IP = PSPACE: Simplified Proof

A. SHEN

Academy of Sciences, Moscow, Russia, CIS

Abstract. Lund et al. [1] have proved that PH is contained in IP. Shamir [2] improved this technique and proved that PSPACE = IP. In this note, a slightly simplified version of Shamir's proof is presented, using degree reductions instead of simple QBFs.

Categories and Subject Descriptors: F. 1. 2 [**Computation by Abstract Devices**]: Modes of computation—*Alternation and nondeterminism*; *probabilistic computation*; F.1.3 [**Computation by Abstract Devices**]: Complexity classes—*relation among complexity classes*; F.4.1 [**Mathematical Logic and Formal Languages**]; Mathematical Logic—*proof theory*

General Terms: Theory

Additional Key Words and Phrases: Interactive proofs, PSPACE

1. *Introduction*

It is well known that IP is contained in PSPACE. So, for equality, it is enough to show that some PSPACE-complete language has an IP-protocol. We use the language of true Quantified Boolean Formulas (QBF), that is, formulas $Q_1 x_1 \cdots Q_n x_n B(x_1 \cdots x_n)$, where $B(x_1 \cdots x_n)$ is a Boolean formula (without quantifiers) and $Q_1 \cdots Q_n \in \{\forall, \exists\}$.

Each Boolean formula $B(x_1 \cdots x_n)$ corresponds to a polynomial $b(x_1 \cdots x_n)$ where $\alpha \wedge \beta$ is replaced by $\alpha \cdot \beta$, $\neg \alpha$ by $1 - \alpha$ and $\alpha \vee \beta$ by $\alpha * \beta = \alpha + \beta - \alpha \cdot \beta \,(= 1 - (1 - \alpha)(1 - \beta))$. Its value coincides with the value of B on boolean arguments (0 = False, 1 = True).

Let $P(x, \dots)$ be a polynomial. Define three polynomials

$$\begin{aligned}
(AxP)(\cdots) &= P(0, \dots) \cdot P(1, \dots),\\
(ExP)(\cdots) &= P(0, \dots) * P(1, \dots),\\
(RxP)(x, \dots) &= P \bmod (x^2 - x)\\
&\qquad (\text{i.e., all } x^n \text{ with } n > 1 \text{ are replaced by } x).
\end{aligned}$$

The polynomial RxP has the same variables as P; in AxP and ExP, variable x is absent. Note that P and RxP coincide on Boolean arguments.

The work of the author was supported by National Science Foundation grant CCR 89-12586 and Air Force contract AFOSR 89-0271.

Author's address: Institute of Problems of Information Transmission, Academy of Sciences, Moscow, 103051, ul. Ermolovoi, 19, Russia, CIS.

Let $S(x_1 \cdots x_k)$ be a polynomial over some finite field F. Assume that we know an IP-protocol α allowing P to persuade V that $S(u_1 \cdots u_k) = v$ with probability 1 for any input $u_1 \cdots u_k, v \in F$ when it is true and with probability less than ϵ when it is false. Let U be a polynomial obtained from S by one of the operations (Ax, Ex, Rx). Let the degree of S with respect to x be less than some constant d (known to V). We construct a protocol β allowing P to persuade V that $U(c_1 \cdots c_l) = e$ with probability 1 for any input $c_1 \cdots c_l, e$ when it is true and with probability $< \epsilon + d/\#F$ when it is false. (Here, $\#F$ denotes the cardinality of F.) This protocol uses α as a procedure *called only once*.

2. *A Construction of IP-protocol* β

Case A. $U(y_1 \cdots y_l) = AxS(x, y_1 \cdots y_l)$.

P wants to persuade V that $U(c_1 \cdots c_l) = e$. P sends V the coefficients of a polynomial $s(x) = S(x, c_1 \cdots c_l)$. If $degree(s) > d$ or $s(0)s(1) \neq e$, V rejects. Otherwise, V sends P a random element $r \in F$. Now (using protocol α), P must persuade V that $S(r, c_1 \cdots c_l) = s(r)$.

Case E. $U(y_1 \cdots y_l) = ExS(x, y_1 \cdots y_l)$.

Replace $s(0)s(1)$ by $s(0) * s(1)$.

Case R. $U(x, y_1 \cdots y_l) = RxS(x, y_1 \cdots y_l)$.

P wants to persuade V that $U(f, c_1 \cdots c_l) = e$. P sends V the coefficients of a polynomial $s(x) = S(x, c_1 \cdots c_l)$. If $degree(s) > d$ or $s(0) + (s(1) - s(0))f \neq e$, V rejects (note that $s(0) + (s(1) - s(0))f$ is the value of $s(x) \bmod (x^2 - x)$ at f). Otherwise, V sends P a random element $r \in F$. Now (using protocol α), P must persuade V that $S(r, c_1 \cdots c_l) = s(r)$.

P can fool V either during α (probability less than ϵ) or if different polynomials $s(x)$ and $S(x, c_1 \cdots c_l)$ coincide at the random point r (probability not greater than $d/\#F$).

Let $\phi = Q_1 x_1 \cdots Q_n x_n B(x_1 \cdots x_n)$ be a *QBF*; $Q_1 \cdots Q_n \in \{\forall, \exists\}$. Consider a polynomial $b(x_1 \cdots x_n)$ corresponding to $B(x_1 \cdots x_n)$ and apply (sequentially) operations

$$\begin{array}{l} Rx_1, Rx_2, \ldots, Rx_n, \\ q_n x_n, \\ Rx_1, Rx_2, \ldots, Rx_{n-1}, \\ q_{n-1} x_{n-1}, \\ \vdots \\ Rx_1, Rx_2, \\ q_1 x_1, \end{array}$$

where $q_i = A$ or E if $Q_i = \forall$ or $\exists$, respectively. After these operations, we get a constant equal to 0 or 1, depending on the truth value of ϕ. P can persuade V that this constant is 1 using the reduction steps described. Ultimately, the equality $b(u_1 \cdots u_n) = v$ must be checked for some $u_1 \cdots u_n, v$; V can do this alone because the formula B is known. The probability of error does not exceed

$$\frac{(\text{number of operations A, E, R}) \cdot (\text{maximal degree})}{(\#F)}.$$

If the length of QBF was l, then number of operations is $O(l^2)$ and maximal degree is $O(l)$ (degree of t does not exceed l, R-operations reduce it to 1 and later all degrees are not greater than 2). If $\#F$ is about l^4, the probability of error tends to 0 when $l \to \infty$. So we can use $F = Z/pZ$ where p is a prime of logarithmic length (p can be chosen by P or V because primality testing is trivial for numbers of this size). It is easy to see that Verifier is weak in the sense of Shamir [2].

ACKNOWLEDGMENTS. I am grateful to Prof. Michael Sipser who explained the original proof of IP-PSPACE to me, and suggested that I write this simplified version of it. I also want to thank MIT's Mathematics Department and Laboratory for Computer Science for their hospitality.

REFERENCES

1. LUND, C., FORTNOW, L., KARLOFF, H., AND NISAN, N. Algebraic methods for interactive proof systems. *J. ACM 39*, 4 (Oct. 1992), 859–868.
2. SHAMIR, A. IP = PSPACE. *J. ACM 39*, 4 (Oct. 1992), 869–877.

RECEIVED MAY 1991; REVISED NOVEMBER 1991; ACCEPTED JULY 1991