An Interactive Proof System for co3SAT

This material was presented in class on March 29 and 31, 2016.

In order to construct interactive proof systems for co3SAT and, later, for TQBF, we introduce a new technical tool: Arithmetization of boolean formulas. Consider the following recursive definition of a function a that maps formulas on boolean variables $\{x_i\}_{i=1}^n$ to multinomials over $\mathbb{Z}$ in indeterminates $\{X_i\}_{i=1}^n$:

ϕ	$a(\phi)$
F	0
T	1
x_i	X_i
$\neg x_i$	$(1 - X_i)$
$f_1 \vee f_2$	$a(f_1) + a(f_2)$
$f_1 \wedge f_2$	$a(f_1) \cdot a(f_2)$

For example, if

$$\phi(x_1, x_2, x_3) = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3),$$

then

$$(a(\phi))(X_1, X_2, X_3) = (X_1 + 1 - X_2 + X_3) \cdot (X_1 + X_2 + 1 - X_3).$$

Let ϕ be a 3CNF formula on n variables $\{x_1, \ldots, x_n\}$ with m clauses $\{c_1, \ldots, c_m\}$. Each clause is itself a formula $c_j(x_{i_1}, x_{i_2}, x_{i_3})$ on three of the variables in $\{x_1, \ldots, x_n\}$, and any truth assignment $(b_1, \ldots, b_n)$ to the variables in ϕ either satisfies of falsifies c_j. In the multinomial $a(\phi)$, there is a factor $a(c_j)$ that corresponds to c_j, and $a(c_j)(X_{i_1}, X_{i_2}, X_{i_3})$ takes on the value 0, 1, 2, or 3 on $(a(b_1), \ldots, a(b_n))$, depending upon whether 0, 1, 2, or 3 of the literals in $c_j(b_{i_1}, b_{i_2}, b_{i_3})$ are true. Moreover, $a(c_j)$ is 0 on $(a(b_1), \ldots, a(b_n))$ if and only if $(b_1, \ldots, b_n)$ falsifies c_j. Since $a(\phi)$ is just the product of the $a(c_j)$'s, $1 \leq j \leq m$, the value of $a(\phi)(a(b_1), \ldots, a(b_n))$ is in the interval $[0, 3^m]$, for any truth assignment $(b_1, \ldots, b_n)$. Using these basic facts about arithmetization, we have

Fact 1. *For any truth assignment* $(b_1, \ldots, b_n)$,

$$\phi(b_1, \ldots, b_n) = F \longleftrightarrow a(\phi)(a(b_1), \ldots, a(b_n)) = 0.$$

Fact 2.

$$0 \leq \sum_{b_1 \in \{T,F\}} \sum_{b_2 \in \{T,F\}} \cdots \sum_{b_n \in \{T,F\}} (a(\phi))(a(b_1), \ldots, a(b_n)) \leq 2^n \cdot 3^m.$$

Fact 3.

$$\phi \notin \text{3SAT} \longleftrightarrow \sum_{b_1 \in \{T,F\}} \sum_{b_2 \in \{T,F\}} \cdots \sum_{b_n \in \{T,F\}} (a(\phi))(a(b_1), \ldots, a(b_n)) = 0$$

Now choose a prime p in the interval $(2^n \cdot 3^m, 2^{n+1} \cdot 3^m)$ (the existence of which is guaranteed by Chebyshev's Theorem, aka Bertrand's Postulate). For the rest of this lecture, we take $a(\phi)$ to be a multinomial in $\mathbb{Z}_p[X_1, \ldots, X_n]$ instead of $\mathbb{Z}[X_1, \ldots, X_n]$. Fact 2 guarantees that there is no wraparound when the computation is done mod p and hence, together with Fact 3, gives us

Fact 4.

$$\phi \notin \text{3SAT} \;\longleftrightarrow\; \sum_{b_1 \in \{T,F\}} \sum_{b_2 \in \{T,F\}} \cdots \sum_{b_n \in \{T,F\}} (a(\phi))(a(b_1), \ldots, a(b_n)) \equiv 0 \pmod{p}.$$

We will give a general *sum-check protocol* that allows the prover to convince the verifier of the truth of claims of the form

$$\sum_{z_1 \in \{0,1\}} \sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(z_1, z_2, \ldots, z_n) \equiv q \pmod{p},$$

where m is the maximum degree of any variable in h and p is a prime that is singly exponential in n and m. Note that, for the arithmetization that we are using here, the maximum degree of a variable in the multinomial h and the number of clauses in the 3SAT formula are equal, because the arithmetization of each clause is linear in each of the three relevant variables.

It will be a public-coin protocol, and hence we use Merlin (M) and Arthur (A) to refer to the prover and verifier, respectively. The special case in which $h = a(\phi)$ for some 3CNF formula ϕ and $q = 0$ allows Merlin to convince Arthur that ϕ is not in 3SAT, because the protocol can start with Merlin's sending Arthur a prime in the interval $(2^n \cdot 3^m, 2^{n+1} \cdot 3^m)$ and Arthur's verifying that it is indeed prime. Note that, although Arthur cannot evaluate a multinomial expression of the form $\sum_{b_1 \in \{T,F\}} \sum_{b_2 \in \{T,F\}} \cdots \sum_{b_n \in \{T,F\}} (a(\phi))(a(b_1), \ldots, a(b_n))$, he can write it down, because its size is polynomial in n and m. Moreover, Arthur can evaluate $h(z_1, z_2, \ldots, z_n)$ for any fixed vector $(z_1, z_2, \ldots, z_n) \in \mathbb{Z}_p^n$. For z_i not equal to 0 or 1, this expression does not correspond to a value of ϕ, even if h is of the form $a(\phi)$, but it is still perfectly well defined as the value of an n-variable multinomial over $\mathbb{Z}_p$.

For any fixed $(z_2, \ldots, z_n)$, $h(X_1, z_2, \ldots, z_n)$ is a univariate polynomial over Z_p of degree at most m. Let

$$h_1(X_1) = \sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(X_1, z_2, \ldots, z_n).$$

Then

$$\sum_{z_1 \in \{0,1\}} \sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(z_1, z_2, \ldots, z_n) \equiv q \pmod{p} \;\longleftrightarrow\; h_1(0) + h_1(1) \equiv q \pmod{p}.$$

Sum-Check Protocol:

Input: $h(X_1, \ldots, X_n)$, q, and p satisfying the above conditions

Merlin's Claim: $\sum_{z_1 \in \{0,1\}} \sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(z_1, z_2, \ldots, z_n) \equiv q \pmod{p}$

A: If $n = 1$, check that $h(0) + h(1) \equiv q \pmod{p}$ and accept if and only if it is. If $n > 1$, ask M for $h_1(X_1)$.

M: Send h_1.

A: Reject if $h_1(0) + h_1(1) \not\equiv q \pmod{p}$. Else, choose $a \in_R \mathbb{Z}_p$ and recursively use the sum-check protocol to have M prove that

$$\sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(a, z_2, \ldots, z_n) \equiv h_1(a) \pmod{p}.$$

Clearly, if Merlin is making a correct claim, then Arthur will always accept, because Merlin can always send the correct univariate polynomial h_1. On the other hand, if Merlin is making an incorrect claim, then Arthur will reject with probability at least $(1 - \frac{m}{p})^n$. We prove this by induction on n. Note first, however, that $(1 - \frac{m}{p})^n \geq (1 - \frac{mn}{p})$, and $p > 2^n \cdot 3^m$.

Clearly, Arthur will always reject if $n = 1$ and $h(0) + h(1) \not\equiv q \pmod{p}$. So assume that the rejection probability is at least $(1 - \frac{m}{p})^{n-1}$ when the number of variables is $n - 1$, and Merlin makes an incorrect claim. Now assume that Merlin claims incorrectly that

$$\sum_{z_1 \in \{0,1\}} \sum_{z_2 \in \{0,1\}} \cdots \sum_{z_n \in \{0,1\}} h(z_1, z_2, \ldots, z_n) \equiv q \pmod{p}$$

and runs the protocol with Arthur. When asked to provide a univariate polynomial, Merlin cannot send $h_1(X)$, because $h_1(0) + h_1(1) \not\equiv q \pmod{p}$. So Merlin must send some other univariate polynomial $s_1(X_1)$ of degree m with the property that $s_1(0) + s_1(1) \equiv q \pmod{p}$. When he and Arthur proceed to the recursive call of the sum-check protocol, Merlin will only be making a correct claim if $s_1(a) \equiv h_1(a) \pmod{p}$ for the a that Arthur chooses uniformly at random from $\mathbb{Z}_p$. Because s_1 and h_1 are different degree-m, univariate polynomials over $\mathbb{Z}_p$, the probability that they have the same value on a uniformly randomly chosen a is at most $\frac{m}{p}$ (which is the probability that this random a is one of the at most m distinct roots of the degree-m polynomial $(s_1 - h_1)(X_1)$). Thus, the probability that Arthur rejects Merlin's incorrect claim about this n-variable h is at least $1 - \frac{m}{p}$ (the probability that Merlin must make an incorrect claim in the recursive call) times $(1 - \frac{m}{p})^{n-1}$ (the probability that Arthur rejects an incorrect claim about an $(n-1)$-variable polynomial in the recursive call), *i.e.*, at least $(1 - \frac{m}{p})^n$.