Institut für theoretische Informatik

Leibniz Universität Hannover

Logical Charactarizations of algebraic circuit classes over rings

Timon Barlag and Sabrina Alexandra Gaube

August 22, 2022

1. Introduction
2. Models and Logics
3. Some Characterizations
4. Outlook

Introduction

- Parallel computation over an integral domain R
  - Algebraic circuits (over R)
    - AC_R^0
- Logics & Descriptive complexity
  - $\mathrm{AC}^0 = \mathrm{FO}$ [Im89]
  - $\mathrm{NC}^1 = \mathrm{FO}[\mathrm{BIT}]+\mathrm{GPR}_{\mathrm{bound}}$ [DHV18]
  - $\mathrm{AC}^1 = \mathrm{FO}[\mathrm{BIT}]+\mathrm{GPR}$ [DHV18]
  - $\mathrm{AC}_{\mathbb{R}}^0 = \mathrm{FO}_{\mathbb{R}}[\mathrm{Arb}_{\mathbb{R}}]$ [BV21]

Basic algebraic definitions

Recall:

A ring (with unity) is a set R equipped with two binary operations $+$ and $\times$, such that

- $(R, +)$ is an abelian group,
- $(R, \times)$ is a monoid,
- in particular, there is a $1 \in R$ such that $r \times 1 = 1 \times r = r$, for all $r \in R$.
- multiplication is distributive with respect to addition

Basic algebraic definitions

Recall:

An integral domain is a nonzero commutative ring without zero divisors, i.e. for every $a, b \neq 0 \in R : a \times b \neq 0$

- every field (e.g. $\mathbb{R}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_p$, for a prime p)
- $\mathbb{Z}$, $\mathbb{R}[x]$, $\mathbb{Z}[i] \ldots$

Algebraic Circuits over R

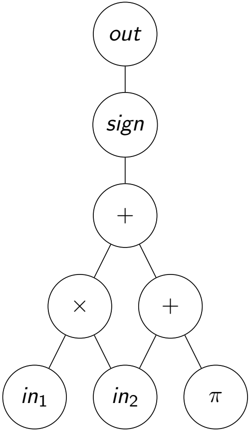
Algebraic Circuits over R

- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)

Algebraic Circuits over R

Algebraic Circuits over R

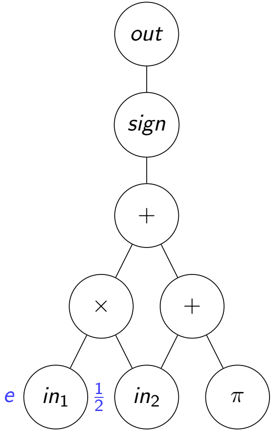
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in $\geq$ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

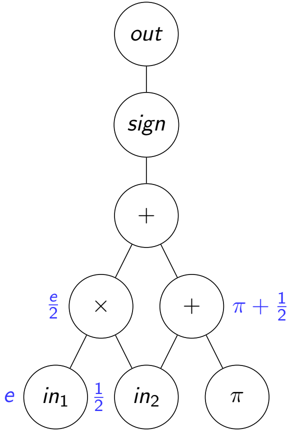
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

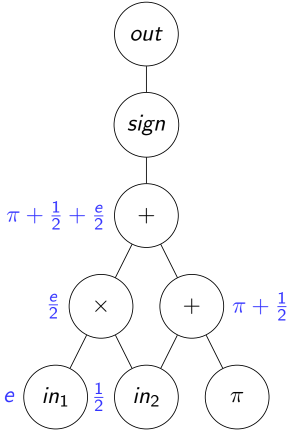
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

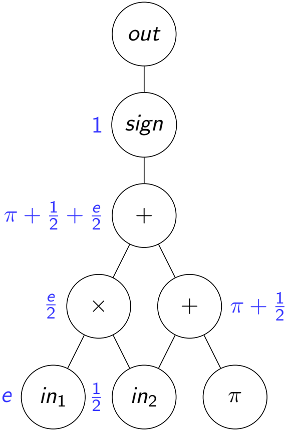
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

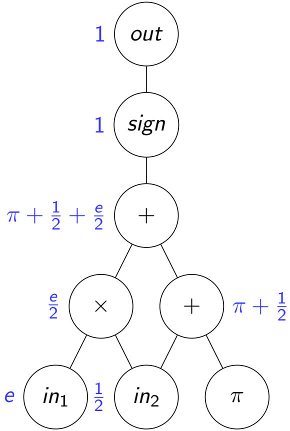
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

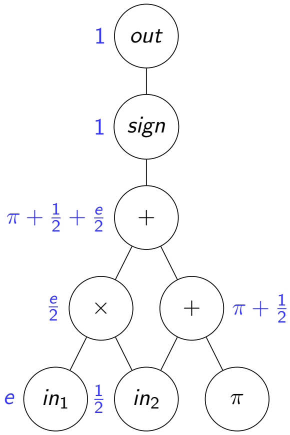
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)



Algebraic Circuits over R

Algebraic Circuits over R

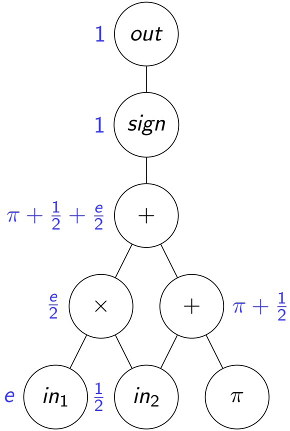
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in $\geq$ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)
- *size*: number of gates
- *depth*: longest path from input to output



Algebraic Circuits over R

Algebraic Circuits over R

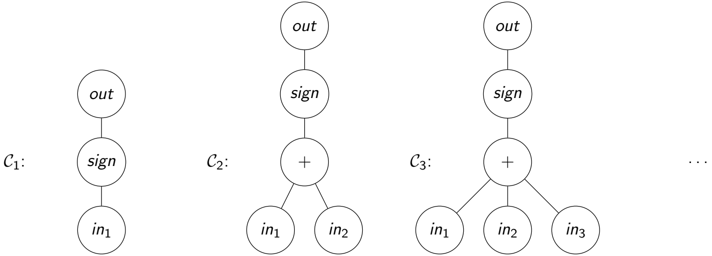
- Directed Acyclic Graph with node types:
  - input (fan-in 0)
  - constant (fan-in 0)
  - arithmetic (fan-in ≥ 0)
  - one of
    - sign (fan-in 1)
    - $<$ (fan-in 2)
  - output (fan-in 1)
- *size*: number of gates
- *depth*: longest path from input to output
- non-uniform!



Circuit families

Circuit families

- Sequences $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2, \ldots)$ of circuits where $\mathcal{C}_i$ has i input gates
- If $\mathcal{C}_i$ computes $f_{\mathcal{C}_i}$ for all $i \in \mathbb{N}$, then $\mathcal{C}$ computes $f_{\mathcal{C}}(w) = f_{\mathcal{C}_{|w|}}(w)$.



Circuit families

Circuit families deciding sets

A circuit family $\mathcal{C}$ decides a set $S \subseteq \bigcup_{n \in \mathbb{N}} R^n$ iff $\mathcal{C}$ computes the characteristic function of S.

Circuit classes

- NC^i_R: sets decided by bounded fan-in circuits families of size $\mathcal{O}(n^{\mathcal{O}(1)})$ and depth in $\mathcal{O}(\log(n)^i)$
- AC^i_R: sets decided by unbounded fan-in circuit families of size $\mathcal{O}(n^{\mathcal{O}(1)})$ and depth in $\mathcal{O}(\log(n)^i)$

Uniformity

Uniform circuit families

- There is an R-machine M producing the circuit family.
  - i.e. M computes the function $n \mapsto \mathcal{C}_n$
- M works in polynomial time $\rightarrow$ P-uniform
  - $\mathrm{U_P}$-C is the subclass of C decided by P-uniform families
- M works in logarithmic time $\rightarrow$ L-uniform
  - $\mathrm{U_{LT}}$-C is the subclass of C decided by L-uniform families

First-order Logic over R

$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

First-order Logic over R

$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

Definition: metafinite R-structures

- R-structure $\mathcal{D} = (\mathcal{A}, \mathcal{F})$ of signature $\sigma = (L_s, L_f)$
  - $\mathcal{A}$: finite structure of L_s with universe A
    - the *skeleton* of $\mathcal{D}$
  - $\mathcal{F}$: finite set of functions $X : A^k \to R$ interpreting symbols in L_f
    - the *arithmetic part* of $\mathcal{D}$

First-order Logic over R

$$\sigma = (\{succ\}, \{val\})$$
$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

Definition: metafinite R-structures

- R-structure $\mathcal{D} = (\mathcal{A}, \mathcal{F})$ of signature $\sigma = (L_s, L_f)$
  - $\mathcal{A}$: finite structure of L_s with universe A
    - the *skeleton* of $\mathcal{D}$
  - $\mathcal{F}$: finite set of functions $X : A^k \to R$ interpreting symbols in L_f
    - the *arithmetic part* of $\mathcal{D}$

First-order Logic over R

$$\sigma = (\{succ\}, \{val\})$$
$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

Definition: FO_R

- formulas and terms over signature $\sigma = (L_s, L_f)$ for variables $x_1, x_2, \ldots$
  - *index terms*: variables, functions $f \in L_s$
  - *number terms*: ring elements, functions $g \in L_f$, $t_1 + t_2$, $t_1 \times t_2$, $sign(t_1)$
  - formulas
    - atomic: $t_1 = t_2$, $t_1 \leq t_2$, predicates $P \in L_s$
    - non-atomic: closure of atomic formulas under Boolean connectives and quantification $(\exists, \forall)$

First-order Logic over R - Example

$$\sigma = (\{succ\}, \{val\})$$
$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \land x_1 = succ(x_2)$$

First-order Logic over R - Example

$$\sigma = (\{succ\}, \{val\})$$

$$\varphi := \exists x_1 \exists x_2 \ val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

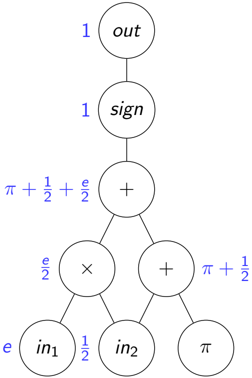
Structure $\mathcal{D}$:

$$A = \{v \mid v \in C\},$$

$$val(v) = \text{the value of } v \text{ on inputs } e \text{ and } \frac{1}{2},$$

$$succ(v) = \text{the successor gate of } v$$

Circuit C:



First-order Logic over R - Example

$$\sigma = (\{succ\}, \{val\})$$

$$\varphi := \exists x_1 \exists x_2 \; val(x_1) = val(x_2) + \pi \wedge x_1 = succ(x_2)$$

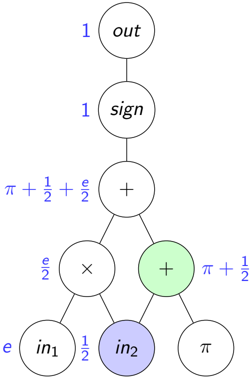
Structure $\mathcal{D}$:

$$A = \{v \mid v \in C\},$$

$$val(v) = \text{the value of } v \text{ on inputs } e \text{ and } \frac{1}{2},$$

$$succ(v) = \text{the successor gate of } v$$

Circuit C:



Extensions to FO_R

Additional functions / relations

- $FO_R[S]$ for a set S of functions and relations

Additional constructions

- the *sum*, *product* and *maximization* rules for creating number terms
  - to use $\sum_{i \in A} t(i)$, $\prod_{i \in A} t(i)$ and $\max_{i \in A}(t(i))$ in formulas

$$\varphi = \exists v_1 \; \underset{v_2}{sum}(g(v_2)) > g(v_1) \times 2$$

$$A = \{\diamondsuit, \spadesuit\}, \; g = \{\diamondsuit \mapsto \pi, \spadesuit \mapsto 42\}$$

Extensions to FO_R

Additional functions / relations

- $FO_R[S]$ for a set S of functions and relations

Additional constructions

- the *sum*, *product* and *maximization* rules for creating number terms
  - to use $\sum_{i \in A} t(i)$, $\prod_{i \in A} t(i)$ and $\max_{i \in A}(t(i))$ in formulas

$$\varphi = \exists v_1 \; \overbrace{\underset{v_2}{sum}(g(v_2))}^{42+\pi} > g(v_1) \times 2$$

$$A = \{\diamondsuit, \spadesuit\}, \; g = \{\diamondsuit \mapsto \pi, \spadesuit \mapsto 42\}$$

Logical Characterizations

Non-uniform AC^0_R

$$\mathrm{AC}^0_R = \mathsf{FO}_R[\mathsf{Arb}_R]$$

Polynomial-time uniform AC^0_R

$$\mathrm{U_P}\text{-}\mathrm{AC}^0_R = \mathsf{FO}_R[\mathsf{FTIME}_R(n^{\mathcal{O}(1)})]$$

Logarithmic-time uniform AC^0_R

$$\mathrm{U_{LT}}\text{-}\mathrm{AC}^0_R = \mathsf{FO}_R[\mathsf{FTIME}_R(\log(n))] + \mathsf{SUM}_R + \mathsf{PROD}_R$$

Guarded predicative/functional recursion

Definition: [Durand, Haak, Vollmer, 2018]

A formula φ is in $\mathrm{FO} + \mathrm{GPR}^1$ if it has the form

$$\varphi ::= [P(\overline{x}, \overline{y}) \equiv \theta(\overline{x}, \overline{y}, P)]\varphi(P) | \psi,$$

where ψ and θ are FO formulae with free variables $\overline{x}, \overline{y}$ such that each atomic sub-formula involving the symbol P

1. is of the form $P(\overline{x}, \overline{z})$, where $\overline{z}$ is in the scope of a guarded quantification $Q\overline{z}.((\overline{z} \leq \overline{y}/2) \wedge \xi(\overline{y}, \overline{z}))$ with $Q \in \{\forall, \exists\}$, $\xi \in \mathrm{FO}$ and
2. never occur in the scope of any quantification not guarded this way.

Results

Former results [DHV, 2018]

1. $\mathrm{NC}^1 = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1_{\mathsf{bound}}$
2. $\mathrm{SAC}^1 = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1_{\mathsf{semi}}$
3. $\mathrm{AC}^1 = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1$
4. $\#\mathrm{AC}^0 = \#\mathsf{Win}\text{-}\mathrm{FO}[\mathrm{BIT}]$
5. $\#\mathrm{NC}^1 = \#\mathsf{Win}\text{-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1_{\mathsf{bound}}$
6. $\#\mathrm{SAC}^1 = \#\mathsf{Win}\text{-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1_{\mathsf{semi}}$
7. $\#\mathrm{AC}^1 = \#\mathsf{Win}\text{-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^1$

Guarded predicative logic - ideas of adaptations

ideas

- Extend GPR in order to characterize the whole AC, NC and SAC -hierachies
  - Substitute the factor $\frac{1}{2}$ by $2^{\log_2(n)/\log_{2^i}(n)}$ for AC^i etc.
    - But exponantiation is not part of the logic

  $\Rightarrow$ GPR^i_R by tuples
- Add a similar construction to logics over metafinite structures
  - probably functional instead of predicative recursion

Further results

Goal

1. $\mathrm{NC}^i = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i_{\text{bound}}$
2. $\mathrm{SAC}^i = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i_{\text{semi}}$
3. $\mathrm{AC}^i = \mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i$
4. $\#\mathrm{NC}^i = \#\text{Win-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i_{\text{bound}}$
5. $\#\mathrm{SAC}^i = \#\text{Win-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i_{\text{semi}}$
6. $\#\mathrm{AC}^i = \#\text{Win-}\mathrm{FO}[\mathrm{BIT}] + \mathrm{GPR}^i$
7. $\mathrm{NC}^i_R = \mathrm{FO}_R[\mathrm{Arb}_R] + \mathrm{GPR}^i_{R,\text{bound}}$
8. $\mathrm{AC}^i_R = \mathrm{FO}_R[\mathrm{Arb}_R] + \mathrm{GPR}^i_R$

Relationship between versions of AC^0_R over different rings – canonical maps

Definition:

A canonical ring map from a ring R_1 to a ring R_2 is a fixed injective function $f_{R_1,R_2}\colon R_1 \to R_2^k$ for some $k \in \mathbb{N}$.
f_{R_1,R_1} is the identity function.

Definition:

For two complexity classes $\mathcal{C}_{R_1}, \mathcal{C}_{R_2}$ over the respective rings R_1, R_2, we write

$$\mathcal{C}_{R_1} \subseteq_c \mathcal{C}_{R_2},$$

if for all languages $A \in \mathcal{C}_{R_1}$ the following holds:

$$\{f_{R_1,R_2}(x) \mid x \in A\} \in \mathcal{C}_{R_2^*}.$$

The relations $=_c$ and $\subset_c$ are defined analogously.

special cases of AC^0_R

Theorem

- $\mathrm{AC}^0_{\mathbb{Q}} =_c \mathrm{AC}^0_{\mathbb{Z}} =_c \mathrm{AC}^0_{\mathbb{Z}[i]} =_c \mathrm{AC}^0_{\mathbb{Z}[x]} =_c \mathrm{AC}^0_{\mathbb{Z}[x,y]}$
- $\mathrm{AC}^0_{\mathbb{C}} =_c \mathrm{AC}^0_{\mathbb{R}[i]} =_c \mathrm{AC}^0_{\mathbb{R}} =_c \mathrm{AC}^0_{\mathbb{R}[x]} =_c \mathrm{AC}^0_{\mathbb{R}[x,y]}$
- $\mathrm{AC}^0_{\mathbb{Z}} \subset_c \mathrm{AC}^0_{\mathbb{R}}$

idea:

Write a number $z = \frac{a}{b} \in \mathbb{Q}$ (resp. $z = a + bi \in \mathbb{C}$) as tuple (a, b) and use a node per tuple.

Future/Current Research

- How can these classes be contextualized?
  - separate AC^0_R and NC^1_R?
- Can we find a meaningful analogue of TC_R?
  - investigate then in particular the question, whether $\mathrm{TC}^0_R = \mathrm{NC}^1_R$?
- How are $\mathrm{AC}^0_{\mathbb{F}_p}$ and $\mathrm{AC}^0_{\mathbb{F}_q}$ for $p \neq q$ (prime?) related?
- Define SAC^i_R classes
  - Is it important which gate type we bound?
- Is there a connection between GPR-variations and fixed point logics like $\mathrm{FO(LFP)}$?

Sources

[Im89] Neil Immerman. Expressibility and Parallel Complexity. SIAM J. Comput. 18(3), 625-638 (1989)

[CM99] Felipe Cucker and Klaus Meer. Logics which capture complexity classes over the reals. J. Symb. Log., 64(1):363-390, 1999.

[BCSS98] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. Complexity and Real Computation. Springer-Verlag, New York, 1998.

[DHV18] Arnaud Durand, Anselm Haak, and Heribert Vollmer. Model-Theoretic Characterization of Boolean and Arithmetic Circuit Classes of Small Depth. LICS 2018: 354-363.

[BV21] Timon Barlag and Heribert Vollmer. A Logical Characterization of Constant-Depth Circuits over the Reals. WoLLIC 2021: 16-30

Definition (GPR^i_R)

For $i \geq 0$, a formula φ is in $\mathrm{FO}_R(\mathrm{GPR}^i_R)$ if it has the form

$$\varphi ::= [P(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}) \equiv \theta(\overline{x}, \overline{y_1}, \ldots, \overline{y_i}, P)]\varphi(P)]|\psi,$$

where ψ and θ are FO_Rformulae with free variables $\overline{x}, \overline{y_1}, \ldots, \overline{y_i}$ such that each atomic sub-formula involving the symbol P

1. is of the form $P(\overline{x}, \overline{y_1}, \ldots, \overline{y_i})$, the $\overline{y_1}, \ldots, \overline{y_i}$ are in the scope of a guarded aggregation

$$A_{\overline{y_1}, \ldots, \overline{y_i}}.(\bigvee_{j=i}^{1} \overline{z_j} \leq \overline{y_j}/2 \wedge \bigwedge_{k=j-1}^{1} \overline{z_k} \leq \overline{y_k} \wedge \xi(\overline{y_1}, \ldots, \overline{y_i}, \overline{z_1}, \ldots, \overline{z_i}))$$

   with $A \in \{max, sum, prod\}$ and
2. never occur in the scope of any aggregation (or quantification) not guarded this way.