Complexity Reading Group 11/19/2015

Brown University

1 Algebraic Computation Models

Three things to discuss:

- Algebraic Circuits
- Computation Trees
- Blum-Shub-Smale Machines

Definition 1: A *Field* is:

- A set F
- The two operators, $+$ and $\times$
- All elements in the set have an additive and multiplicative inverse, so implicitly $-$ and $\div$ are included as well.

Examples: $\mathbb{R}, \mathbb{C}, \mathbb{Q}$.

1.1 Algebraic Circuits

Recall: boolean circuits have $\wedge, \vee, \neg$ gates.

Instead, we now have $+, \times$ gates. Sometimes also allowed the constants 1 and -1, and $\div$.

A circuit is an algebraic circuit if it has one output.

Define polynomials in many variables $f(x_1, \ldots, x_n)$. If $\div$ is allowed, get rational functions $\frac{f}{g}$.

Example: the determinant of a matrix:

$X \in M_n(\mathbb{F})$, $n \times n$ matrix s.t. the elements of M are in the field $\mathbb{F}$.

$$det(x) = \sum_{\sigma \in S_n} (-1)^{sgn(\sigma)} \prod_{i=1}^{n} x_{i\sigma(i)} \tag{1}$$

In general, determinant defined by a polynomial of $n!$ length.

Can compute the determinant via an algebraic circuit of size $\mathcal{O}(n^3)$. Also, there is an NC^2 algorithm for computing determinant, gives algorithm form of size $2^{\mathcal{O}(\log^2 n)}$

Example: the permanent of a matrix:

Given $X \in M_n(\mathbb{F})$:

$$perm(x) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)} \tag{2}$$

There is *no known poly size circuit to compute permanent.* Note: it's $\#P$-complete[1].

---

2 Complexity

Some definitions:

Definition 3: If $\{f_i\}$ is a set of polynomials in n variables of a field $\mathbb{F}$, then this set has poly-bounded degree if there is a $c \in \mathbb{N}$ s.t. each f_i has degree at most $\mathcal{O}(n^c)$.

Definition 4: The class $AlgP$ is a set of polynomials of poly degree that are computable by algebraic formulas of polynomial size

Definition 5: $AlgNP$ is the the class of polynomials:

$$f(x_1, x_2, \dots, x_n) = \sum_{e \in \{0,1\}^{m-n}} g(x_1, \dots, x_n, e_{n+1}, \dots, e_m) \tag{3}$$

Where $g_n \in AlgP$

Definition 6: If we have two functions $f(x_1, \dots, x_n)$ and $g(y_1, \dots, y_m)$, we say that f is *projection reducible* to g if there is a relabeling, $\sigma : \{y_1, \dots, y_m\} \mapsto \{0, 1, x_1, \dots, x_n\}$ such that: $f(x_1, \dots, x_n) = g(\sigma(y_1), \dots, \sigma(y_n))$

Theorem 1 (Valiant):

1. Every polynomial in n variables computable by circuit of size u is projection reducible to the determinant function on $u + 2$ variables.
2. Every function in $AlgNP$ is projection reducible to the permanent function.

[1] Similar to NP-complete, but you also count the number of accepting paths in the non-deterministic computation tree

Neat: since we're defined on fields, which are possibly infinite, there is not necessarily a way to create a boolean circuit for each algebraic circuit.

---

3 Blum-Shub-Smale (BSS) Model

Definition 7: Say we have a field $\mathbb{F}$. A *BSS* machine is a Turing Machine in which cells store elements from the field $\mathbb{F}$. Also:

- Shift state: move left or right
- Branch state: current cell has value q_1, go to cell q_1, otherwise go to cell q_2.
- Computation state: replace contents of cell a with $f(a)$, where f is a hardwired rational function , $f = \frac{p}{q}$ for p, q polynomials over $\mathbb{F}$.
- Register containing an element in the field.

Add other abilities:

- If we add the ability to compare $a \in \mathbb{F}$, whether $a > 0$, then we can compute anything P_{poly} in polynomial time (and recall that P_{poly} contains undecidable problems).[2]
- If we add $\lfloor x \rfloor$ then can do integer factorization in poly time. (Shamir)

Consider $\mathbb{F} = \mathbb{C}$.

Definition 8:

1. Then $P_{\mathbb{C}}$ is the set of languages that can be decided by a Turing Machine over $\mathbb{C}$ in polynomial time.
2. $NP_{\mathbb{C}}$ is the set of languages L s.t. $\exists L_0 \in P_{\mathbb{C}}$, s.t. $x \in L \equiv \exists (y_1, \ldots, y_{p(n)} \in \mathbb{C}^{p(n)})$ s.t. $(x, y) \in L^0$

Can also consider 0-1-$NP_{\mathbb{C}} = \{L \cap \{0,1\}^* \mid L \in NP_{\mathbb{C}}\}$. And 0-1-$NP_{\mathbb{C}} \subseteq PSPACE$

Just like $3SAT$ is the canonical NP-complete problem, we have $HN_{\mathbb{C}}$:

Definition 9: The decision problem $HN_{\mathbb{C}}$ is: given p_i polynomials in $x_1, \ldots, x_n$, do these polynomials have a common root?

[2] Also requires an ordered field, since we're using $>$

Note: can convert this into $3SAT$ via: $x \vee y \vee z \leftrightarrow (1-x)(1-y)(1-z) = 0$.

Theorem 2: 0-1-$HN_{\mathbb{C}}$ is complete for 0-1-$NP_{\mathbb{C}}$.

3.1 Undecidability

Definition 10: The *Mandelbrot Set* is:

$$a \in \mathbb{C}, P_a(z) = z^2 + a$$
$$\mathcal{M} = \{a : P_a(0), P_a(P_a(0)), \ldots, \} \text{ is bounded}$$

If we have comparison operations, can recognize the complement of $\mathcal{M}$: $a \in \overline{\mathcal{M}} \equiv \exists_j k : |P_a^k(0)| > 2$.

Theorem 3: $\mathcal{M}$ is undecidable