A Measure of Space for Computing over the Reals

Paulin Jacobé de Naurois

LIPN - Université Paris XIII

Plan of the Talk

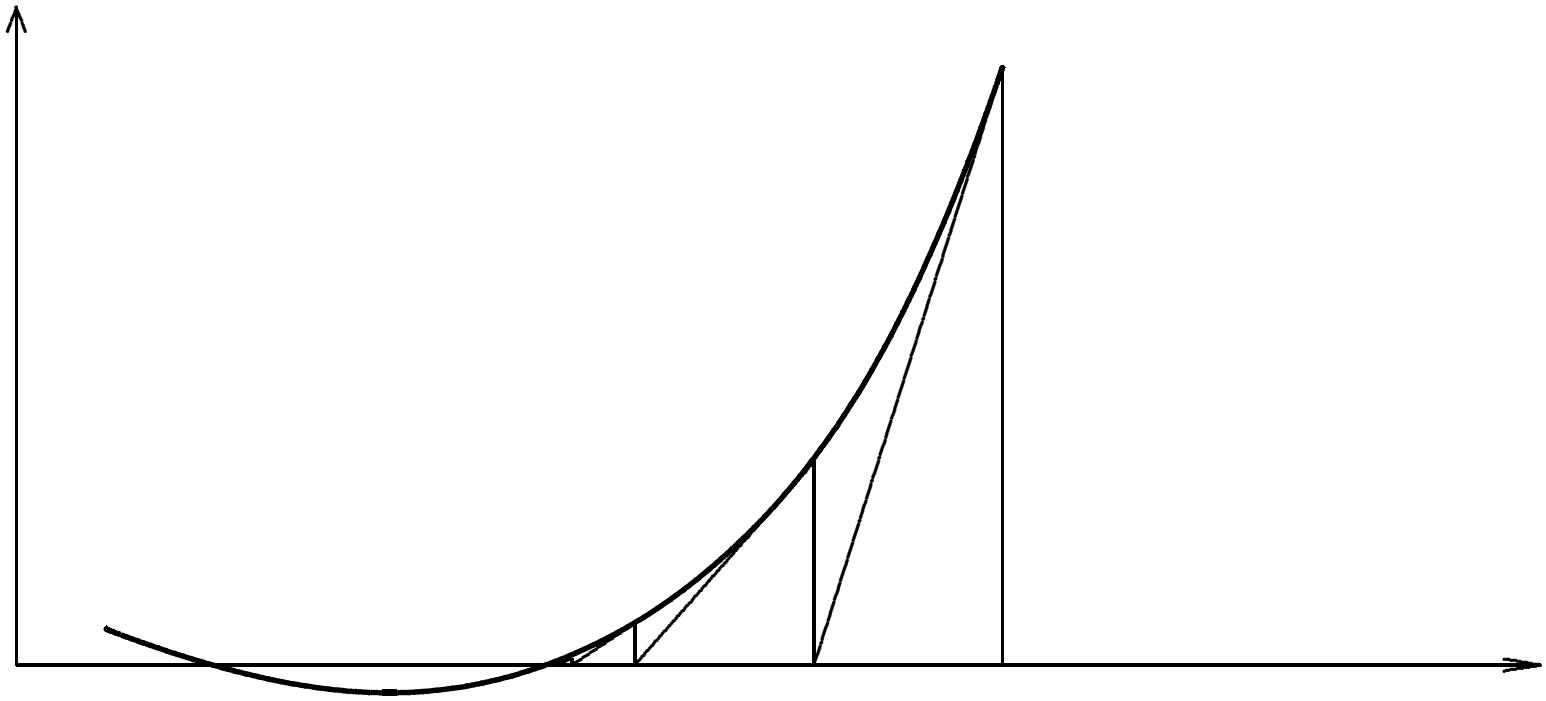
- The BSS Model of Computation over the Reals
- Michaux's Result - Computing in Constant Space
- Koiran's Weak Model
- A Weak Measure of Space

Motivation of Blum, Shub et Smale

Provide a theoretical framework for studying calculability and complexity properties for natural problems and algorithms over real numbers, in particular, problems of numerical analysis, geometry, topology...

Motivation of Blum, Shub et Smale

Provide a theoretical framework for studying calculability and complexity properties for natural problems and algorithms over real numbers, in particular, problems of numerical analysis, geometry, topology...
Example:



Newton's Method for finding a zero of a function.

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

- the tape cells hold arbitrary numbers in $\mathbb{R}$

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

- the tape cells hold arbitrary numbers in $\mathbb{R}$
- some computation nodes compute an arithmetical operation $+, -, *, /$, with unbounded precision, at unit cost,

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

- the tape cells hold arbitrary numbers in $\mathbb{R}$
- some computation nodes compute an arithmetical operation $+, -, *, /$, with unbounded precision, at unit cost,
- some constant nodes write a constant of $\mathbb{R}$ on the tape,

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

- the tape cells hold arbitrary numbers in $\mathbb{R}$
- some computation nodes compute an arithmetical operation $+, -, *, /$, with unbounded precision, at unit cost,
- some constant nodes write a constant of $\mathbb{R}$ on the tape,
- some branch nodes branch on a test "$a \leq b$", at unit cost,

The Model

A BSS Machine is essentially a Turing Machine over $\mathbb{R}$, such that

- the tape cells hold arbitrary numbers in $\mathbb{R}$
- some computation nodes compute an arithmetical operation $+, -, *, /$, with unbounded precision, at unit cost,
- some constant nodes write a constant of $\mathbb{R}$ on the tape,
- some branch nodes branch on a test "$a \leq b$", at unit cost,
- some shift nodes move the scanning head on the tape,
- some copy nodes duplicate the content of some cells.

Input - Output

Convention: One Input Tape, one Work Tape, one Output Tape.

- Inputs and Outputs are vectors in $\mathbb{R}^* = \bigsqcup_{n \in \mathbb{N}} \mathbb{R}^n$,

Input - Output

Convention: One Input Tape, one Work Tape, one Output Tape.

- Inputs and Outputs are vectors in $\mathbb{R}^* = \bigsqcup_{n \in \mathbb{N}} \mathbb{R}^n$,
- Decision Problems - or Languages - are subsets of $\mathbb{R}^*$.

Calculability

- There exist universal BSS machines.

Calculability

- There exist universal BSS machines.
- The Halting problem is undecidable.

Calculability

- There exist universal BSS machines.
- The Halting problem is undecidable.
- The Mandelbrot Set is undecidable.

Calculability

- There exist universal BSS machines.
- The Halting problem is undecidable.
- The Mandelbrot Set is undecidable.
- The set of points that converge under Newton's algorithm is undecidable.

Sequential Time Complexity

Unit measure of time: # of computation steps

Sequential Time Complexity

Unit measure of time: # of computation steps

- $\mathrm{P}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in polynomial time

Sequential Time Complexity

Unit measure of time: # of computation steps

- $\mathrm{P}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in polynomial time
- $\mathrm{NP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in non-deterministic polynomial time (existential witnesses in $\mathbb{R}^*$)

Sequential Time Complexity

Unit measure of time: # of computation steps

- $\mathrm{P}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in polynomial time
- $\mathrm{NP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in non-deterministic polynomial time (existential witnesses in $\mathbb{R}^*$)
- $\mathrm{coNP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in non-deterministic polynomial time (universal witnesses in $\mathbb{R}^*$)

Sequential Time Complexity

Unit measure of time: # of computation steps

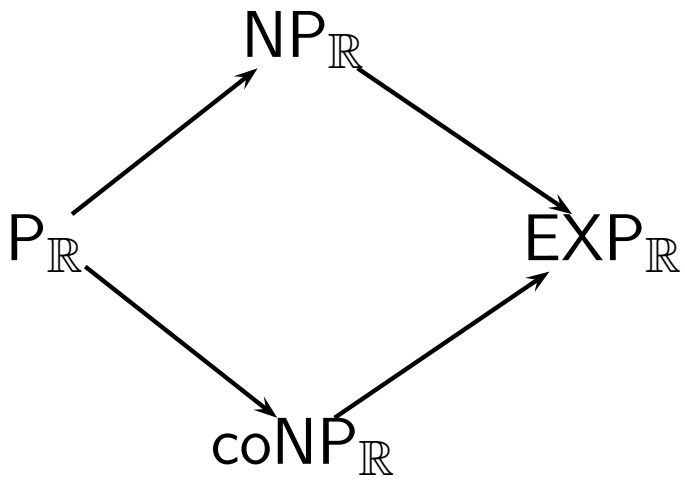
- $\mathrm{P}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in polynomial time
- $\mathrm{NP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in non-deterministic polynomial time (existential witnesses in $\mathbb{R}^*$)
- $\mathrm{coNP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in non-deterministic polynomial time (universal witnesses in $\mathbb{R}^*$)
- $\mathrm{EXP}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided in exponential time.

Complexity (2)

There exist natural $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{coNP}_{\mathbb{R}}$-complete problems. Ex: $4\mathrm{FEAS}_{\mathbb{R}}$ (existence of a zero for a real polynomial of degree 4) is $\mathrm{NP}_{\mathbb{R}}$-complete (reductions in $\mathrm{P}_{\mathbb{R}}$).

Complexity (2)

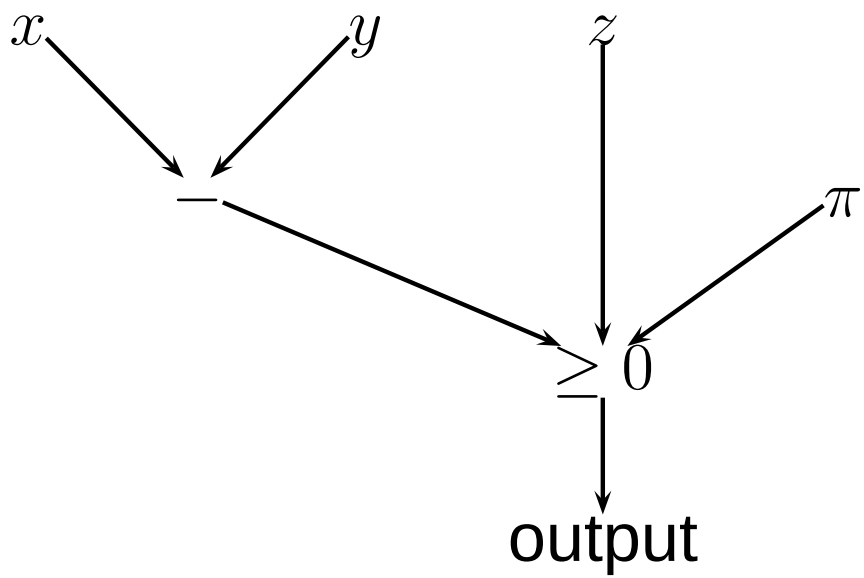
There exist natural $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{coNP}_{\mathbb{R}}$-complete problems.
Ex: $4\mathrm{FEAS}_{\mathbb{R}}$ (existence of a zero for a real polynomial of degree 4) is $\mathrm{NP}_{\mathbb{R}}$-complete (reductions in $\mathrm{P}_{\mathbb{R}}$).
Inclusions:



Question: $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$?

Algebraic Circuits

An algebraic circuit $\mathscr{C}$ computes $F_{\mathscr{C}} : \mathbb{R}^n \to \mathbb{R}^m$.



if $x - y \geq 0$ then z else π, $n = 3, m = 1$

Parallel Computation

Some complexity classes can be defined in terms of circuits:

Parallel Computation

Some complexity classes can be defined in terms of circuits:

- $\mathsf{NC}^k_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided by a uniform family of circuits of polynomial size and $O(\log(n)^k)$ depth.

Parallel Computation

Some complexity classes can be defined in terms of circuits:

- $\mathsf{NC}^k_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided by a uniform family of circuits of polynomial size and $O(\log(n)^k)$ depth.
- $\mathsf{NC}_{\mathbb{R}} = \bigcup_{k\in\mathbb{N}} \mathsf{NC}^k_{\mathbb{R}}$.

Parallel Computation

Some complexity classes can be defined in terms of circuits:

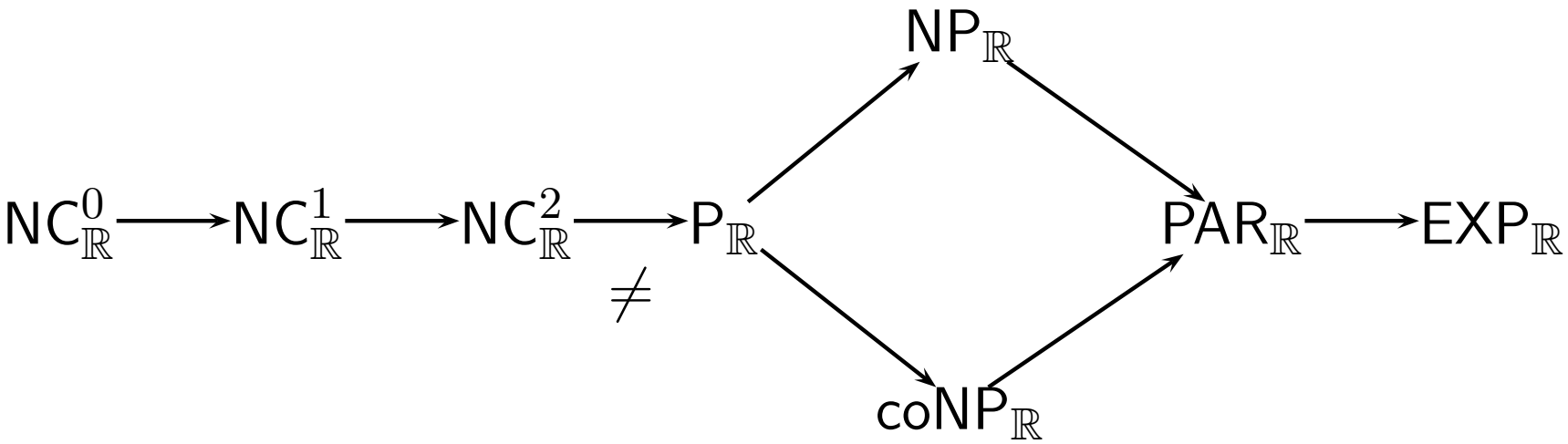
- $\mathsf{NC}^k_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided by a uniform family of circuits of polynomial size and $O(\log(n)^k)$ depth.
- $\mathsf{NC}_{\mathbb{R}} = \bigcup_{k\in\mathbb{N}} \mathsf{NC}^k_{\mathbb{R}}$.
- $\mathsf{PAR}_{\mathbb{R}}$: subsets of $\mathbb{R}^*$ decided by a uniform family of circuits of polynomial depth.

Complexity (3)

there exist natural $\mathsf{P}_{\mathbb{R}}$-complete problems (reductions in $\mathsf{NC}^2_{\mathbb{R}}$).
Ex: $\mathsf{RCDP}_{\mathbb{R}}$ (Real Circuit Decision Procedure).

Complexity (3)

there exist natural $\mathsf{P}_{\mathbb{R}}$-complete problems (reductions in $\mathsf{NC}^2_{\mathbb{R}}$).
Ex: $\mathsf{RCDP}_{\mathbb{R}}$ (Real Circuit Decision Procedure).
Inclusions:



Questions of Space

Unit measure of space: # of tape cells used.

Michaux's Result
Let $L \subseteq \mathbb{R}^*$ be a language decided in bounded time by a machine M. There exists $k \in \mathbb{N}$ and a machine M' deciding L in bounded time and working space less than k.

Motivation of this Work

Let M be a boolean algorithm in LOGSPACE

Motivation of this Work

Let M be a boolean algorithm in LOGSPACE
Let M' be a real algorithm, such that:

- M' reads $(x_1, \ldots, x_n) \in \mathbb{R}^n$.
- M' computes $(\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$,

$$\sigma_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- M' applies M on input $(\sigma_1, \ldots, \sigma_n)$

Motivation of this Work

Let M be a boolean algorithm in LOGSPACE
Let M' be a real algorithm, such that:

- M' reads $(x_1, \ldots, x_n) \in \mathbb{R}^n$.
- M' computes $(\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$,

$$\sigma_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- M' applies M on input $(\sigma_1, \ldots, \sigma_n)$

$M' \in \mathsf{NC}^2_{\mathbb{R}}$.

Motivation of this Work

Let M be a boolean algorithm in LOGSPACE
Let M' be a real algorithm, such that:

- M' reads $(x_1, \ldots, x_n) \in \mathbb{R}^n$.
- M' computes $(\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$,

$$\sigma_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- M' applies M on input $(\sigma_1, \ldots, \sigma_n)$

$M' \in \mathsf{NC}^2_{\mathbb{R}}$. Existence of a more natural class?

Michaux's Result: Computed Values

Let M be a machine, with

- m constant nodes, $A_1, \ldots, A_m \in \mathbb{R}^m$.
- a bound $t(n)$ on the computation time, for inputs of size n.

Michaux's Result: Computed Values

Let M be a machine, with

- m constant nodes, $A_1, \ldots, A_m \in \mathbb{R}^m$.
- a bound $t(n)$ on the computation time, for inputs of size n.

Lemma:
On any input $x_1, \ldots, x_n \in \mathbb{R}^n$, at any computation step k, any non-empty cell e_l on the work tape holds the evaluation of a rational fraction $f_{l,k} \in \mathbb{Z}(X_1, \ldots, X_{n+m})$ on $(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

$y_{l,k}$ can be represented by $\overline{y_{l,k}} \in \{0,1\}^*$: binary representation of $f_{l,k}$.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

$y_{l,k}$ can be represented by $\overline{y_{l,k}} \in \{0,1\}^*$: binary representation of $f_{l,k}$.

The work tape can be represented by $(\overline{w_l}, \overline{w_r}) \in (\{0,1\}^*)^2$: binary representation of the left and right parts.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

$y_{l,k}$ can be represented by $\overline{y_{l,k}} \in \{0,1\}^*$: binary representation of $f_{l,k}$.

The work tape can be represented by $(\overline{w_l}, \overline{w_r}) \in (\{0,1\}^*)^2$: binary representation of the left and right parts.

The work tape can be represented by $(c_l, c_r) \in \mathbb{R}^2$: numerical values for $(\overline{w_l}, \overline{w_r})$.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

$y_{l,k}$ can be represented by $\overline{y_{l,k}} \in \{0,1\}^*$: binary representation of $f_{l,k}$.

The work tape can be represented by $(\overline{w_l}, \overline{w_r}) \in (\{0,1\}^*)^2$: binary representation of the left and right parts.

The work tape can be represented by $(c_l, c_r) \in \mathbb{R}^2$: numerical values for $(\overline{w_l}, \overline{w_r})$.

Simulation of one arithmetical computation step: symbolic binary computation.

Michaux's Result: Simulation

Step k: $y_{l,k} = f_{l,k}(x_1, \ldots, x_n, A_1, \ldots, A_m)$.

$y_{l,k}$ can be represented by $\overline{y_{l,k}} \in \{0,1\}^*$: binary representation of $f_{l,k}$.

The work tape can be represented by $(\overline{w_l}, \overline{w_r}) \in (\{0,1\}^*)^2$: binary representation of the left and right parts.

The work tape can be represented by $(c_l, c_r) \in \mathbb{R}^2$: numerical values for $(\overline{w_l}, \overline{w_r})$.

Simulation of one arithmetical computation step: symbolic binary computation.

Simulation of one branch step: numerical evaluation of the arguments, and comparison.

Koiran's Weak Model

- Unit measure of time $\rightarrow$ not realistic enough?
- Weak measure of time: a repeated sequence of additions or multiplications has an increasing cost

Koiran's Weak Model

- Unit measure of time $\rightarrow$ not realistic enough?
- Weak measure of time: a repeated sequence of additions or multiplications has an increasing cost

Ex: $y * z$

$$\begin{aligned} y &\rightarrow f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m) \\ z &\rightarrow f_z \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m) \end{aligned}$$

Koiran's Weak Model

- Unit measure of time $\rightarrow$ not realistic enough?
- Weak measure of time: a repeated sequence of additions or multiplications has an increasing cost

Ex: $y * z$

$$\begin{aligned} y &\rightarrow f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m) \\ z &\rightarrow f_z \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m) \end{aligned}$$

Weak cost of $y * z$: max of the degrees and of the coefficient heights of f_y and f_z.

Complexity (4)

Lemma: $L \in \mathsf{P}_W$ if and only if:

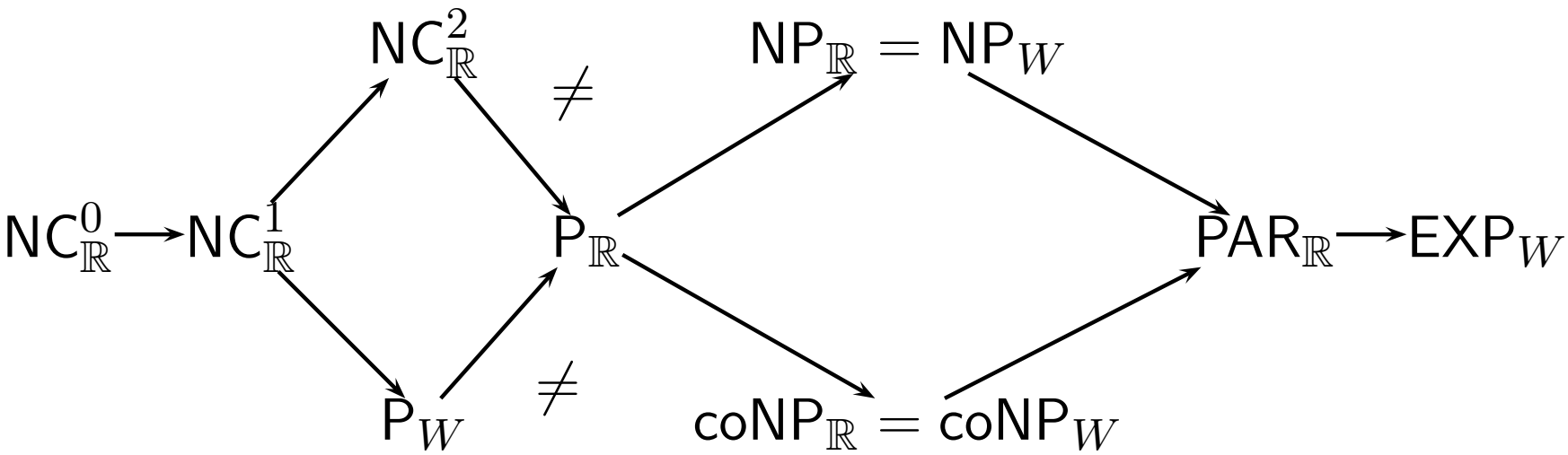
- $L \in \mathsf{P}_{\mathbb{R}}$, and
- Every computed rational fraction has polynomial degree and coefficient heights.

Complexity (4)

Lemma: $L \in \mathsf{P}_W$ if and only if:

- $L \in \mathsf{P}_{\mathbb{R}}$, and
- Every computed rational fraction has polynomial degree and coefficient heights.

Inclusions:



Motivation of this Work (2)

Let M be a boolean algorithm in LOGSPACE
Let M' be a real algorithm, such that:

- M' reads $(x_1, \ldots, x_n) \in \mathbb{R}^n$.
- M' computes $(\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$,

$$\sigma_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- M' applies M on input $(\sigma_1, \ldots, \sigma_n)$

Motivation of this Work (2)

Let M be a boolean algorithm in LOGSPACE
Let M' be a real algorithm, such that:

- M' reads $(x_1, \ldots, x_n) \in \mathbb{R}^n$.
- M' computes $(\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$,

$$\sigma_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- M' applies M on input $(\sigma_1, \ldots, \sigma_n)$

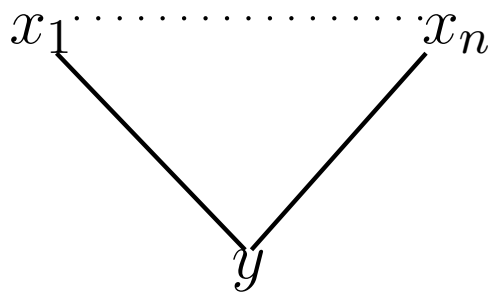
$M' \in \mathsf{NC}^2_{\mathbb{R}} \cap \mathsf{P}_W$
$M' \notin \mathsf{NC}^1_{\mathbb{R}}$?

A Weak Measure of Space

$$y \to f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$$

A Weak Measure of Space

$$y \to f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$$

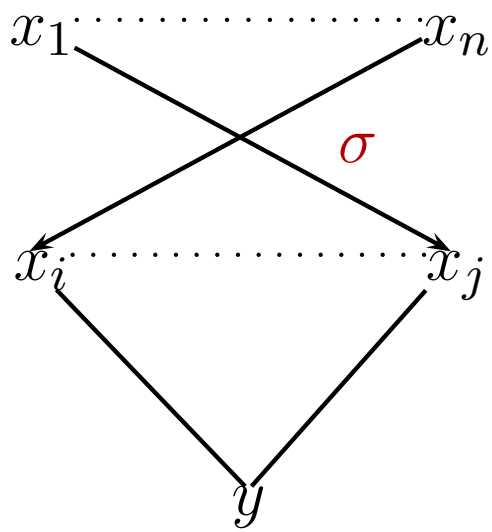


Syntactic Tree T_y of f_y

Original Idea: $|y| \sim |T_y|$.

A Weak Measure of Space

$$y \rightarrow f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$$



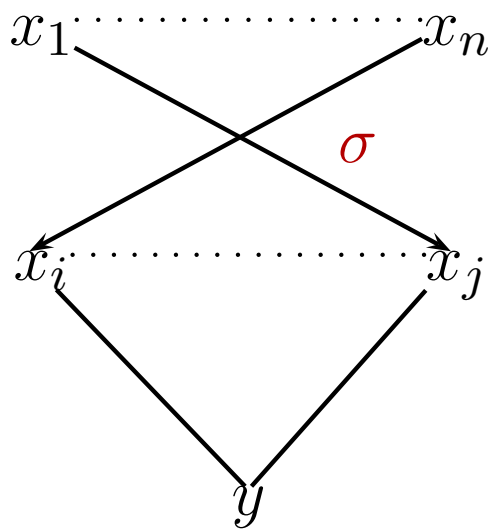
Syntactic Tree T_y of f_y

Original Idea: $|y| \sim |T_y|$.

PB 1: encoding of a permutation $\sigma : \{1, \ldots n\} \rightarrow \{1, \ldots n\}$.

A Weak Measure of Space

$$y \to f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$$



Syntactic Tree T_y of f_y

Original Idea: $|y| \sim |T_y|$.

PB 1: encoding of a permutation $\sigma : \{1, \ldots n\} \to \{1, \ldots n\}$.

PB 2: computation of a minimal tree $\to$ factorization problem.

Weak Size of a Number

$$y \to f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$$

Weak Size of a Number

$y \rightarrow f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$

We restrict ourselves to circular permutations $\sigma : \{1, \ldots n\} \rightarrow \{1, \ldots n\}$. A circular permutation can be represented by an offset of size $\log(n)$.

Weak Size of a Number

$y \rightarrow f_y \in \mathbb{Z}(x_1, \ldots, x_n, A_1, \ldots, A_m)$
We restrict ourselves to circular permutations $\sigma : \{1, \ldots n\} \rightarrow \{1, \ldots n\}$. A circular permutation can be represented by an offset of size $\log(n)$.

$|y|_W$ is the size of a explicit (sequence of monomials) boolean description of f_y, modulo the permutation.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\rightarrow$ additive constant of size $\log(n)$.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\rightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\rightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

- A constant A_i has weak size 1.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\longrightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

- A constant A_i has weak size 1.
- An integer $k \in \mathbb{N}$ has weak size $\log(k)$.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\longrightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

- A constant A_i has weak size 1.
- An integer $k \in \mathbb{N}$ has weak size $\log(k)$.
- The algorithm M' is in $\mathsf{LOGSPACE}_W$.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\longrightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

- A constant A_i has weak size 1.
- An integer $k \in \mathbb{N}$ has weak size $\log(k)$.
- The algorithm M' is in $\mathsf{LOGSPACE}_W$.
- Michaux's simulation of M' is in $\mathsf{LOGSPACE}_W$.

Weak Size of a Configuration

The permutation σ is common for all numbers on the tape $\rightarrow$ additive constant of size $\log(n)$.

The weak size of a configuration is the minimum for all circular permutations of the sum of the weak sizes of the numbers on the tape.

- A constant A_i has weak size 1.
- An integer $k \in \mathbb{N}$ has weak size $\log(k)$.
- The algorithm M' is in $\mathsf{LOGSPACE}_W$.
- Michaux's simulation of M' is in $\mathsf{LOGSPACE}_W$.
- There exists some problems decidable in bounded time, not decidable in constant weak space.

$\mathsf{P}_{\mathbb{R}}$-completeness

Theorem: $\mathsf{RCDP}_{\mathbb{R}}$ (Real Circuit Decision Procedure) is $\mathsf{P}_{\mathbb{R}}$-complete under $\mathsf{LOGSPACE}_W$ reductions.

$\mathsf{P}_{\mathbb{R}}$-completeness

Theorem: $\mathsf{RCDP}_{\mathbb{R}}$ (Real Circuit Decision Procedure) is $\mathsf{P}_{\mathbb{R}}$-complete under $\mathsf{LOGSPACE}_W$ reductions.

Proof: P-completeness of the Boolean Circuit Decision Problem under $\mathsf{LOGSPACE}$-reductions.

Structural Complexity

Theorem:

- $\mathsf{LOGSPACE}_W \subset \mathsf{P}_W \cap \mathsf{NC}^2_{\mathbb{R}}$.
- $\mathsf{PSPACE}_W \subset \mathsf{PAR}_{\mathbb{R}}$.

Structural Complexity

Theorem:

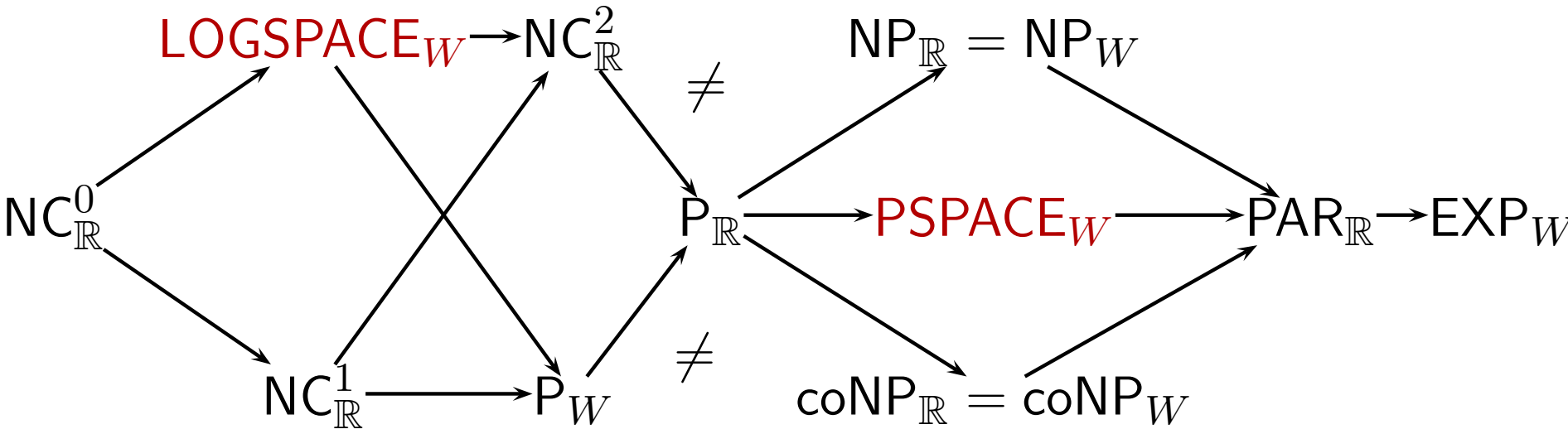
- $\mathsf{LOGSPACE}_W \subset \mathsf{P}_W \cap \mathsf{NC}^2_{\mathbb{R}}$.
- $\mathsf{PSPACE}_W \subset \mathsf{PAR}_{\mathbb{R}}$.

Proof:

- $\mathsf{LOGSPACE}_W \subset \mathsf{P}_{\mathbb{R}}$: enumeration of all configurations.
- $\mathsf{LOGSPACE}_W \subset \mathsf{P}_W$: Koiran's Lemma.
- $\mathsf{LOGSPACE}_W \subset \mathsf{NC}^2_{\mathbb{R}}$: $\mathsf{P}_{\mathbb{R}}$-uniform construction of the configuration graph of a $\mathsf{LOGSPACE}_W$ machine, and graph reachability in NC^2.
- $\mathsf{PSPACE}_W \subset \mathsf{PAR}_{\mathbb{R}}$: Corollary.

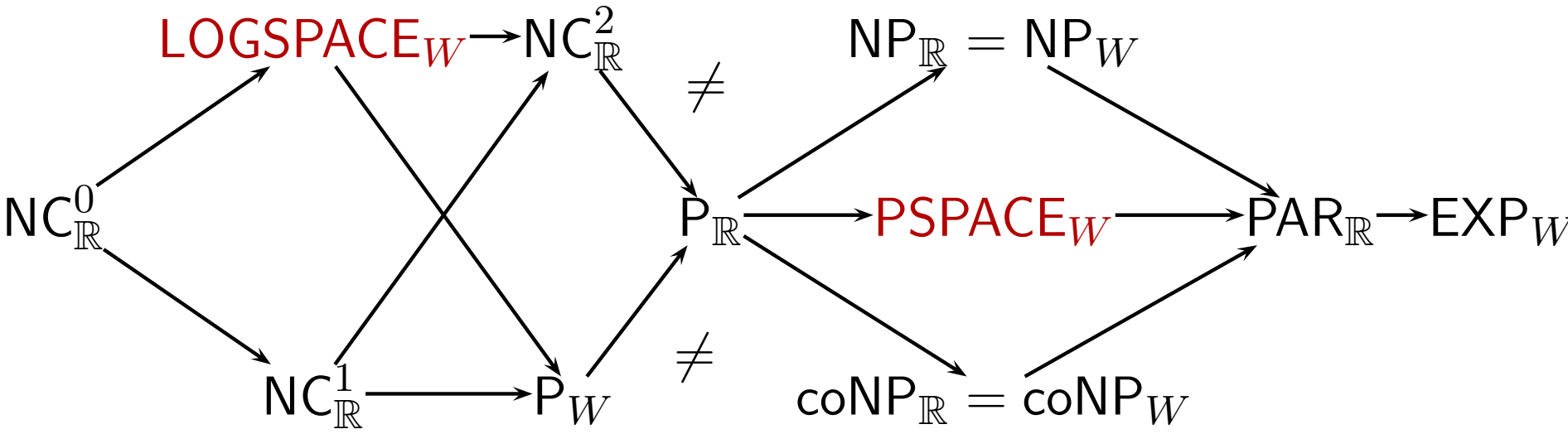
Summary and Open Questions

Inclusions:



Summary and Open Questions

Inclusions:



Conjectures:

- $\mathsf{NC}^1_{\mathbb{R}} \not\subset \mathsf{LOGSPACE}_W$, $\mathsf{LOGSPACE}_W \not\subset \mathsf{NC}^1_{\mathbb{R}}$.
- $\mathsf{PSPACE}_W = \mathsf{PAR}_{\mathbb{R}}$