A Quasi-Polynomial Black-Box Algorithm for Fixed Point Evaluation

André Arnold, Damian Niwiński, **Paweł Parys**

CSL 2021

Plan

- parity games ≈ modal μ-calculus

$$\cap\!\!\!\!|$$

(black-box) fixed point evaluation

- quasi-polynomial algorithms for parity games

$\Downarrow$

quasi-polynomial black-box algorithms for fixed point evaluation

- Our algorithm is an abstract version of recent quasi-polynomial algorithms solving parity games
- We unify two kinds of parity-games algorithms (asymmetric, symmetric) in a common framework
- Some lower bounds for the method (universal trees are needed)

Considered problem: fixed point evaluation

Compute: $\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$

where $x_i \in \{0,1\}^n$

$f : (\{0,1\}^n)^d \to \{0,1\}^n$ monotone

access to f: only evaluation for given arguments (f is a black-box)

Considered problem: fixed point evaluation

Compute: $\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$

where $x_i \in \{0,1\}^n$

$f : (\{0,1\}^n)^d \to \{0,1\}^n$ monotone

access to f: only evaluation for given arguments (f is a black-box)

Relation to parity games

parity game (n nodes, d priorities) → fixed point evaluation (n bits, d arguments) f of a special form

parity game ($exp(n)$ nodes, d priorities) ← fixed point evaluation (n bits, d arguments) arbitrary f

Considered problem: fixed point evaluation

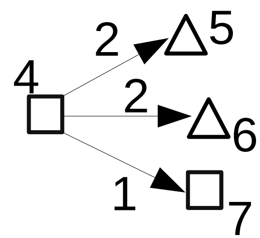
Compute: $\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$

$f: (\{0,1\}^n)^d \to \{0,1\}^n$ monotone

Relation to parity games

parity game
(n nodes, d priorities)

fixed point evaluation
(n bits, d arguments)
f of a special form



$f(x_1,x_2)[4] = x_1[7] \vee x_2[5] \vee x_2[6]$

$x_i \subseteq V$

f describes a game:

$f(x_1,x_2,\ldots,x_{d-1},x_d)$ returns nodes from which Eve can reach in one step:

- a node in x_1 via an edge of priority 1, or
- a node in x_2 via an edge of priority 2, or
- …

Then $\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$ is the set of nodes where Eve wins the parity game

Parity games vs. fixed point evaluation

$$\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$$

For parity games:

- f is of a special form: every output bit is either AND or OR of some input bits
- the game graph can be accessed also in other ways, not only by evaluating f

Recent quasipolynomial algorithms for parity games:

- access the game graph only by evaluating f
- work for arbitrary f, not only for f coming from parity games

$\Downarrow$

After a careful analysis, they give black-box algorithms for fixed point evaluation

This paper / this talk:

- Why?
- How to prove this in a nice way?

Recent results on parity games

- Calude, Jain, Khoussainov, Li, Stephan 2017
- Fearnley, Jain, Schewe, Stephan, Wojtczak 2017
- Jurdziński, Lazić 2017
- Lehtinen 2018

$\Big\}$ asymmetric algo. (separator approach)

complexity:
$n^{\lg(d/\lg n)+O(1)} \approx |U_{n,d}|$

- Bojańczyk, Czerwiński 2018
- Czerwiński, Daviaud, Fijalkow, Jurdziński, Lazić, Parys 2019

- Parys 2019
- Lehtinen, Schewe, Wojtczak 2019
- Jurdziński, Morvan 2020

$\Big\}$ symmetric algo. (recursive)

complexity:
$n^{2\lg(d/\lg n)+O(1)} \approx |U_{n,d}|^2$

- Jurdziński, Morvan, Ohlmann, Thejaswini 2020 – symmetric, in $n^{\lg(d/\lg n)+O(1)} \approx |U_{n,d}|$

fixed point evaluation:
- Hausmann, Schröder 2019
- Hausmann, Schröder 2020

(blue = after writing this paper)

Standard exponential algorithm

Notation: $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = \nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$ for $\Theta=\nu\mu\ldots\nu\mu$

$f^{\hookrightarrow A}(x_1,x_2,\ldots,x_{d-1}) = f\,(x_1,x_2,\ldots,x_{d-1},A)$

Algorithm evaluating $|(\Theta,f,(\mathbf{0},\mathbf{1}))|$ (where $j=1,2,\ldots,n$)

for $\Theta=\mu\Theta'$:

$A_0 = \mathbf{0}$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(\mathbf{0},\mathbf{1}))|$

return A_n

for $\Theta=\nu\Theta'$:

$B_0 = \mathbf{1}$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(\mathbf{0},\mathbf{1}))|$

return B_n

Standard exponential algorithm

Notation: $|(\Theta,f,(\mathbf{0},\mathbf{1}))|=\nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f(x_1,x_2,\ldots,x_{d-1},x_d)$ for $\Theta=\nu\mu\ldots\nu\mu$

$f^{\hookrightarrow A}(x_1,x_2,\ldots,x_{d-1}) = f(x_1,x_2,\ldots,x_{d-1},A)$

Algorithm evaluating $|(\Theta,f,(\mathbf{0},\mathbf{1}))|$ (where $j=1,2,\ldots,n$)

for $\Theta=\mu\Theta'$:

$A_0=\mathbf{0}$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(\mathbf{0},\mathbf{1}))|$

return A_n

for $\Theta=\nu\Theta'$:

$B_0=\mathbf{1}$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(\mathbf{0},\mathbf{1}))|$

return B_n

How to make it quasipolynomial?

- do not start from $\mathbf{0}$ / $\mathbf{1}$, but from some intermediate values (restrictions)
- perform less iterations (follow a structure of some universal trees)

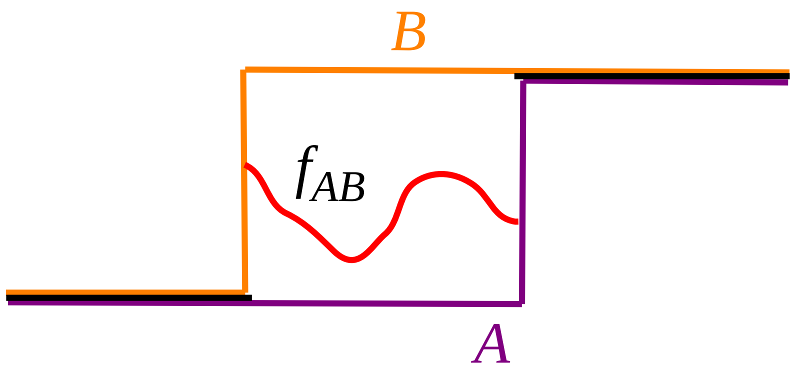
Restrictions

$A \leq B$

Notation: $f_{AB}(x_1,x_2,\ldots,x_d) = A + B * f\,(x_1,x_2,\ldots,x_d)$

= sup
= bitwise OR

= inf
= bitwise AND



Restrictions

$A \le B$

Notation: $f_{AB}(x_1,x_2,\ldots,x_d) = A + B * f(x_1,x_2,\ldots,x_d)$

$|(\Theta,f,(A,B))| = \nu x_d.\mu x_{d-1}\ldots\nu x_2.\mu x_1.f_{AB}(x_1,x_2,\ldots,x_{d-1},x_d)$ for $\Theta=\nu\mu\ldots\nu\mu$

Algorithm evaluating $|(\Theta,f,(A,B))|$ (where $j=1,2,\ldots,n$)

for $\Theta=\mu\Theta'$:

$A_0 = A$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(A_{j-1},B))|$

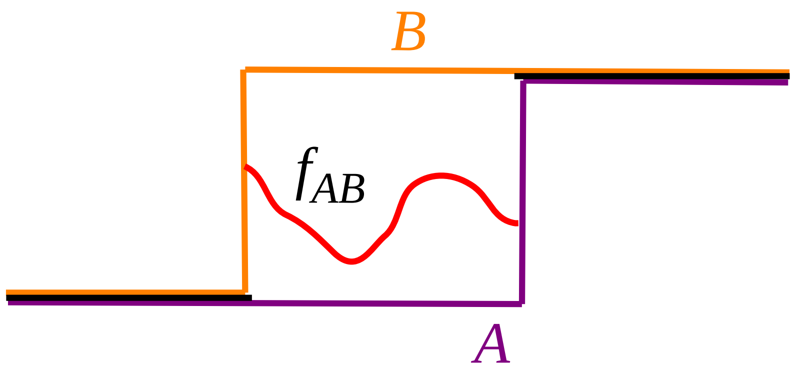
return A_n

for $\Theta=\nu\Theta'$:

$B_0 = B$

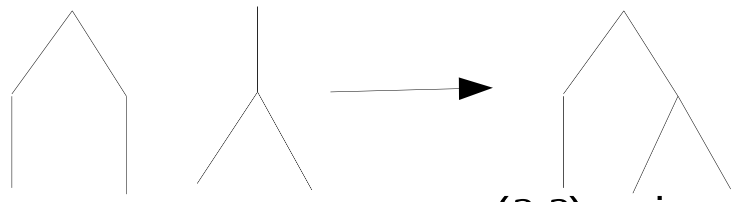
$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|$

return B_n



Universal trees

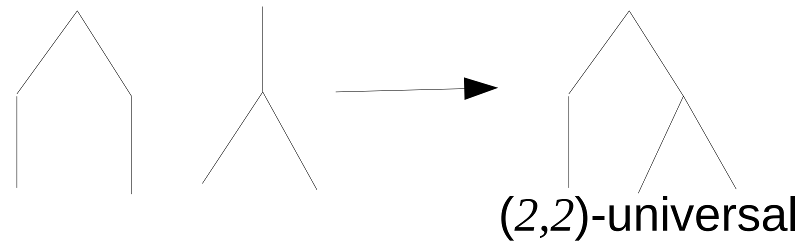
A tree U (of height h) is <u>(n,h)-universal</u> if every tree of height h with n leaves embeds in U.



$(2,2)$-universal

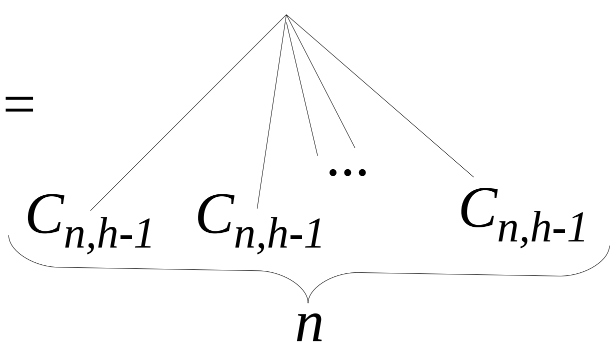
Universal trees

A tree U (of height h) is <u>(n,h)-universal</u> if every tree of height h with n leaves embeds in U.

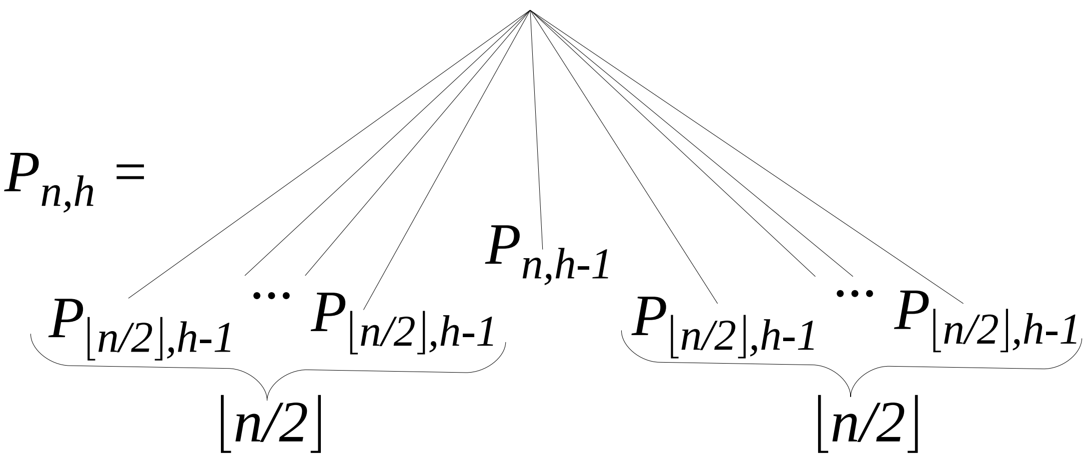


<u>Examples</u>:

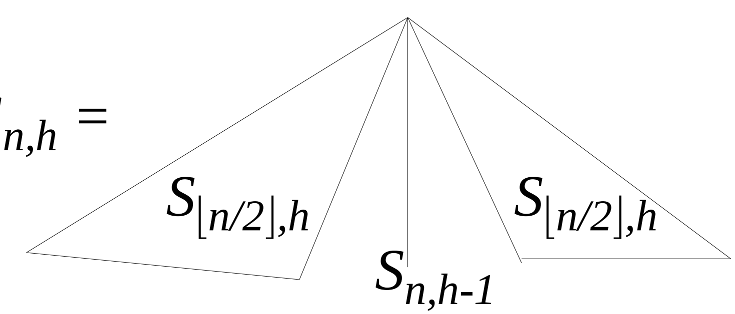
$C_{n,h} =$



$P_{n,h} =$

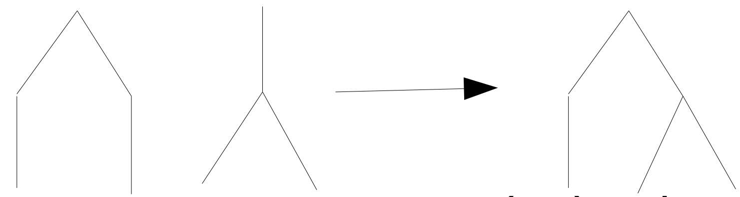


$S_{n,h} =$



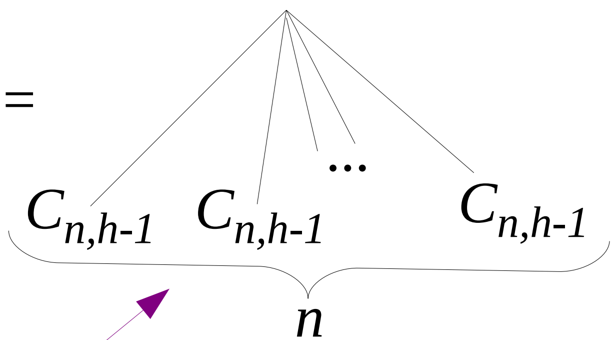
Universal trees

A tree U (of height h) is (n,h)-universal if every tree of height h with n leaves embeds in U.



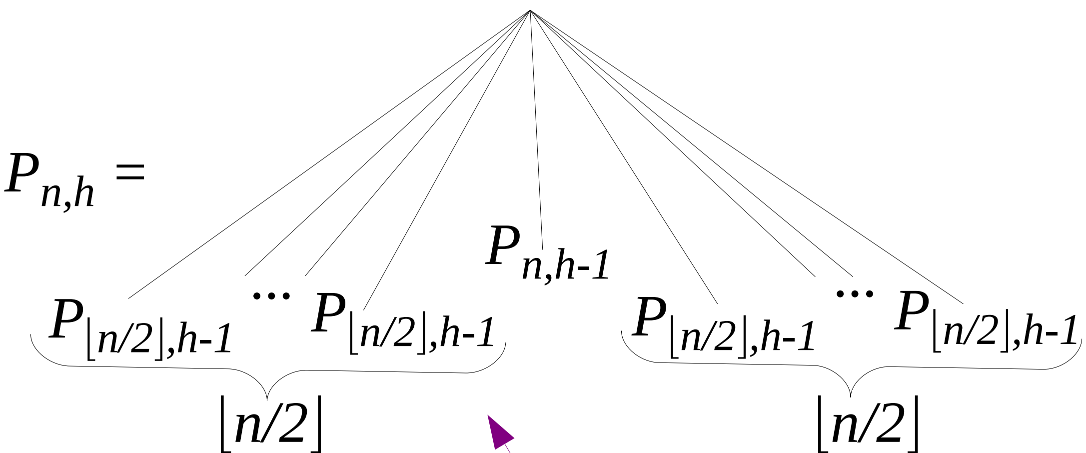
Examples:

$C_{n,h}$ =



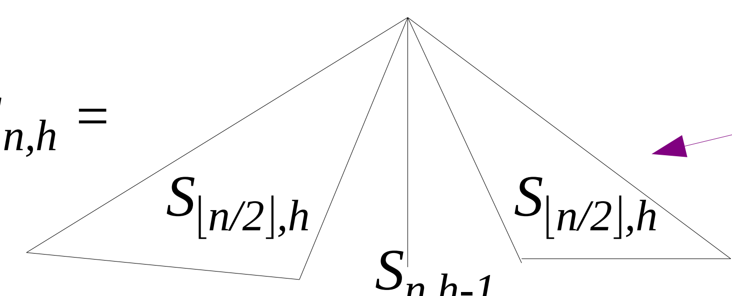
size n^h

$P_{n,h}$ =



size $n^{\lg n + \lg(h/\lg n)+O(1)}$

$S_{n,h}$ =



size $n^{\lg(h/\lg n)+O(1)}$

Symmetric algorithm based on universal trees

≈ the symmetric algorithm for parity games

U,V – (universal) trees

Definition / Algorithm evaluating $|(\Theta,f,(A,B))|_{U,V}$ (where $j=1,2,\ldots,p$)

for $\Theta=\mu\Theta'$, $U=\langle U_1,\ldots,U_p\rangle$

$A_0 = A$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(A_{j-1},B))|_{U_j,V}$

return A_p

for $\Theta=\nu\Theta'$, $V=\langle V_1,\ldots,V_p\rangle$

$B_0 = B$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|_{U,V_j}$

return B_p

Symmetric algorithm based on universal trees

≈ the symmetric algorithm for parity games

U,V – (universal) trees

Definition / Algorithm evaluating $|(\Theta,f,(A,B))|_{U,V}$ (where $j=1,2,\ldots,p$)

for $\Theta=\mu\Theta'$, $U=\langle U_1,\ldots,U_p\rangle$

$A_0 = A$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(A_{j-1},B))|_{U_j,V}$

return A_p

for $\Theta=\nu\Theta'$, $V=\langle V_1,\ldots,V_p\rangle$

$B_0 = B$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|_{U,V_j}$

return B_p

Correctness

If U,V are $(n,d/2)$-universal then $|(\Theta,f,(\mathbf{0},\mathbf{1}))|_{U,V} = |(\Theta,f,(\mathbf{0},\mathbf{1}))|$.

Proof is based on:

- dominions
- dominion decomposition

adapted from parity games [Jurdziński, Morvan 2020]

Symmetric algorithm based on universal trees

≈ the symmetric algorithm for parity games

U,V – (universal) trees

Definition / Algorithm evaluating $|(\Theta,f,(A,B))|_{U,V}$ (where $j=1,2,\ldots,p$)

for $\Theta=\mu\,\Theta'$, $U=\langle U_1,\ldots,U_p\rangle$

$A_0 = A$

$A_j = |(\Theta',f^{\hookrightarrow A_{j-1}},(A_{j-1},B))|_{U_j,V}$

return A_p

for $\Theta=\nu\,\Theta'$, $V=\langle V_1,\ldots,V_p\rangle$

$B_0 = B$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|_{U,V_j}$

return B_p

Correctness

If U,V are $(n,d/2)$-universal then $|(\Theta,f,(\mathbf{0},\mathbf{1}))|_{U,V} = |(\Theta,f,(\mathbf{0},\mathbf{1}))|$.

Proof is based on:

- dominions
- dominion decomposition

adapted from parity games [Jurdziński, Morvan 2020]

Time complexity: $|U|\cdot|V| = n^{2\lg(d/\lg n)+O(1)}$
(two universal trees)

Asymmetric algorithm (Seidl's idea, 1996)

- evaluate recursively

time: n^d

Seidl '96

- create a system of **least** fixed point equations, and solve it

time: $n^{d/2+1}$

- universal trees
- restrictions
- evaluate recursively

time: $n^{2\lg(d/\lg n)+O(1)}$

- universal trees
- restrictions
- create a system of **least** fixed point equations, and solve it

time: $n^{2\lg(d/\lg n)/2+O(1)}$

$\approx$

symmetric algorithm for parity games

$\approx$

asymmetric algorithm for parity games

Asymmetric algorithm (Seidl's idea, 1996)

[exponential version]

Algorithm evaluating $|(\Theta, f, (A,B))|$

(where $j=1,2,\ldots,n$)

for $\Theta = \mu\Theta'$:

create equation:

$x = |(\Theta', f^{\hookrightarrow x}, (A,B))|$

return x

(where x is a fresh variable)

the result is: $\mu x.|(\Theta', f^{\hookrightarrow x}, (A,B))|_V$

for $\Theta = \nu\Theta'$:

create equations:

$B_0 = B$

$B_j = |(\Theta', f^{\hookrightarrow B_{j-1}}, (A,B))|$

return B_n

(where $B_0, B_1, \ldots, B_n$ are fresh variables)

We obtain a system of **least** fixed point equations (only μ) of size $n^{d/2}$.
It can be solved in linear time.

Asymmetric algorithm based on universal trees

V – (universal) tree

Definition of $|(\Theta,f,(A,B))|_V$

for $\Theta=\mu\Theta'$

return $\mu x.|(\Theta',f^{\hookrightarrow x},(A,B))|_V$

(where $j=1,2,\ldots,p$)

for $\Theta=\nu\Theta'$, $V=\langle V_1,\ldots,V_p\rangle$

$B_0 = B$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|_{V_j}$

return B_p

Asymmetric algorithm based on universal trees

$\approx$ the asymmetric algorithm for parity games

V – (universal) tree

Algorithm evaluating $|(\Theta,f,(A,B))|_V$

for $\Theta=\mu\Theta'$

create equation:

$x = |(\Theta',f^{\hookrightarrow x},(A,B))|_V$

return x

(where x is a fresh variable)

(where $j=1,2,\ldots,p$)

for $\Theta=\nu\Theta'$, $V=\langle V_1,\ldots,V_p\rangle$

create equations:

$B_0= B$

$B_j = |(\Theta',f^{\hookrightarrow B_{j-1}},(A,B_{j-1}))|_{V_j}$

return B_p

(where $B_0,B_1,\ldots,B_p$ are fresh variables)

We obtain a system of **least** fixed point equations (only μ) of size $|V|=n^{\lg(d/\lg n)+O(1)}$.

It can be solved in linear time.

Correctness (of the symmetric variant)

Sup-dominion for $(\Theta,f,(A,B))$: a value D such that $D=|(\Theta,f,(A,D))|$
intuition: one can prove that $|(\Theta,f,(A,B))|\geq D$ without looking for bits outside of D

(like in parity games: Even can win from D without going outside of D)

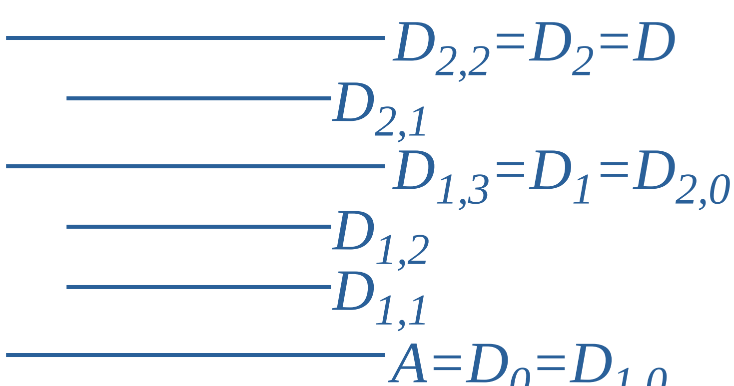
Correctness (of the symmetric variant)

Sup-dominion for $(\Theta,f,(A,B))$: a value D such that $D=|(\Theta,f,(A,D))|$
intuition: one can prove that $|(\Theta,f,(A,B))|\geq D$ without looking for bits outside of D

(like in parity games: Even can win from D without going outside of D)

Sup-dominion decomposition for $(\Theta,f,(A,B))$
a pair (D,H) such that D is a dominion for $(\Theta,f,(A,B))$ and
if $\Theta=\mu\Theta'$ then
$H=\langle(D_1,H_1),..,(D_k,H_k)\rangle$ s.t. $D_k=D$ and for $D_0=A$
every (D_i,H_i) is a sup-dominion decomposition for $(\Theta',f^{\hookrightarrow D_{i-1}},(D_{i-1},D))$
if $\Theta=\nu\Theta'$ then
(D,H) is a sup-dominion decomposition for $(\Theta',f^{\hookrightarrow D},(A,D))$

$$(D,\langle(D_1,\langle(D_{1,1},\langle\rangle),(D_{1,2},\langle\rangle),(D_{1,3},\langle\rangle)\rangle),(D_2,\langle(D_{2,1},\langle\rangle),(D_{2,2},\langle\rangle)\rangle)\rangle)$$



$\mu \quad \nu \quad \mu$

$f(D_{1,1},D_1,D_0)\geq D_{1,2}$

Correctness (of the symmetric variant)

Theorem: If U,V are $(n,d/2)$-universal, then
$|(\Theta,f,(A,B))| = |(\Theta,f,(A,B))|_{U,V}$

Correctness (of the symmetric variant)

Theorem: If U,V are $(n,d/2)$-universal, then
$|(\Theta,f,(A,B))| = |(\Theta,f,(A,B))|_{U,V}$

Technical lemma: If A $\leq$ C $\leq |(\Theta,f,(A,B))| \leq D \leq B$ then

$$|(\Theta,f,(A,B))|=|(\Theta,f,(C,D))|$$

Proof: definition + induction

Correctness (of the symmetric variant)

Theorem: If U,V are $(n,d/2)$-universal, then
$|(\Theta,f,(A,B))| = |(\Theta,f,(A,B))|_{U,V}$

Proof
- $D=|(\Theta,f,(A,B))|$ is a sup-dominion
- Lemma 1: every sup-dominion D has a sup-dominion decomposition (D,H)
- It has a shape of a tree T_H of height $d/2$ with at most n leaves
- Lemma 2: if (D,H) is a sup-dominion decomposition for $|(\Theta,f,(A,B))|$ then $D \leq |(\Theta,f,(A,B))|_{T_H,V}$ for every V
- If T embeds in U then $|(\Theta,f,(A,B))|_{T,V} \leq |(\Theta,f,(A,B))|_{U,V}$
- + other side – by symmetry

Technical lemma: If $A \leq C \leq |(\Theta,f,(A,B))| \leq D \leq B$ then
$$|(\Theta,f,(A,B))|=|(\Theta,f,(C,D))|$$

Proof: definition + induction

Correctness (of the symmetric variant)

Lemma 1: Every sup-dominion D has a sup-dominion decomposition (D,H)

Proof

Assumption: D is a sup-dominion for $(\Theta,f,(A,B))$

Case $\Theta=\nu\Theta'$

- [by definition: decomposition for $(\Theta,f,(A,B))$ = decomposition for $(\Theta',f^{\hookrightarrow D},(A,D))$]
- immediate: D is also a sup-dominion for $(\Theta',f^{\hookrightarrow D},(A,D)) \rightarrow$ we can use I.H.

Correctness (of the symmetric variant)

Lemma 1: Every sup-dominion D has a sup-dominion decomposition (D,H)

Proof

Assumption: D is a sup-dominion for $(\Theta,f,(A,B))$

Case $\Theta=\nu\Theta'$

- [by definition: decomposition for $(\Theta,f,(A,B))$ = decomposition for $(\Theta',f^{\hookrightarrow D},(A,D))$]
- immediate: D is also a sup-dominion for $(\Theta',f^{\hookrightarrow D},(A,D)) \rightarrow$ we can use I.H.

Case $\Theta=\mu\Theta'$

- [by definition: we need $(D_1,H_1),..,(D_k,H_k)$ s.t. $D_k=D$ and for $D_0=A$ every (D_i,H_i) is a sup-dominion decomposition for $(\Theta',f^{\hookrightarrow D_{i-1}},(D_{i-1},B))$]
- We take $D_i=|(\Theta',f^{\hookrightarrow D_{i-1}},(D_{i-1},D))|$ as long as $D_i<D$
- We construct decompositions H_i using I.H.

Correctness (of the symmetric variant)

Lemma 1: Every sup-dominion D has a sup-dominion decomposition (D,H)

Proof

Assumption: D is a sup-dominion for $(\Theta,f,(A,B))$

Case $\Theta = \nu\,\Theta'$

- [by definition: decomposition for $(\Theta,f,(A,B))$ = decomposition for $(\Theta',f^{\hookrightarrow D},(A,D))$]
- immediate: D is also a sup-dominion for $(\Theta',f^{\hookrightarrow D},(A,D))$ $\rightarrow$ we can use I.H.

Case $\Theta = \mu\,\Theta'$

- [by definition: we need $(D_1,H_1),..,(D_k,H_k)$ s.t. $D_k=D$ and for $D_0=A$ every (D_i,H_i) is a sup-dominion decomposition for $(\Theta',f^{\hookrightarrow D_{i-1}},(D_{i-1},B))$]
- We take $D_i=|(\Theta',f^{\hookrightarrow D_{i-1}},(D_{i-1},D))|$ as long as $D_i<D$
- We construct decompositions H_i using I.H.

Lemma 2: If (D,H) is a sup-dominion decomposition for $|(\Theta,f,(A,B))|$ then $D \le |(\Theta,f,(A,B))|_{T_H,V}$ for every V

Proof: definitions + induction

A lower bound (for our method)

Theorem: Fix n,d.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{U,V}$ for all f, then U,V are $(n,d/2)$-universal.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{V}$ for all f, then V is $(n,d/2)$-universal.

Corollary:
It is known that every universal tree has size at least $n^{\lg(h/\lg n)+\Omega(1)}$
Thus our algorithm cannot work faster (using potentially some smaller tree).

A lower bound (for our method)

Theorem: Fix n,d.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{U,V}$ for all f, then U,V are $(n,d/2)$-universal.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{V}$ for all f, then V is $(n,d/2)$-universal.

Corollary:
It is known that every universal tree has size at least $n^{\lg(h/\lg n)+\Omega(1)}$
Thus our algorithm cannot work faster (using potentially some smaller tree).

Remark:
It is enough to assume equality for functions f defined by parity games
(so the lower bound applies also to parity games)

A lower bound (for our method)

Theorem: Fix n,d.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{U,V}$ for all f, then U,V are $(n,d/2)$-universal.
If $|(\Theta,f,(\mathbf{0},\mathbf{1}))| = |(\Theta,f,(\mathbf{0},\mathbf{1}))|_{V}$ for all f, then V is $(n,d/2)$-universal.

Corollary:
It is known that every universal tree has size at least $n^{\lg(d/\lg n)+\Omega(1)}$
Thus our algorithm cannot work faster (using potentially some smaller tree).

Remark:
It is enough to assume equality for functions f defined by parity games
(so the lower bound applies also to parity games)

Proof idea:
By contradiction: If some T (with n leaves) does not embed in U, then we can construct f such that the only sup-dominion decomposition has shape T.
For this f the algorithms does not work.

Conclusions

- quasi-polynomial algorithms for fixed-point evaluation
- an abstract formulation using universal trees
- unified treatment of symmetric / asymmetric variants
- a lower bound for the method

Open problem:

- prove a (quasi-polynomial?) lower bound for the number of queries for black-box fixed point evaluation
  [we only have $\Omega(n^2/\log n)$ – Parys 2009]

Thank you!