf{n}$ nodes cannot be solved in time $\mathbf{2^{o(k \cdot \log(k) \cdot m)} \cdot n^{O(1)}}$ unless $\mathbf{W[1] = FPT}$; even if $\mathbf{k}$ is fixed to a constant.

# Summary

The winner of a parity game can be found in quasipolynomial time where the exponent depends only logarithmically on the number of values used in the parity game.

Parity games are fixed parameter tractable in the case that the number of values is fixed. If $\mathbf{m} \leq \log(\mathbf{n})$ then the parity game can be solved in $\mathbf{O}(\mathbf{n}^5)$ and one can give a general formula of the type $\mathbf{O}(\mathbf{2}^{\mathbf{m}} \cdot \mathbf{n}^4)$ which works for all $\mathbf{n}$ and $\mathbf{m}$ [Krishnendu Chatterjee].

The bounds transfer to Muller games and show that these can be decided in $\mathbf{O}((\mathbf{m}^{\mathbf{m}} \cdot \mathbf{n})^5)$. The bound cannot be improved to $\mathbf{2}^{\mathbf{o}(\mathbf{m} \cdot \log(\mathbf{m}))} \cdot \mathbf{n}^{\mathbf{O}(1)}$ unless $\mathbf{FPT} = \mathbf{W}[1]$.