Equivalents of the weak multifunction pigeonhole

Chris Pollett

San Jose State University

Mar 22, 2005.

(Joint work with Norman Danner, Wesleyan)

Outline

- Motivation
- Weak Pigeonhole Principles
- Connections to Circuit Lower Bounds
- Some new results

Motivations for our Paper

- We wanted to understand how much mathematics is needed to show that there exists a set which requires a large circuit.

- Recently, Jerabek [J04] has shown this problem to be connected to the weak pigeonhole principle.

- So we wanted to explore this connection further…

Weak Pigeonhole Principles

Given a relation $R(x,y,z)$ (sometimes $R := f(x,z) = y$ for some f.)

- iWPHP(R):

  $\forall x < n^2 \; \exists! \; y < n \; R(x,y,z) \supset$

  $\quad \exists x_1, x_2 < n^2 \; \exists y < n \; [x_1 \neq x_2 \wedge R(x_1,y,z) \wedge R(x_2,y,z)]$

  If R is a function from n^2 into n, it is not one-to-one (two points map
  to the same value).

- sWPHP(R):

  $\forall x < n \; \exists! \; y < n^2 \; R(x,y,z) \supset \exists y < n^2 \; \forall x < n \neg R(x,y,z)$

  If R is a function from n into n^2, then it is not onto (some value for y
  is missed).

- mWPHP(R):

  $\forall x < n^2 \; \exists y < n \; R(x,y,z) \supset$

  $\quad \exists x_1, x_2 < n^2 \; \exists y < n \; [x_1 \neq x_2 \wedge R(x_1,y,z) \wedge R(x_2,y,z)]$

  If R is a multifunction from n^2 into n it is not one-to-one (two points
  map to the same value).

Relationships between Principles

- Using essentially just logic can show:

  mWPHP(R) ⊃ iWPHP(R)

  and

  mWPHP(R) ⊃ sWPHP(R)

- Depending on what formal system you are using it is not known the exact relationship between iWPHP(R) and sWPHP(R). (More on this later)

Circuits

We say a predicate Y (or function f) is computed by a family of AND, OR, NOT circuits $\{F_n\}$, if F_n correctly computes $Y(x)$ (resp. $f(x)$) for n bit numbers. If each F_n can be coded as a string of size less than $g(n)$, then we say the circuit family has size $g(n)$.

How hard is it to show there is a predicate that requires n^2-size circuits?

- Not known if any sets in NP requires n^2 size circuit families.

- If we allow sets in harder complexity classes can use Kannan style or fancier arguments.

What is a hard relation for circuits of size n^k?

Consider the p-time function f whose input is an encoding of a 0-1 valued circuit $C(x_1 \ldots x_n)$ of size $< n^k$ and whose output is a string $S = s_0 \ldots s_{m-1}$ where s_i is the output of C on input i (where i is suitably padded with 0's).

$$C(00\ldots0) \; C(00\ldots1) \ldots C(00\ldots(m-1)_2)$$
$$f: \lceil C \rceil \longrightarrow s_0 \, s_1 \ldots s_{m-1}$$

By our definition of size C can be written as a binary string of length $< n^k$. This in turn is a number less than $< 2^{n^k}$. If $m = 2n^k$, then S is a number $< 2^{2n^k}$, and we can apply sWPHP(f), to get a string which disagrees on some input $i < m$ with any circuit of size n^k.

Hard relation cont'd…

- Once we know such an S exists we can search for the least such S and use it to get a hard relation.

- Can use this idea to show there are hard relations for n^k sized circuits in NP^{NP}. (There is a slightly stronger result original noticed by Kannan.)

- Well studied theories such as S^3_2 or T^2_2 can carry out this argument.

Remembering what we are doing

- We are interested in how strong a formal system is needed to prove the previous result. (Lower bound).

- NP $\not\subset$ P/poly => P $\neq$ NP. If a formal system can't prove lower bounds, it can't prove NP $\not\subset$ P/poly; therefore, P=NP is consistent with the system.

- Understanding why such a consistency might be possible might shed light on how to prove P $\neq$ NP.

Formal Systems

- Have BASIC axioms like:

  $y \leq x \supset y \leq S(x)$

  $x+Sy = S(x+y)$

  for the symbols $0, S, +, \cdot, x\#y := 2^{|x||y|}, |x| :=$ length of x, $\dot{-},$
  $\lceil x/2^i \rceil, \leq$

- Have IND_m induction axioms of the form:

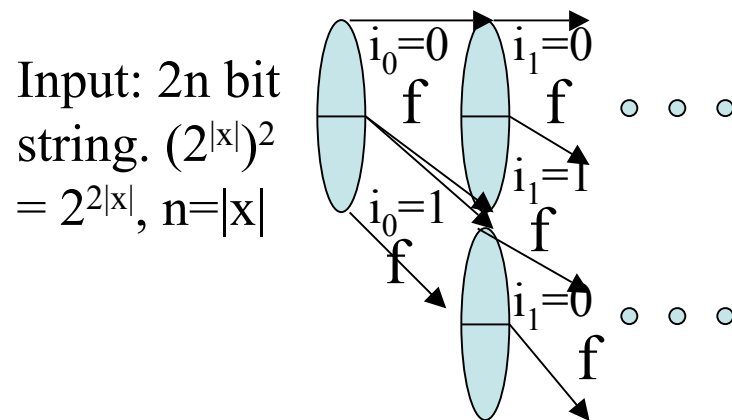 $A(0) \wedge \forall x<|t|_m[A(x) \supset A(S(x))] \supset A(|t|_m)$

  Here t is a term made of compositions of variables and
  our function symbols and $|x|_0=x, |x|_m=||x|_{m-1}|$.

Formal Systems cont'd

- A Σ^b_i-formula is a formula of the form:

$$\exists x_1 \leq t_1 \forall x_2 \leq t_2 \cdots Q x_i \leq t_i \, Q x_{i+1} \leq |t_{i+1}| A$$

$i+1$ alternations, innermost begin length bounded

where A is an open formula. A $\prod^b_i$-formula is defined similarly but with the outer quantifier being universal.

T^i_2 is the theory BASIC + Σ^b_i-IND$_0$

S^i_2 is the theory BASIC + Σ^b_i-IND$_1$

R^i_2 is the theory BASIC + Σ^b_i-IND$_2$

- These theories are well-studied and are known to be clsely connected with computational complexity.

- If we add to the language a function symbol $x\#_3 y$ with $|x\#_3 y| = |x|\#|y|$, then get theories T^i_3, S^i_3, R^i_3.

Formalizing Hard Sets

- Let $HARD_k$ be the formalization of the statement: "There is a string S of size $2n^k$ which is not computed correctly on all values $<2n^k$ by a circuit of size n^k."

- Let PV be the class of p-time functions (actually PV is really the function symbols plus defining axioms for these functions). It is open whether S^1_2 can prove sWPHP(PV).

- Jerabek [J04] shows over S^1_2 the statements $HARD_k$ for k>0 are equivalent to sWPHP(PV).

Intuition behind Jerabek

- We've already seen $sWPHP(FP) \supset HARD_k$.
- For $HARD_k \supset sWPHP(FP)$, suppose there is a p-time function f for which the sWPHP fails.
- Then there is a $n^{k'}$ size circuit family $\{C^f_n\}$ computing this function for some k'. Can iterate f according to a string $i_0 i_1 \cdots$

Input: 2n bit string. $(2^{|x|})^2 = 2^{2|x|}$, n=|x|

$i_0=0$ f $i_1=0$ f   ○ ○ ○

$i_0=1$ $i_1=1$ f

f $i_1=0$ f   ○ ○ ○

For any k>k' , iterating C^f_n O(|n|) times, we can get a circuit C' of size $n^{k'+1}$ whose domain is $|2n^{k-1}| \times 2n$-bit numbers but whose range is all strings of size $2n^k$.

Let C be the circuit which on input $i < 2n^k$ and s and an 2n bit number computes the ith bit of C'. For any fixed S of length $<2n^k$ we can now hard code the s that maps to it in C to get a circuit showing S is not the hard string of $HARD_k$.

Towards our results

- As mentioned before the relationship between sWPHP and iWPHP is not known for weak theories like S^1_2.

- The witnessing theorem for S^1_2 says if S^1_2 proves a formula like $\exists y \le s \forall z \le |u| A(x,y,z)$ then there is a p-time function f(x) such that $\forall z \le |u| A(x,f(x),z)$. For R^2_3 the analogous result gives an f contained in quasi-polynomial time.

- Using this Krajicek and Pudlak showed if S^1_2 proves iWPHP(FP) then RSA is insecure against p-time attacks.

- We asked two questions:

  1. Can one get a connection between circuit lower bound provability and RSA? Idea: Try mWPHP use pre-images.
  2. What happens when one takes relations in the pigeonhole principles rather than functions?

Our Results I

- Let HardBlks(k) be the formula which says there is a string S of length $2n^k$ such that there is no circuit C(i,s) of size less than n^k which outputs true iff s is the ith block of n bits from S.

- We show for each k>0, S^1_2 + sWPHP(P^{NP}(log)) proves HardBlks(k).

- On the other hand, S^1_2 + U_k HardBlks(k) proves sWPHP(NP).

- This does not yet give a connection with RSA. For that we needed to look at mWPHP since it implies both iWPHP and sWPHP.

Our Results II

- Given a relation R suppose we know there is a value for y of length < p(x) for some polynomial p such that R(x, y). Could then imagine the relation which computes $R(B(z), y_1) \Lambda R(y_1, y_2) \Lambda \ldots \Lambda R(y_m, E(z))$.

- The class Iter(PV,polylog) consists of such relations where R is p-time and iterate at most polylog times.

- Similarly, we define an IterHardBlks(k) which says an iterated circuit of size n^k cannot block recognize some string of size $2n^k$.

- We show R^2_2 proves IterHardBlks(k) is equivalent to mWPHP(Iter(PV,polylog)) and implies iWPHP(PV).

- Therefore, if R^2_3 proves lower bounds for iterated circuits, then RSA is vulnerable to quasi-polynomial time attacks.

Conclusion

- Since RSA is considered hard above seems to suggest R^2_3 cannot prove qNP $\not\subset$ qP/poly or least this harder circuit principle.

- On the other hand, it is known by Paris, Wilkie, and Woods that T^2_2 can prove mWPHP(Σ^b_1). So RSA is insecure against the NP-definable multifunctions of this theory (a class called GLS[†]).

- It would be cool to understand what happens for S^2_2. Its NP-definable multifunctions are projections of PLS (polynomial local search) problems.

Appendix RSA

- Public key crypto scheme proposed by Rivest, Shamir, Adleman 1977.

- For this talk, an RSA instance consists of (1) n=pq (where p and q are primes), (2) d and e which are inverses modulo (p-1)(q-1), (3) a message m < n and a ciphertext c < n such that $c \equiv m^e$ mod n and $m \equiv c^d$ mod n.

- Can solve this instance if given n, e, and c one can compute m.

RSA and the iWPHP(f) (Krajicek and Pudlak)

- Assume $\gcd(c,n) = 1$; otherwise, trivial.
- Suppose had a black box that given the function $f(x) = c^x$ mod n computes $x_1 < x_2 < n^2$ such that $c^{x1} \equiv c^{x2}$ mod n. Let $r_0 = x_1 - x_2.$
- Now calculate $r_1 = r_0 / \gcd(e, r_0) \ldots r_v = r_{v-1} / \gcd(e, r_{v-1})$ until $r_v = r_{v-1}$ (at most log r_0 steps). Call this last value r. (gcd is p-time using Euclid's Algorithm.)
- If s is order of c mod n, then can show $\gcd(e,s) = 1$. So also have that s divides r_i for each i. Hence s divides r.

More RSA and iWPHP

- Since by construction gcd(e, r) = 1 can using Euclid to get a d' such that d'e = 1 + tr.

- Now calculate $c^{d'}$ mod n.

- Done.

- This works since s divides r and $c^{d'} \equiv m^{ed'} \equiv m^{1+tr} \equiv m$ mod n