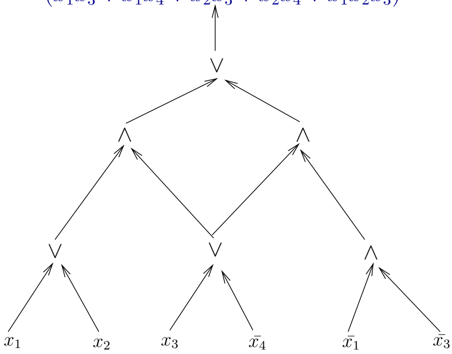
Small Width Arithmetic Circuits

Meena Mahajan.
joint work with
Raghavendra Rao B.V

October 1, 2008

Boolean Circuits

$(x_1x_3 \vee x_1\bar{x_4} \vee x_2x_3 \vee x_2\bar{x_4} \vee \bar{x_1}\bar{x_2}\bar{x_3})$



Boolean Circuit: Formal definition

- Directed acyclic graph
- Internal nodes labeled with $\{\vee, \wedge, \neg\}$.
- Leaves labeled with $\{0, 1, x_1, \ldots, x_n\}$.
- A designated output node, of out-degree zero.
- Circuit inputs $x_i \in \{0, 1\}$

Resource Measures:

- fan in (fan out) of a node: its in-degree (out-degree)
- size – number of internal nodes
- depth – length of longest path from output node to input node
- width – maximum number of nodes at any particular level

Polynomial-size circuits and parallelizability

P: Polynomial-size uniform circuits.

NC: Polynomial-size Poly-logarithmic depth circuits.
short, fat
(named after Nicholas Pippenger.)
algorithms implementable in parallel polylog time.
NC^i: polynomial size $O((\log n)^i)$ depth.

NC^1: parity of n bits,
sorting n numbers,
evaluating a boolean formula,
membership in any fixed regular language.

NC^2: computing the determinant of an integer matrix,
membership in fixed context-free language (CFL).

Polynomial-size circuits and space-efficiency

P: Polynomial-size uniform circuits.

SC: Polynomial-size Poly-logarithmic width circuits.
tall, skinny
(named after Steve Cook).
algorithms needing poly time and polylog space.
SC^i: polynomial size $O((\log n)^i)$ width.

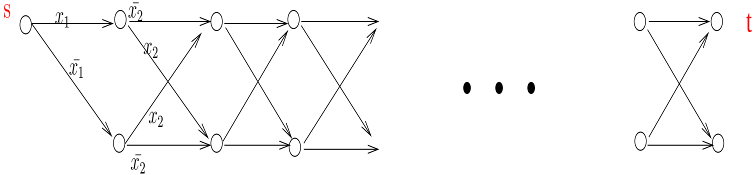
SC^0: Known to be same as NC^1; hence parity of n bits, sorting n numbers, evaluating a boolean formula, membership in any fixed regular language.

SC^1: Known to be same as log-space; hence undirected graph connectivity, planarity testing, isomorphism testing for trees and planar graphs.

SC^2: membership in any fixed deterministic CFL, (PLoSS) any randomized logspace algorithm.

Branching programs

A width-2 branching program for parity



BWBP: Bounded Width Branching Programs (of poly size).
Known to be same as NC^1.

BP: Poly size Branching Programs.
Known to be same as nondeterministic logspace.

Formulae

- Formula: A circuit where every node has out-degree at most 1. (The underlying graph is a forest.)
- Every circuit C has an equivalent formula F of the same depth, but F may be much (exponentially) bigger.
- NC^1 circuits have equivalent poly-size log-depth formulae.
- NC^1 circuits also have equivalent poly-size log-width formulae.
- Every poly-size formula has an equivalent NC^1 circuit.

A set of Equivalences

- $BWBP \subseteq NC^1$ (divide-and-conquer *a la* Savitch)
- $BWBP \subseteq SC^0$ (folklore)
- $SC^0 \subseteq NC^1 \subseteq BWBP$ (Barrington)
- $NC^1 \subseteq F$ (folklore)
- $LWF \subseteq F \subseteq NC^1$ (Spira)
- $NC^1 \subseteq LWF$ (Istrail,Zilkovich)

Thus NC^1, BWBP, SC^0, F, LWF are all equivalent.

Counting Classes

Arithmetizing a Boolean circuit:

- Move all negations to the leaves. (de Morgan's laws)
- Replace
  - every $\wedge$ gate by a $\times$ gate;
  - every $\vee$ gate by a $+$ gate;
  - leaf-level negation $\overline{x_i}$ by $1 - x_i$.

An arithmetic circuit

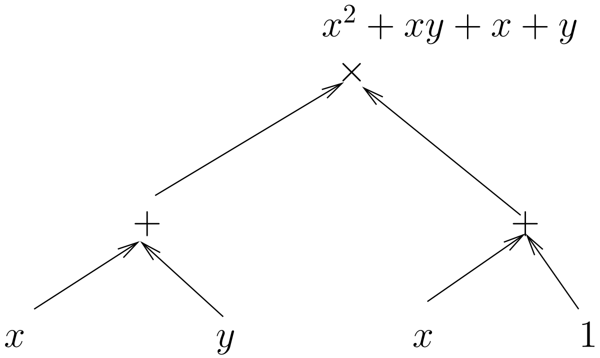


Figure: An arithmetic circuit, computing the polynomial $x^2 + xy + x + y$

Arithmetic Circuit Classes

Arithmetic Circuits: Similar to counting classes.

- Computation over arbitrary rings $\mathbb{K}$.
- Internal nodes labeled $\times$ or $+$.
- Leaves labeled by 0, 1, -1, or x_i for $i \in \{1, \ldots, n\}$.
- x_i can take any value in $\mathbb{K}$.

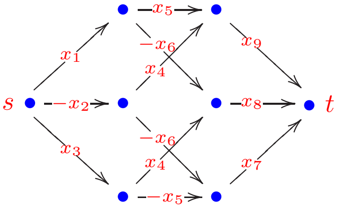
Arithmetic Branching Program

- Edges of the BP labeled by $0, 1, -1$ or x_i for $i \in \{1, \ldots, n\}$.
- Weight of a path: product of weights of labels of edges on the path
- Function computed: sum of weights of all $s \longrightarrow t$ paths.

Arithmetic Branching Program: An Example

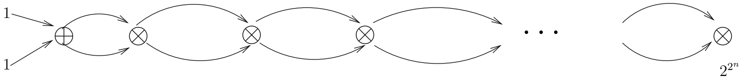
An arithmetic branching program to compute the determinant of the matrix

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix}$$



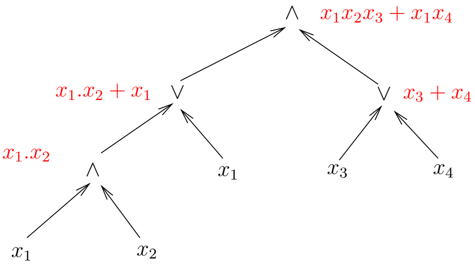
Relationships among arithmetic classes?

- Not all equivalences carry over to arithmetizations.
- In particular, a-SC^0 seems too powerful:
- A width two circuit can compute super-exponential values requiring super-polynomial bits in a binary representation:



- We need a resource measure that restricts circuit output to feasible values.

Degree of a Circuit: an Example



Degree of a Circuit

- Circuit degree: roughly speaking, algebraic degree of associated polynomial.
- Caveats:
  1. constants $(0, 1, -1)$ at leaves are replaced by new variables.
  2. cancellations not accounted for.
     degree of $(x_1x_2 + x_3) + (x_4 - x_1x_2)$ is 2, not 1.
- Recursive definition:
  - degree of leaf $= 1$,
  - degree of $\vee$ or $+$ gate $=$ max degree of children,
  - degree of $\wedge$ or $\times$ gate $=$ sum of degrees of children.

Restricting the degree

- Define small SC, denoted sSC:
  sSC^i = SC^i circuit of polynomial degree
- sSC is in NC; whatever the width. (Venkateswaran)
  (In fact, any poly-size poly-degree circuit has an equivalent circuit in NC.)
- BWBP $\subseteq$ sSC^0 $\subseteq$ SC^0 $\subseteq$ BWBP
  i.e. degree bound not a restriction for SC^0.
- sSC^0 $\subseteq$ sSC^1 $\subseteq$ SC^1
  i.e. sSC^1 is sandwiched between NC^1 and L.

Relating poly degree classes

- Boolean: sSC^0 $=$ NC^1
- Arithmetic: a-NC^1 $=$ a-BWBP $\subseteq$ a-sSC^0
  Ben-Or, Cleve; Caussinius, McKenzie, Therien, Vollmer
- Arithmetic: a-F $=$ a-NC^1 Brent
- Open: Is a-sSC^0 contained in a-NC^1?
  That is, can tall skinny circuits be converted to equivalent short fat ones?
  Can we perform *depth-reduction*?
- In what follows: a restricted setting where we can ...

Multilinear Circuits

- A polynomial $f \in \mathbb{K}[x_1, \ldots, x_n]$ is multilinear if the degree of each variable is bounded by 1.
- A circuit is multilinear if every gate computes a multilinear polynomial.
- In a syntactic multilinear circuit, the left child and right child of every $\times$ gates contain disjoint sets of variables.

Remark: All multilinear formulae have equivalent syntactic multilinear formulae, though the construction is non-uniform.

$$\text{ma-F} = \text{sma-F}$$

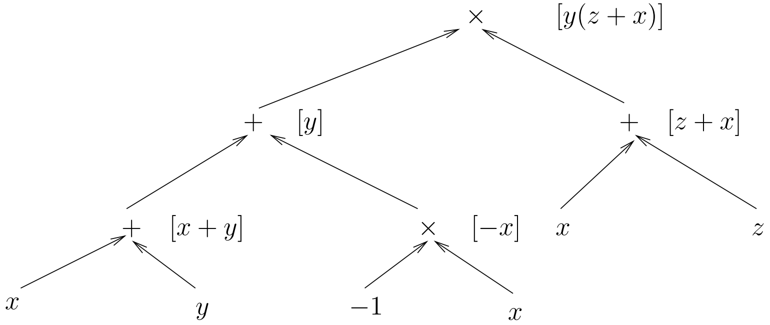


Figure: A *multilinear circuit*, which is not *syntactic multilinear*

Why Syntactic Multilinear ?

- Any syntactic multilinear formula for computing the Permanent or the Determinant requires super-polynomial size [Raz 2004]
- Syntactic multilinear circuits are strictly more powerful than syntactic multilinear formula; sma-F $\subset$ sma-NC^2. [Raz 2004]

Question 1: What is the relationship among the Syntactic Multilinear arithmetic circuits around NC^1?
sma-BWBP $\subseteq$ sma-sSC^0.
sma-BWBP $\subseteq$ sma-NC^1 $=$ sma-F.
Question 2: Can width bounded syntactic multilinear circuits be depth reduced?
Is sma-sSC^0 in sma-NC^1?

Main theorem

Theorem

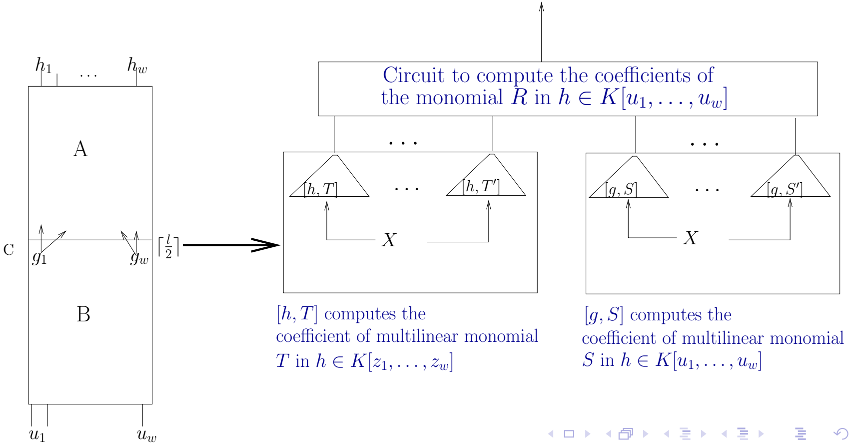
For any syntactic multilinear arithmetic circuit of width w, depth l and degree d, there is an equivalent bounded fan-in arithmetic circuit of depth $O(w(\log l + \log d))$ and size $O((ld)^w)$

In English: syntactic multilinear tall thin circuits can be depth-reduced.

Note: The depth-reduced circuit need not be syntactic multilinear.

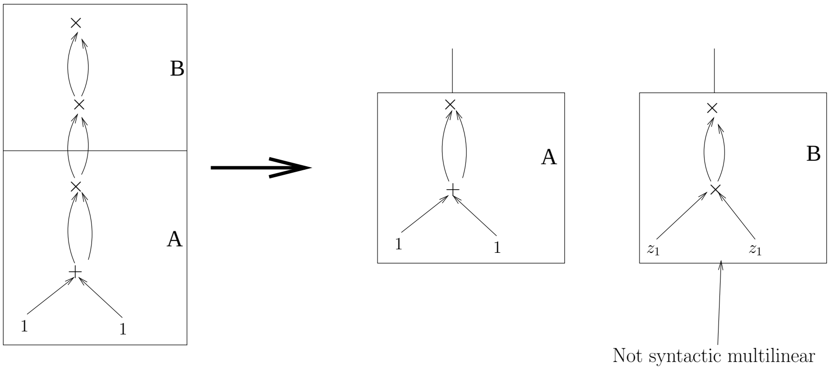
Proof sketch

Idea: Divide and conquer



Proof sketch cont'd...

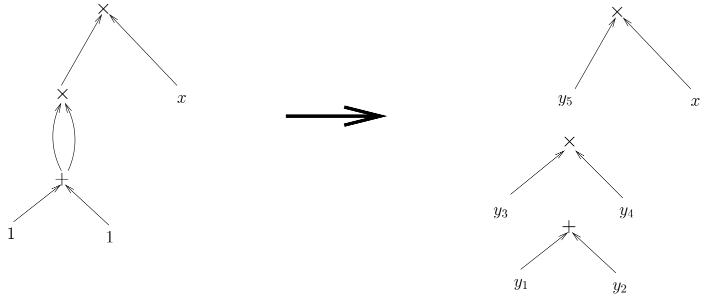
A mind block



Proof sketch cont'd...

- Introduce new variables for each of the wires carrying only constants.

Proof sketch cont'd...



Proof sketch cont'd...

- Introduce new variables for each of the wires carrying only constants.
- The resulting circuit is *syntactic multilinear* in $X \cup Y$

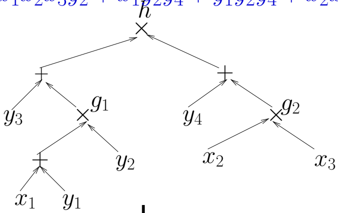
Proof sketch cont'd...

Steps:

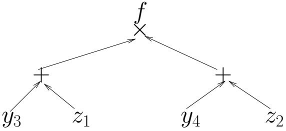
- Break the circuit at depth $l/2$ into sub-circuits A, B

Proof sketch cont'd ...

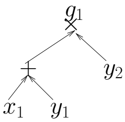
$p_h = x_1x_2x_3y_2 + x_1y_2y_4 + y_1y_2y_4 + x_2x_3y_3 + y_3y_4$



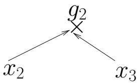
$p_f(z_1, z_2) = y_3y_4 + z_1y_4 + z_2y_3 + z_1z_2$



$p_{g_1} = x_1y_2 + y_1y_2$



$p_{g_2} = x_2x_3$

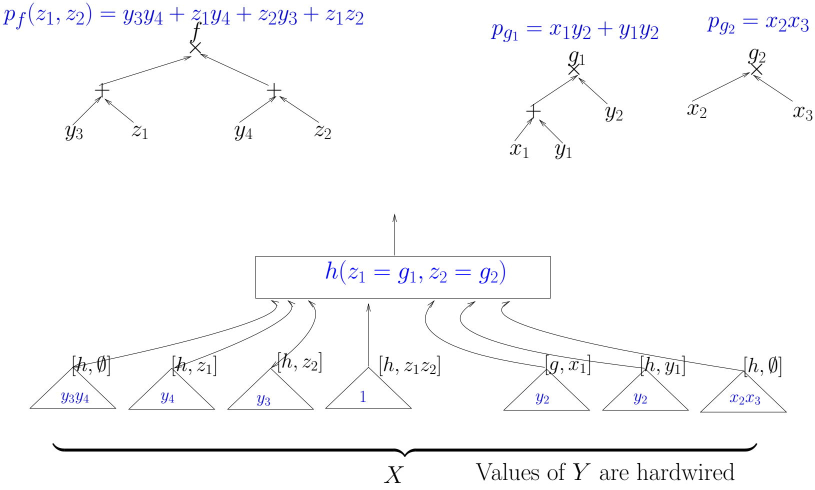


Proof sketch cont'd...

Steps:

- Break the circuit at depth $l/2$ into sub-circuits A, B
- Inductively build circuits which compute the coefficients of polynomials computed by A and B

Proof sketch cont'd ...



Main theorem

Hence we have,

Theorem

For any syntactic multilinear arithmetic circuit of width w, depth l and degree d, there is an equivalent bounded fan-in arithmetic circuit of depth $O(w(\log l + \log d))$ and size $O((ld)^w)$

Corollary: sma-sSC^0 $\subseteq$ a-NC^1.
poly-size constant width $\longrightarrow$ poly-size log depth

Paradoxical Improvement

A recent modification by Jansen makes the depth-reduced circuit also syntactic multilinear; i.e. sma-sSC^0 $\subseteq$ sma-NC^1.

This is unexpected, because without the sma- restriction, not only is such a containment open, but in fact exactly the converse containment is known to hold; a-NC^1 $\subseteq$ a-sSC^0.

Open Questions

- Depth reduction for general constant width arithmetic circuits?
- Can the constructions be made uniform?
- Can we separate sma-BWBP from sma-BP?

Thank You