Algorithms for Model Checking (2IW55)

Lecture 13

The Small Progress Measures algorithm for Parity games

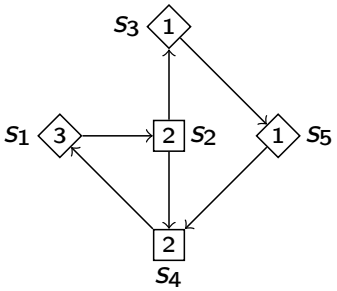
Background material:

"Small Progress Measures for Solving Parity Games", Marcin Jurdzinski

December 20, 2011

Parity games (recap)

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



Recursive algorithm (recap)

Goal: compute winning sets

Relevant concepts:

- 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 set

Small progress measures (intuition)

- Characterise cycles reachable from each vertex
- Cycles can be used to decide winner
- Assign measure to each vertex counting, for odd priorities, maximal number of times the priority can be seen if $\Box$ moves over the graph, until a vertex with smaller priority is seen
- Measure computed using fixed point iteration

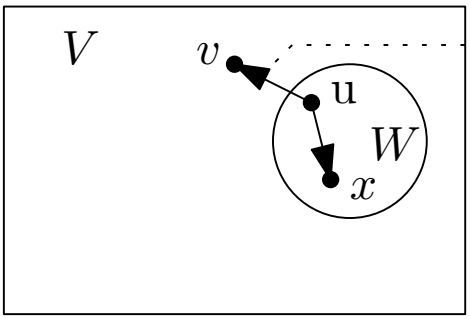
Closedness

Let $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game. A strategy $\psi_\Diamond : V_\Diamond \to V$ is closed on a set $W \subseteq V$ if for all $v \in W$, we have:

- if $v \in V_\Diamond$ then $\psi_\Diamond(v) \in W$, and
- if $v \in V_\Box$ then $(v, w) \in E$ implies $w \in W$.

Each play consistent with strategy $\psi_\Diamond$ closed on W, starting in W, stays within W

Illustration of closedness



Edges (u, v) only for $u \in V_{Even}$, and only if there also is edge (u, x) for $x \in W$.

Cycles

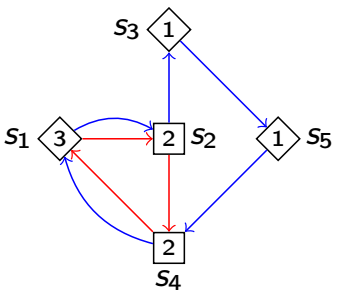
A cycle in a parity game is a path $v_1, v_2, \ldots, v_n$, with $v_1 = v_n$

We say that a cycle $v_1, v_2, \ldots, v_n$ is

- an i-cycle if $i = \min\{p(v_j) \mid 1 \leq j \leq n\}$ (i is the smallest priority occurring on the cycle)
- an even cycle if it is an i-cycle, and i is even

Cycle (example)

Two simple cycles in the following graph, indicated in different colors.



Cycles vs winning strategies

Let:

- $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game
- $\psi_\Diamond$ be a strategy for player $\Diamond$, closed on $W \subseteq V$

Define $G^W_{\psi_\Diamond} = (V', E, p, (V_\Diamond, V_\Box))$

- $V' = W$
- $V'_\Diamond = V_\Diamond \cap W$
- $V'_\Box = V_\Box \cap W$
- $E' = \{(v, w) \in E \mid v \in V'_\Diamond \wedge w = \psi_\Diamond(v)\} \cup \{(v, w) \in E \mid v \in V'_\Box\}$
- $p'(v) = p(v)$ for $v \in W$

$\psi_\Diamond$ is winning for player $\Diamond$ from W if and only if all cycles in $G^W_{\psi_\Diamond}$ are even

Small progress measures (intuition)

- Characterise cycles reachable from each vertex
- Cycles can be used to decide winner
- Assign measure to each vertex counting, for odd priorities, maximal number of times the priority can be seen if $\Box$ moves over the graph, until a vertex with smaller priority is seen
- Measure computed using fixed point iteration

d-tuples

Let $\alpha \in \mathbb{N}^d$ be a d-tuple of natural numbers

- we number its components from 0 to $d-1$, i.e. $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{d-1})$,
- $<, \leq, =, \neq, \geq, >$ on tuples denote lexicographic ordering,
- $(n_0, n_1, \ldots, n_k) \equiv_i (m_0, m_1, \ldots, m_l)$ iff $(n_0, n_1, \ldots, n_i) \equiv (m_0, m_1, \ldots, m_i)$, for $\equiv \in \{<, \leq, =, \neq, \geq, >\}$
- Note that if $i > k$ or $i > l$, the tuples will be suffixed with 0s

d-tuples (example)

- $(0,1,0,1) =_0 (0,2,0,1) \equiv (0) = (0) \equiv \text{true}$
- $(0,1,0,1) <_1 (0,2,0,1) \equiv (0,1) < (0,2) \equiv \text{true}$
- $(0,1,0,1) \geq_3 (0,2,0,1) \equiv (0,1,0,1) \geq (0,2,0,1) \equiv \text{false}$

Restricted d-tuples

Let $G = (V, E, p, (V_\diamond, V_\Box))$ be a parity game, and let $d = \max\{p(v) \mid v \in V\} + 1$.

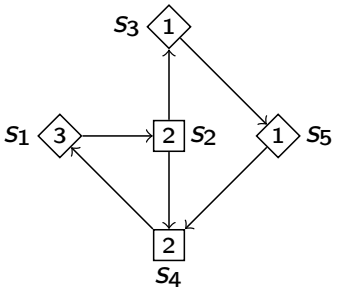
- For $i \in \mathbb{N}$, let $V_i = \{v \in V \mid p(v) = i\}$,
- Denote $n_i = \mid V_i \mid$, the number of vertices with priority i,

Define $\mathbb{M}_G \subseteq \mathbb{N}^d$, such that it is the finite set of d-tuples, with:

- 0 on even positions
- Natural numbers $\leq n_i$ on odd positions i

$\mathbb{M}_G$ (example)

Determine maximum value of $\mathbb{M}_G$ for the following parity game:



Maximum value of $\mathbb{M}_G$ is $(0, 2, 0, 1)$

$\mathbb{M}_G = \{0\} \times \{0, 1, 2\} \times \{0\} \times \{0, 1\}$

Parity progress measure

Recall: $\psi_\diamond$ is winning for player $\diamond$ from W if and only if all cycles in $G^W_{\psi_\diamond}$ are even

Idea: characterise vertices that can only reach even cycles.

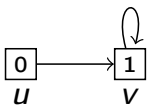
Definition (Parity progress measure)

Let $G = (V, E, p, (V_\diamond, V_\Box))$ be a parity game. Function $\varrho : V \to \mathbb{N}^d$ is a parity progress measure for G if for all $(v, w) \in E$ it holds that:

- $\varrho(v) \geq_{p(v)} \varrho(w)$ if $p(v)$ is even
- $\varrho(v) >_{p(v)} \varrho(w)$ if $p(v)$ is odd

There exists a parity progress measure for G iff all cycles in G are even

Parity progress measure (problem)



Problem: no parity progress measure can be assigned to these vertices, as parity progress measure only exists for even cycles. (Second clause requires $\varrho(v) >_1 \varrho(v)$)

Game parity progress measures (or, allowing odd cycles)

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

Define $\mathbb{M}_G^\top = \mathbb{M}_G \cup \{\top\}$, such that:

- $m\{<, <_i\}\top$ for all $m \in \mathbb{M}_G$, and $m\{\neq, \neq_i\}\top$
- $\top =_i \top$ for all i.

Definition (Prog)

If $\varrho{:}V \to \mathbb{M}_G^\top$ and $(v, w) \in E$, then $\mathit{Prog}(\varrho, v, w)$ is the least $m \in \mathbb{M}_G^\top$, such that

- $m \geq_{p(v)} \varrho(w)$ if $p(v)$ is even,
- $m >_{p(v)} \varrho(w)$, or $m = \varrho(w) = \top$ if $p(v)$ is odd.

Prog (examples)

Let $\mathbb{M}_G = \{0\} \times \{0, 1, 2\} \times \{0\} \times \{0, 1\}$

Suppose $p(v) = 0$, $\varrho(w) = (0, 2, 0, 0)$. Then $\mathit{Prog}(\varrho, v, w) = (0, 0, 0, 0)$

Suppose $p(v) = 1$, $\varrho(w) = (0, 2, 0, 0)$. Then $\mathit{Prog}(\varrho, v, w) = \top$

Suppose $p(v) = 3$, $\varrho(w) = (0, 2, 0, 0)$. Then $\mathit{Prog}(\varrho, v, w) = (0, 2, 0, 1)]$

Game parity progress measure

Definition (Game parity progress measure)

Let $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game. A function $\varrho : V \to \mathbb{M}_G^\top$ is a game parity progress measure if for all $v \in V$, it holds that:

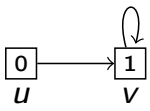
- if $v \in V_\Diamond$, then $\exists_{(v,w)\in E}\, \varrho(v) \geq_{p(v)} \mathit{Prog}(\varrho, v, w)$
- if $v \in V_\Box$, then $\forall_{(v,w)\in E}\, \varrho(v) \geq_{p(v)} \mathit{Prog}(\varrho, v, w)$

ϱ is least game parity progress measure

$\Longrightarrow$

$\varrho(v) \neq \top$ iff all cycles reachable from vertex v are even.

Example



Observe: $\varrho(u) = \varrho(v) = \top$

Measure can identify both even and odd reachable cycles.

Small progress measures (intuition)

- Characterise cycles reachable from each vertex
- Cycles can be used to decide winner
- Assign measure to each vertex counting, for odd priorities, maximal number of times the priority can be seen if $\Box$ moves over the graph, until a vertex with smaller priority is seen
- Measure computed using fixed point iteration

Fixed points

Characterise game parity progress measure as fixed point of monotone operators in a finite complete lattice:

- a least game parity progress measure φ exists (Knaster-Tarski),
- computable by fixed point iteration (see Lecture 3, slide 13 for an algorithm),

Let $G = (V, E, p, (V_\Diamond, V_\Box))$, and $\varphi, \varrho : V \to \mathbb{M}_G^\top$.

- $\varphi \sqsubseteq \varrho$ if $\varphi(v) \leq \varrho(v)$ for all $v \in V$
- write $\varphi \sqsubset \varrho$ if $\varphi \sqsubseteq \varrho$ and $\varphi \neq \varrho$.

$\sqsubseteq$ gives a complete lattice structure on the set of functions $V \to \mathbb{M}_G^\top$.

Lifting progress measures

Define $Lift_v(\varrho)$ for $v \in V$ as follows:

$$Lift_v(\varrho) = \begin{cases} \varrho[v := \min\{Prog(\varrho, v, w) \mid (v, w) \in E\}] & \text{if } v \in V_\Diamond \\ \varrho[v := \max\{Prog(\varrho, v, w) \mid (v, w) \in E\}] & \text{if } v \in V_\Box \end{cases}$$

Observe:

- For every $v \in V$, $Lift_v$ is $\sqsubseteq$-monotone.
- A function $\varrho : V \to \mathbb{M}_G^\top$ is a game parity progress measure if and only if $Lift_v(\varrho) \sqsubseteq \varrho$ for all $v \in V$.

The algorithm

Compute least game parity progress measure using fixed point approximation:

Algorithm *(ProgressMeasureLifting)*

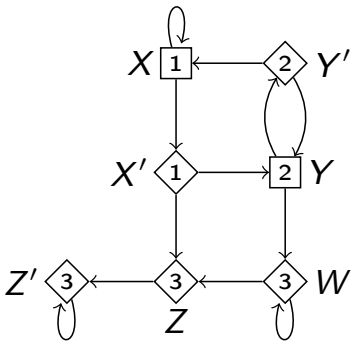
$\varrho \leftarrow \lambda v \in V.(0, \ldots, 0)$
while $\varrho \sqsubset Lift_v(\varrho)$ for some $v \in V$ **do**
  $\varrho \leftarrow Lift_v(\varrho)$
end while

Post condition:

- ϱ is least game parity progress measure
- $\{v \in V \mid \varrho(v) \neq \top\}$ is winning set for player $\Diamond$

Small progress measures (example)

Consider parity game G:



Maximum value of $\mathbb{M}_G$ is $(0, 2, 0, 3)$

Small progress measures (example) (1)

Initially: $\varrho \leftarrow \lambda v \in V.(0, 0, 0, 0)$, **so**

v	$\varrho(v)$
X	$(0, 0, 0, 0)$
X'	$(0, 0, 0, 0)$
Y	$(0, 0, 0, 0)$
Y'	$(0, 0, 0, 0)$
Z	$(0, 0, 0, 0)$
Z'	$(0, 0, 0, 0)$
W	$(0, 0, 0, 0)$

Step 2: $\varrho \leftarrow \mathit{Lift}_X(\varrho) = \varrho[X := \max\{\mathit{Prog}(\varrho, X, X'), \mathit{Prog}(\varrho, X, X)\}] = \varrho[X := \max\{(0, 1, 0, 0), (0, 1, 0, 0)\}] = \varrho[X := (0, 1, 0, 0)]$

v	$\varrho(v)$
X	$(0, 1, 0, 0)$
X'	$(0, 0, 0, 0)$
Y	$(0, 0, 0, 0)$
Y'	$(0, 0, 0, 0)$
Z	$(0, 0, 0, 0)$
Z'	$(0, 0, 0, 0)$
W	$(0, 0, 0, 0)$

Small progress measures (example) (2)

Step 3: $\varrho \leftarrow \mathit{Lift}_{\boldsymbol{X}}(\varrho) = \varrho[X := \max\{\mathit{Prog}(\varrho, X, X'), \mathit{Prog}(\varrho, X, X)\}] = \varrho[X := \max\{(0,1,0,0),(0,2,0,0)\}] = \varrho[X := (0,2,0,0)]$

v	$\varrho(v)$
X	$(0,2,0,0)$
X'	$(0,0,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,0)$
W	$(0,0,0,0)$

Step 4: $\varrho \leftarrow \mathit{Lift}_{\boldsymbol{X}}(\varrho) = \varrho[X := \max\{\mathit{Prog}(\varrho, X, X'), \mathit{Prog}(\varrho, X, X)\}] = \varrho[X := \max\{(0,1,0,0), \top\}] = \varrho[X := \top]$

v	$\varrho(v)$
X	$\top$
X'	$(0,0,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,0)$
W	$(0,0,0,0)$

Small progress measures (example) (3)

Step
5: $\mathit{Lift}_{\boldsymbol{Y'}}(\varrho) = \varrho[Y' := \min\{\mathit{Prog}(\varrho, Y', X), \mathit{Prog}(\varrho, Y', Y)\}] = \varrho[Y' := \min\{\top, (0,0,0,0)\}] = \varrho[Y' := (0,0,0,0)]$
$\mathit{Lift}_{\boldsymbol{Y}}(\varrho) = \varrho[Y := \max\{\mathit{Prog}(\varrho, Y, W), \mathit{Prog}(\varrho, Y, Y')\}] = \varrho[Y := \max\{(0,0,0,0),(0,0,0,0)\}] = \varrho[Y := (0,0,0,0)]$
$\varrho \leftarrow \mathit{Lift}_{\boldsymbol{X'}}(\varrho) = \varrho[X' := \min\{\mathit{Prog}(\varrho, X', Y), \mathit{Prog}(\varrho, X', Z)\}] = \varrho[X' := \min\{(0,1,0,0),(0,1,0,0)\}] = \varrho[X' := (0,1,0,0)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,0)$
W	$(0,0,0,0)$

Step 6: $\varrho \leftarrow \mathit{Lift}_{\boldsymbol{Z'}}(\varrho) = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{(0,0,0,1)\}] = \varrho[Z' := (0,0,0,1)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,1)$
W	$(0,0,0,0)$

Small progress measures (example) (4)

Step 7: $\varrho \leftarrow \mathit{Lift}_{Z'}(\varrho) = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{(0,0,0,2)\}] = \varrho[Z' := (0,0,0,2)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,2)$
W	$(0,0,0,0)$

Step 8: $\varrho \leftarrow \mathit{Lift}_{Z'}(\varrho) = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{(0,0,0,3)\}] = \varrho[Z' := (0,0,0,3)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,0,0,3)$
W	$(0,0,0,0)$

Small progress measures (example) (5)

Step 9: $\varrho \leftarrow \mathit{Lift}(\varrho, Z') = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{(0,1,0,0)\}] = \varrho[Z' := (0,1,0,0)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,1,0,0)$
W	$(0,0,0,0)$

Step 10: $\varrho \leftarrow \mathit{Lift}_{Z'}(\varrho) = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{(0,1,0,1)\}] = \varrho[Z' := (0,1,0,1)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$(0,1,0,1)$
W	$(0,0,0,0)$

Small progress measures (example) (6)

Step 11*: Repeat lifting Z' even more often
$\varrho \leftarrow \mathit{Lift}_{Z'}(\varrho) = \varrho[Z' := \min\{\mathit{Prog}(\varrho, Z', Z')\}] = \varrho[Z' := \min\{\top\}] = \varrho[Z' := \top]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$(0,0,0,0)$
Z'	$\top$
W	$(0,0,0,0)$

Step 12: $\varrho \leftarrow \mathit{Lift}_{Z}(\varrho) = \varrho[Z := \min\{\mathit{Prog}(\varrho, Z, Z')\}] = \varrho[Z := \min\{\top\}] = \varrho[Z := \top]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$\top$
Z'	$\top$
W	$(0,0,0,0)$

Small progress measures (example) (7)

Step 13:
$\varrho \leftarrow \mathit{Lift}_{W}(\varrho) = \varrho[W := \min\{\mathit{Prog}(\varrho, W, Z), \mathit{Prog}(\varrho, W, W')\}] = \varrho[W := \min\{\top, (0,0,0,1)\}] = \varrho[W := (0,0,0,1)]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$\top$
Z'	$\top$
W	$(0,0,0,1)$

Step 14*: Repeat lifting of W often
$\varrho \leftarrow \mathit{Lift}_{W}(\varrho) = \varrho[W := \min\{\mathit{Prog}(\varrho, W, Z), \mathit{Prog}(\varrho, W, W')\}] = \varrho[W := \min\{\top, \top\}] = \varrho[W := \top]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$(0,0,0,0)$
Y'	$(0,0,0,0)$
Z	$\top$
Z'	$\top$
W	$\top$

Small progress measures (example) (8)

Step 15: $\varrho \leftarrow \mathit{Lift}_{Y}(\varrho, Y) = \varrho[Y := \max\{\mathit{Prog}(\varrho, Y, W), \mathit{Prog}(\varrho, Y, Y')\}] = \varrho[Y := \max\{\top, (0,0,0,0)\}] = \varrho[Y := \top]$

v	$\varrho(v)$
X	$\top$
X'	$(0,1,0,0)$
Y	$\top$
Y'	$(0,0,0,0)$
Z	$\top$
Z'	$\top$
W	$\top$

Step 16: $\varrho \leftarrow \mathit{Lift}_{X'}(\varrho) = \varrho[X' := \min\{\mathit{Prog}(\varrho, X', Z), \mathit{Prog}(\varrho, X', Y)\}] = \varrho[X' := \min\{\top, \top\}] = \varrho[X' := \top]$

v	$\varrho(v)$
X	$\top$
X'	$\top$
Y	$\top$
Y'	$(0,0,0,0)$
Z	$\top$
Z'	$\top$
W	$\top$

Small progress measures (example) (9)

Step 17: $\varrho \leftarrow \mathit{Lift}_{Y'}(\varrho) = \varrho[Y' := \min\{\mathit{Prog}(\varrho, Y', X), \mathit{Prog}(\varrho, Y', Y)\}] = \varrho[Y' := \min\{\top, \top\}] = \varrho[Y' := \top]$

v	$\varrho(v)$
X	$\top$
X'	$\top$
Y	$\top$
Y'	$\top$
Z	$\top$
Z'	$\top$
W	$\top$

ϱ is least game parity progress measure, and $\{v \in V \mid \varrho(v) \neq \top\} = \emptyset$ is winning set for player $\diamond$. Hence player $\Box$ wins from all vertices

Strategies from progress measures

Let $G = (V, E, p, (V_\Diamond, V_\Box))$ be a parity game, and $\varrho : V \to \mathbb{M}_G^\top$ be least game parity progress measure.

- Define strategy $\overline{\varrho} : V_\Diamond \to V$ for player $\Diamond$, by setting $\overline{\varrho}(v)$ to be a successor w of $v \in V_\Diamond$ that minimises $\varrho(w)$.
- $\overline{\varrho}$ is a winning strategy for player $\Diamond$ from $\{v \in V \mid \varrho(v) \neq \top\}$.

Strategy (example)

As the winning set for player $\Diamond$ is empty, the strategy for player $\Diamond$ can be chosen arbitrarily

Stategy for player $\Box$ cannot be inferred directly (winning set *can* be determined), some tricks have to be applied...

Complexity

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

Worst-case running time complexity:

$$\mathcal{O}(de \cdot (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$$

Lowerbound on worst-case:

$$\Omega((\lceil n/d \rceil)^{\lceil d/2 \rceil})$$

Complexity

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

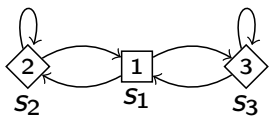
- Algorithm with best known upper bound: Big step algorithm due to Schewe, with complexity
$$\mathcal{O}(d \cdot n^{d/3})$$
- Also: best known upper bound for μ-calculus model checking
- Big step combines recursive algorithm with small progress measures;

Summary

- Parity games
- Relation to Boolean Equation Systems
- Link to model checking
- Simplification techniques (self-loop elim. priority compaction/propagation, bisimulation)
- Solving:
  - Recursive algorithm
  - Small progress measures

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 small progress measures algorithm.
- Compare the solution with the solution obtained using the recursive algorithm and Gauß elimination.