An Interactive Proof System for TQBF

Recall: Interactive Proof System (IPS)

- A Turing machine whose non-halting states are partitioned into two types: existential/guessing and coin-flipping. There are exactly two possible next steps from each coin-flipping state

Recall: Interactive Proof System (IPS)

- Let M be an IPS that always halts, and let C be a configuration of M. C is either an existential, coin-flipping, accepting, or rejecting configuration depending on its state.
- Let $Prob_a[C]$ denote the probability of reaching an accepting configuration from C

Recall: Interactive Proof System (IPS)

Let Prob[M accepts x] be $\text{Prob}_a[C_0]$, where C_0 is the initial configuration of M on x. We say that the IPS M accepts language L with bounded error if:

- for all $x \in L$, Prob[M accepts x] $\geq 2/3$, and
- for all $x \notin L$, Prob[M accepts x] $\leq 1/3$

- IP is the class of languages accepted by polynomial time bounded IPS's

Recall: Arithmetization

$$\phi = \forall x \exists y[(x \vee y) \wedge \forall z[(x \wedge z) \vee (y \wedge \bar{z}) \vee \exists w(z \vee (y \wedge \bar{w}))]]$$

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Claim: φ is valid iff $A_\varphi > 0$. Also, $A_\varphi \leq 2^{2^n}$, where $n = |A_\varphi|$

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Issue: the value of A_φ could be 2^{2^n}, where n = $|A_\varphi|$

Prover: "A_φ= 96"

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Issue: the value of A_φ could be 2^{2^n}, where $n = |A_\varphi|$

Workaround: do arithmetic mod a prime

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x\in\{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

....

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x \in \{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

....

Issue: can this polynomial be written down in polynomial time?

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x \in \{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

....

Issue: can this polynomial be written down in polynomial time?

From last time:

- If φ is simple, then $A_1(x)$ has degree at most $2|A_\varphi|$ (and so the prover *can* write $A_1(x)$ down in polynomial time)
- We can assume wlog that φ is simple (homework)

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x\in\{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1} \sum_{y=0}^{1} [(x+y) \cdot \prod_{z=0}^{1} [(x \cdot z + y \cdot (1-z)) + \sum_{w=0}^{1} (z + y \cdot (1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x \in \{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

Verifier:

- Check that $\alpha_1(0) \,.\, \alpha_1(1) = 96$

Recall: An IPS to test if $A_\varphi > 0$

(rough sketch)

$$A_\phi = \prod_{x=0}^{1}\sum_{y=0}^{1}[(x+y)\cdot\prod_{z=0}^{1}[(x\cdot z + y\cdot(1-z)) + \sum_{w=0}^{1}(z + y\cdot(1-w))]]$$

Prover: "$A_\varphi = 96$"

Verifier: Let $A_\varphi = \prod_{x\in\{0,1\}} A_1(x)$. What is $A_1(x)$?

Prover: "$A_1(x)$ is $\alpha_1(x) = 2x^2 + 8x + 6$"

Verifier:

- Check that $\alpha_1(0) . \alpha_1(1) = 96$
- Check that $A_1(x) = \alpha_1(x)$, i.e., that the prover isn't cheating, by plugging in a random number r for x and using recursion

Why Arithmetic Mod a Prime Works

Lemma: for sufficiently large $n = |A|$, $v(A) > 0$ iff there is a prime p between 2^n and 2^{2n} such that $v(A) \neq 0 \bmod p$

The proof uses two results from number theory:

Chinese Remainder Theorem: Let m be the product of distinct primes $p_1, p_2, \ldots, p_k$. Then for any integers $r_1, r_2, \ldots, r_k$, there is a unique v in the range $0 \leq v < m$ such that for all i, $v = r_i \bmod p_i$.

Prime Number Theorem: For any sufficiently large x, the number of primes that are $\leq x$ is at least $x/\ln x$.

Summary So Far

- Our goal is to show that TQBF is in IP
- Ideas:
  - Prover will help verifier evaluate an arithmetization of the TQBF instance
  - WLOG, work with *simple* qbf instances
  - Arithmetizations of simple qbf's can be expressed as low-degree polynomials
  - Polynomial evaluation can be done modulo primes to avoid working with large values

An IPS for TQBF

Input: a QBF φ; let φ be simple and have m quantifiers

Arithmetize φ to obtain A_φ ; let $A_0 = A_\varphi$; let $n = |A_\varphi|$

Prover:

Guess a prime p in the range in $[2^n , 2^{2n}]$

Guess a_0 in the range [1,…,p-1]

Verifier:

Check that p is prime, and p, a_0 are in the proper range

// check that $v(A_0) = a_0 \bmod p$

An IPS for TQBF, continued

// check that $v(A_0) = a_0 \bmod p$
For i from 1 to m do // m is # quantifiers of φ

Let $A_{i-1} = c_i + c_i' (O_u A_i(u))$, where O_u is leftmost ∑ or ∏
Prover:
Guess a polynomial $\alpha_i(u)$ of degree at most $2|A_\varphi|$
Verifier:
Check that $c_i + c_i' (O_u \alpha_i(u)) = a_{i-1} \bmod p$; if not, reject
Choose r_i randomly and uniformly in the range [0 ... p-1]
Let $a_i = \alpha_i(r_i) \bmod p$
Let A_i be the expression $A_i(r_i)$

Verifier: Check that $v(A_m) = a_m \bmod p$; if not, reject and otherwise accept

Proof of correctness (outline)

- A *strategy* $S(\varphi)$ is the Prover's choices of $\alpha_i(u)$

- Claim 1: If $v(A_\varphi) = a_0 \bmod p$ then for some strategy $S(\varphi)$, the IPS accepts with probability 1
- Claim 2: If $v(A_\varphi) \neq a_0 \bmod p$ then for all strategies $S(\varphi)$, the IPS rejects with probability at least $(1-2n/2^n)^n$ (where $n = |A_\varphi|$)

Proof of correctness (outline)

- Claim 1: If $v(A_\varphi) = a_0 \bmod p$ then for some strategy $S(\varphi)$, the IPS accepts with probability 1

Proof of correctness (outline)

- Claim 1: If $v(A_\varphi) = a_0$ mod p then for some strategy $S(\varphi)$, the IPS accepts with probability 1

- Proof : The strategy $S(\varphi)$ simply returns the polynomial $\alpha_i(u)$ that is equal to $A_i(u)$ (mod p)

Proof of correctness (outline)

- Claim 2: If $v(A_\varphi) \neq a_0$ mod p then for all strategies $S(\varphi)$, the IPS rejects with probability at least $(1\text{-}2n/2^n)^n$

Proof of correctness (outline)

- Claim 2: If $v(A_\varphi) \neq a_0 \bmod p$ then for all strategies $S(\varphi)$, the IPS rejects with probability at least $(1-2n/2^n)^n$

- Proof ideas: Fix any strategy $S = S(\varphi)$.
  - For each i between 0 and m, let $E_i = E_i(\varphi,S)$ be the event that $v(A_i) \neq a_i \bmod p$, or that the protocol rejects before the end of round i.
  - Show by induction that $\mathrm{Prob}[E_i] \geq (1-2n/2^n)^i$, where the probability is taken over the choice of r_i

Summary

- We've shown an interactive proof system that accepts TQBF
- Thus, IP = PSPACE: for any language L in PSPACE a prover can convince a coin-flipping verifier in polynomial time that a yes-instance x is indeed in L, and can fool the verifier with low probability when x is a no-instance of L

Summary

- The IP = PSPACE result raises other questions:
- If all of PSPACE can be proved (with low error probability) to a computationally limited coin-flipping verifier, can we limit the verifier further when proving membership in an NP language with low error probability?
- We'll come back to this question after a detour to approximation algorithms for NP-hard problems