Boolean and arithmetic circuit complexity

Asa Goodwillie, Amherst College

Advisor: Dr. Eric Allender, Department of Computer Science, Rutgers University

July 17 2015

Context and motivation

(Computational) complexity theory

- Some (hopefully!) familiar classes
  - P: problems solvable in polynomial time
  - NP: problems solvable in nondeterministic polynomial time / checkable in polynomial time
  - Part of large hierarchy:

$$\mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXPTIME} \subseteq \cdots$$

- We say a problem is *complete* for a class if it is "at least as hard" (in an appropriate sense) as every problem in that class
  - A problem A is "at least as hard" as another problem B if we can "easily" reduce B to A
  - e.g., NP-complete problems include 3SAT, traveling salesman, and many more

Context and motivation, cont.

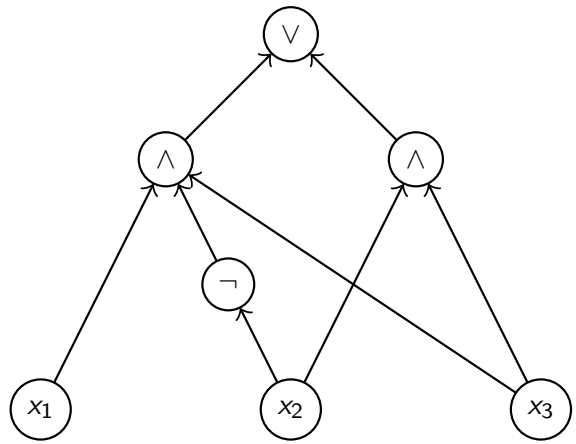
Complexity below P

- We want to define subclasses of P to understand its internal structure
- We use circuits to define many of these subclasses
- Those subclasses and their complete problems (for an appropriate notion of "at least as hard") divide P up more finely

Basics about circuits

What is a circuit?

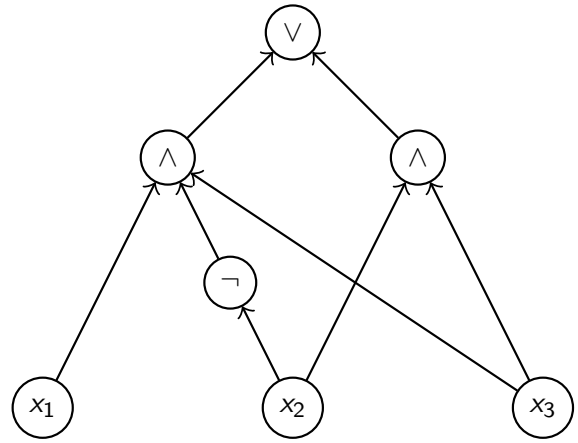
- A (Boolean) circuit looks like this:



- Given Boolean input values, place them on the inputs (x_1, x_2, x_3) and see what the output of the circuit is

Circuit complexity

- Circuits output 0/1 values, so we can use circuits (one for each input length) to recognize a language
- Circuits have size and depth, e.g.



- For today: polynomial size, logarithmic depth

Some examples of Boolean circuit classes

Definition

AC^1: *languages recognized by poly. size, log. depth circuits using AND, OR, and NOT gates.*

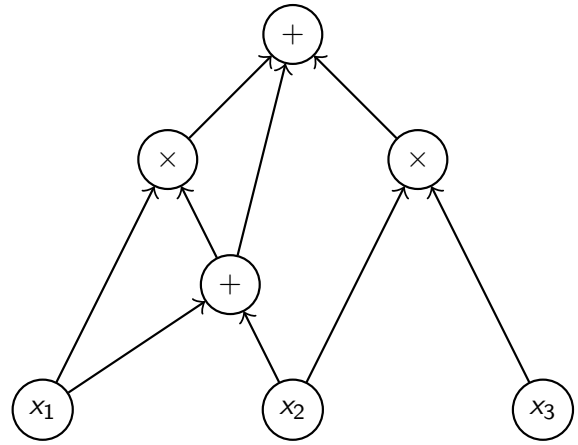
$\mathrm{AC}^1[m]$: *languages recognized by poly. size, log. depth circuits using AND, OR, NOT, and* MOD_m *gates.*

2-$\mathrm{AC}^1[m]$: *languages recognized by poly. size, log. depth circuits using AND, OR, NOT, and* MOD_m *gates, where the AND and OR gates have fan-in 2.*

Arithmetic circuit classes

Arithmetic circuits

- Nodes compute arithmetic operations $(+, \times)$ over some algebraic structure (rather than Boolean operations), e.g.



- Values carried by "wires" in circuit are elements of the algebraic structure, not Boolean (0/1) values
- We restrict the final output to be 0 or 1 in order to define languages using these circuits

Arithmetic circuit classes

- Again, we consider families of circuits (one for each input length) and impose size and depth restrictions
- We also impose restrictions on fan-in (number of inputs)

Definition

Let R be a (semi)ring.

$\mathrm{VP}(R)$*: languages recognized by poly. size, log. depth arithmetic circuits over R using unbounded fan-in $+$ gates and $\times$ gates of fan-in 2.*

$\Lambda\mathrm{P}(R)$*: languages recognized by poly. size, log. depth arithmetic circuits over R using unbounded fan-in $\times$ gates and $+$ gates of fan-in 2.*

One goal is to establish connections between Boolean and arithmetic circuit classes. (In particular, Boolean characterizations of $\mathrm{VP}(R)$ and $\Lambda\mathrm{P}(R)$ for various choices of R.)

Results (previous and new)

Earlier this year, Dr. Allender (and others) established results for VP and ΛP over finite fields.

Theorem
(Allender, Gál, and Mertz, 2015[1]) $\mathrm{VP}(\mathbb{F}_{p^k}) = \text{2-}\mathrm{AC}^1[p]$ *and* $\Lambda\mathrm{P}(\mathbb{F}_{p^k}) = \mathrm{AC}^1[p^k - 1]$.

My results concern $\mathrm{VP}(\mathbb{Z}_m)$ and $\Lambda\mathrm{P}(\mathbb{Z}_m)$.

[1] "Dual VP Classes", Eric Allender, Anna Gál, and Ian Mertz, 2015.

Arithmetic circuits over $\mathbb{Z}_m$: the prime power case

If p is prime, then $\mathbb{Z}_p$ is equal to $\mathbb{F}_p$, so (by prev. thm.)
$\mathsf{VP}(\mathbb{Z}_p) = \text{2-}\mathsf{AC}^1[p]$ and $\Lambda\mathsf{P}(\mathbb{Z}_p) = \mathsf{AC}^1[p-1]$.

For $k > 1$, $\mathbb{Z}_{p^k}$ is *not* equal to $\mathbb{F}_{p^k}$. It turns out that the arithmetic classes over $\mathbb{Z}_{p^k}$ are the same as those over $\mathbb{Z}_p$.

Theorem

$\mathsf{VP}(\mathbb{Z}_{p^k}) = \mathsf{VP}(\mathbb{Z}_p) = \text{2-}\mathsf{AC}^1[p]$ *and*
$\Lambda\mathsf{P}(\mathbb{Z}_{p^k}) = \Lambda\mathsf{P}(\mathbb{Z}_p) = \mathsf{AC}^1[p-1]$.

Arithmetic circuits over $\mathbb{Z}_m$: the general case

The following result describes VP and ΛP over product rings.

Theorem

Let R and S be rings. Then $\mathrm{VP}(R \times S) = \mathrm{VP}(R) \cap \mathrm{VP}(S)$ *and* $\Lambda\mathrm{P}(R \times S) = \Lambda\mathrm{P}(R) \cap \Lambda\mathrm{P}(S)$.

This lets us describe VP and ΛP over $\mathbb{Z}_m$ for a general modulus m in terms of the prime power case.

Corollary

Let $m = p_1^{k_1} \cdots p_l^{k_l}$ be the prime factorization of $m > 1$. Then $\mathbb{Z}_m = \mathbb{Z}_{p_1^{k_1}} \times \cdots \times \mathbb{Z}_{p_l^{k_l}}$, *so*

$$\begin{aligned}
\mathrm{VP}(\mathbb{Z}_m) &= \mathrm{VP}(\mathbb{Z}_{p_1^{k_1}}) \cap \cdots \cap \mathrm{VP}(\mathbb{Z}_{p_l^{k_l}}) \\
&= \text{2-AC}^1[p_1] \cap \cdots \cap \text{2-AC}^1[p_l] \text{ and} \\
\Lambda\mathrm{P}(\mathbb{Z}_m) &= \Lambda\mathrm{P}(\mathbb{Z}_{p_1^{k_1}}) \cap \cdots \cap \Lambda\mathrm{P}(\mathbb{Z}_{p_l^{k_l}}) \\
&= \mathrm{AC}^1[p_1 - 1] \cap \cdots \cap \mathrm{AC}^1[p_l - 1].
\end{aligned}$$

In other words...

We can express VP and ΛP over the ring of integers mod m as the intersection of a few familiar Boolean circuit classes. For example:

- $6 = 2 \times 3$, so $\mathbb{Z}_6 = \mathbb{Z}_2 \times \mathbb{Z}_3$, so
  - $\mathrm{VP}(\mathbb{Z}_6) = \text{2-AC}^1[2] \cap \text{2-AC}^1[3]$ and
  - $\Lambda\mathrm{P}(\mathbb{Z}_6) = \mathrm{AC}^1 \cap \mathrm{AC}^1[3] = \mathrm{AC}^1$.
- $77 = 7 \times 11$, so $\mathbb{Z}_{77} = \mathbb{Z}_7 \times \mathbb{Z}_{11}$, so
  - $\mathrm{VP}(\mathbb{Z}_{77}) = \text{2-AC}^1[7] \cap \text{2-AC}^1[11]$ and
  - $\Lambda\mathrm{P}(\mathbb{Z}_{77}) = \mathrm{AC}^1[6] \cap \mathrm{AC}^1[10]$.

Many thanks to...

- Dr. Allender, for all of his help and guidance;
- DIMACS, for supporting this project;
- and you all, for listening!

Any questions?