Algorithms for Model Checking (2IW55)

Lecture 8: Small Progress Measures for Solving Parity Games

Background material:

M. Jurdziński, *Small Progress Measures for Solving Parity Games*

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

Algorithms for Parity games

- McNaughton's/Zielonka's Recursive algorithm
- Today: Jurdziński's Small progress measures

Small progress measures (Bird's-eye view)

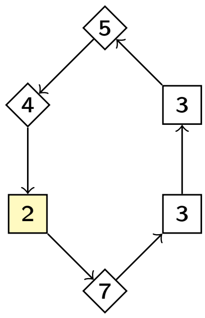
- *Characterise cycles reachable from each vertex*
  - Cycles can be used to decide the winner.
- Assign a certain *measure* to each vertex
  - "bad" priority encountered: measure decreases
  - "good" priority encountered: measure can increase
- Efficiently compute measure
  - fixed point iteration

Technische Universiteit Eindhoven University of Technology

Cycles

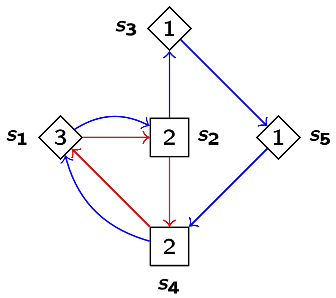
Definition (Even Cycles and Odd Cycles)

An even (resp. odd) cycle is a cycle in which the lowest priority is even (resp. odd)



Technische Universiteit Eindhoven University of Technology

$\square$ wins a vertex in a parity game iff she can ensure all cycles appearing in plays are odd.



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

Definition (Solitaire game)

G is a $\bigcirc$-solitaire game if for all vertices $v \in V_{\overline{\bigcirc}}$ we have:

$$|\{w \in V \mid (v, w) \in E\}| \leq 1$$

i.e., only one player makes (nontrivial) choices.

A strategy ϱ for player $\bigcirc$ in G induces a solitaire game $G_\varrho = (V, E_\varrho, p, (V_\Diamond, V_\square))$, where

$$E_\varrho = \{(v, w) \in E \mid v \in V_\bigcirc \Rightarrow w = \varrho(v)\}\}$$

Cycles

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

- $W \subseteq V$
- strategy ϱ for $\diamond$ closed on W.
- $G_\varrho \cap W$ is a solitaire game.

Property

ϱ is winning for player $\diamond$ from all $v \in W$ if and only if all cycles in $G_\varrho \cap W$ are even

Small progress measures (Bird's-eye view)

- Characterise cycles reachable from each vertex
  - Cycles can be used to decide the winner.
- Assign a certain measure to each vertex
  - "bad" priority encountered: measure decreases
  - "good" priority encountered: measure can increase
- Efficiently compute measure
  - fixed point iteration

Progress Measures

We wish to record information about plays such that:

- when, along a play we encounter priority i, we then have the means to ignore information about less significant priorities (i.e., $> i$)
- the information we record about priorities k outweighs information about l if $k < l$

Represent information as follows:

- Tuples to record information about priorities
- Order tuples lexicographically

Progress Measures

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

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

Example (d-tuples)

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

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

Let $d = \max\{p(v) \mid v \in V\} + 1$.

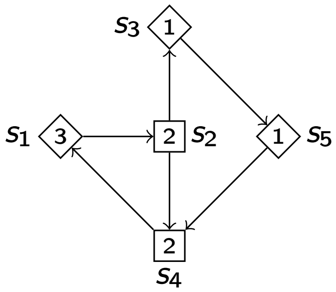
- Define $V_i = \{v \in V \mid p(v) = i\}$,
- Denote $n_i = \mid V_i \mid$, the number of vertices with priority i,

Define $\mathbb{M}^\Diamond \subseteq \mathbb{N}^d$ with:

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

Example

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



- Maximum value of $\mathbb{M}^\diamond$ is $(0, 2, 0, 1)$
- $\mathbb{M}^\diamond = \{0\} \times \{0, 1, 2\} \times \{0\} \times \{0, 1\}$

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

Definition (Parity progress measure)

Let G be an $\Box$-solitaire game. Mapping $\varrho: V \to \mathbb{M}^\diamond$ is a parity progress measure for G if for all $(v, w) \in E$:

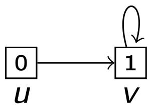
- $\varrho(v) \geq_{p(v)} \varrho(w)$ if $p(v)$ is even
- $\varrho(v) >_{p(v)} \varrho(w)$ if $p(v)$ is odd

For all strategies ψ for player $\diamond$, closed on W:

- ψ is winning for player $\diamond$ from W if and only if all cycles in $G_\psi \cap W$ are even
- All cycles in $G_\psi \cap W$ are even iff there exists a parity progress measure ϱ for $G_\psi \cap W$

Progress Measures

Problem: parity progress measures only exist for even-dominated cycles.



Second clause requires $\varrho(v) >_1 \varrho(v)$

Technische Universiteit Eindhoven University of Technology

Progress Measures

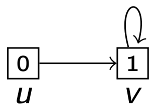
Dealing with odd-dominated cycles.

- Define $\mathbb{M}^{\diamond,\top} = \mathbb{M}^{\diamond} \cup \{\top\}$
- Extend ordering:
  - for all $m \in \mathbb{M}^{\diamond}$, define $m < \top$, $m <_i \top$, $m \neq \top$ and $m \neq_i \top$
  - $\top =_i \top$ for all i
- Replace co-domain of parity progress measures with $\mathbb{M}^{\diamond,\top}$.
- For an $\Box$-solitaire game $G = (V, E, p, (V_\diamond, V_\Box))$:
  - $W_\diamond = \{v \in V \mid \exists \varrho : \ \varrho(v) \neq \top\}$
  - $W_\Box = V \setminus W_\diamond$.
- Note: $\exists \varrho : \ \varrho(v) \neq \top$ iff the least ϱ (ordered pointwise) is such that $\varrho(v) \neq \top$

Technische Universiteit Eindhoven University of Technology

Progress Measures

Example



- Observe: $\varrho(u) = \varrho(v) = \top$
- Measure can identify both even and odd reachable cycles in a solitaire game.

Progress Measures

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

Towards measures for two-player games

Definition (Prog)

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

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

Example

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

- Suppose $p(v) = 0$, $\varrho(w) = (0,2,0,0)$.
  Then $Prog(\varrho, v, w) = (0,0,0,0)$
- Suppose $p(v) = 1$, $\varrho(w) = (0,2,0,0)$.
  Then $Prog(\varrho, v, w) = \top$
- Suppose $p(v) = 3$, $\varrho(w) = (0,2,0,0)$.
  Then $Prog(\varrho, v, w) = (0,2,0,1)$

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

Definition (Game parity progress measure)

Mapping $\varrho: V \to \mathbb{M}^{\diamond,\top}$ is a game parity progress measure if for all $v \in V$:

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

If ϱ is the least game parity progress measure for G, then:

$$\varrho(v) \neq \top$$
$$\Leftrightarrow$$
player $\diamond$ can prevent reaching $\Box$-dominated cycles

Small progress measures (Bird's-eye view)

- Characterise cycles reachable from each vertex
  - Cycles can be used to decide the winner.
- Assign a certain measure to each vertex
  - "bad" priority encountered: measure decreases
  - "good" priority encountered: measure can increase
- Efficiently compute measure
  - fixed point iteration

Computing Least Game Parity Progress Measures

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

- $\varphi, \varrho : V \to \mathbb{M}^{\diamond,\top}$.
- Define $\varphi \sqsubseteq \varrho$ if $\varphi(v) \leq \varrho(v)$ for all $v \in V$
- write $\varphi \sqsubset \varrho$ if $\varphi \sqsubseteq \varrho$ and $\varphi \neq \varrho$.

The set of mappings $([V \to \mathbb{M}^{\diamond,\top}], \sqsubseteq)$ is a complete lattice

Computing Least Game Parity Progress Measures

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

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

Observe:

- For every $v \in V$, $Lift_v$ is $\sqsubseteq$-monotone.
- A mapping $\varrho: V \to \mathbb{M}^{\Diamond,\top}$ is a game parity progress measure if and only if $Lift_v(\varrho) \sqsubseteq \varrho$ for all $v \in V$.
- Least game parity progress measure computable by fixpoint iteration (algorithm Lfp of Lecture 2)

Computing Least Game Parity Progress Measures

Algorithm $SPM(G)$

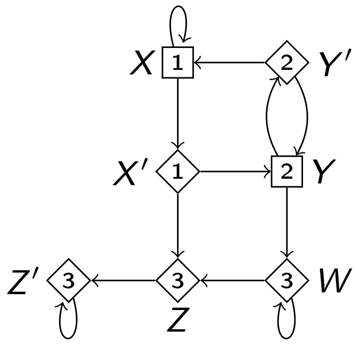
$\varrho \colon V \to \mathbb{M}^{\Diamond,\top} \leftarrow \lambda v \in V.(0, \ldots, 0)$
while $\varrho \sqsubset Lift_v(\varrho)$ for some $v \in V$ **do**
  $\varrho \leftarrow Lift_v(\varrho)$
end while

Post condition:

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

Small progress measures (example)

Consider parity game G:



Maximum value of $\mathbb{M}^{\diamond}$ 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	$(0,0,0,0)$

Small progress measures (example) (2)

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

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) (3)

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

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	$(0,0,0,0)$

Small progress measures (example) (4)

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) (5)

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	$(0,0,0,0)$

Small progress measures (example) (6)

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

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) (7)

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	$(0,0,0,0)$

Small progress measures (example) (8)

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) (9)

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	$(0,0,0,0)$

Small progress measures (example) (10)

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) (11)

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	$(0,0,0,0)$

Small progress measures (example) (12)

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) (13)

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	$(0,0,0,1)$

Small progress measures (example) (14)

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) (15)

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	$\top$

Small progress measures (example) (16)

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	$\top$

Technische Universiteit Eindhoven University of Technology

Small progress measures (example) (17)

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

Technische Universiteit Eindhoven University of Technology

Complexity and Strategies

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

Let $\varrho : V \to \mathbb{M}^{\bigcirc,\top}$ be the least game parity progress measure.

- Define strategy $\varrho : V_\Diamond \to V$ for player $\Diamond$, by setting $\varrho(v)$ to be a successor w of $v \in V_\Diamond$ that minimises $\varrho(w)$
- ϱ is a winning strategy for player $\Diamond$ from $\{v \in V \mid \varrho(v) \neq \top\}$
- Strategy for $\Box$ cannot be inferred directly
  - ...but can be computed on-the-fly . . . . Gazda&Willemse'13

Complexity and Strategies

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

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

Worst-case running time complexity:

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

Lowerbound on worst-case:

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

Summary Part II

- Model checking L_μ = solving Boolean equation systems
  - Gauß Elimination for solving BES . . . . $\mathcal{O}(2^{|\mathcal{E}|})$
- Solving BES = solving Parity games
  - Recursive . . . . $\mathcal{O}(m \cdot n^d)$
  - Small progress measures . . . . $\mathcal{O}(d \cdot m \cdot (\frac{n}{\lfloor d/2 \rfloor})^{\lfloor d/2 \rfloor})$
  - bigstep (combination of the two above) . . . . $\approx \mathcal{O}(n^{d/3})$