19/11/2021 (IP).

$$\overline{GI} := \{ (G_1, G_2) \mid G_1 \text{ is non-isomorphic to } G_2 \}$$

$$(G_1, G_2) \overset{?}{\in} \overline{GI}$$

Prover

Verifier.

1) tosses a random coin and selects the graph G_1 or G_2 based on this random coin

2) selects a random permutation of the vertices.

3) Applies the permutation to G_i where G_i is graph chosen in step (1)

Call this permuted graph H.

$\xleftarrow{\quad\quad}$ sends H to prover.

1) needs to send $i \in \{1,2\}$ which was picked by verifier

$\xrightarrow{\ i'\ }$

4) accepts if $i' = i$ rejects if $i' \neq i$

Suppose G_1, G_2 are non-isomorphic.

Prover can figure out with certainity the graph G_i isomorphic to H.

whereas if G_1 and G_2 are isomorphic

$G_1 \cong G_2 \cong H$

Prover can at best guess G_1 or G_2

$\therefore$ the acceptance prob. $\leq \frac{1}{2}$.
(error)

(private-coin)
Interactive proof systems ($IP[k]$)
$\uparrow$ # rounds

[Completeness] : if $x \in L$, $\exists$ a proof s.t. Verifier accepts with prob. $\geq 2/3$

[Soundness] : if $x \notin L$, $\forall$ proofs Verifier rejects with prob $\geq 2/3$

Facts :-

1) $NP \subseteq IP$
2) $\overline{GI} \subseteq IP$
3) Make Completeness Perfect.
   if $x \in L$, acceptance prob $= 1$.
4) What happens if Prover is made probabilistic ?

Again it adds no power.

5) pubic - coin Interactive proof system
(Arthur - Merlin proof system)
AM

6) $IP[K]$ vs $IP[K+1]$

1) $IP[O(1)] = IP[2]$
2) if $coNP \subseteq IP[2]$, then PH collapses.

[Goldwasser - Micali - Rackoff '80s]

Is UNSAT $\in$ IP?

[Fortnow - Sipser conjectured UNSAT $\notin$ IP.] false

Thm [LFKN'89]
$\#3SAT \in IP$ poly. rounds. $\Rightarrow \#P \subseteq IP$

Thm:- [Shamir '90] $TQBF \in IP$
$\Rightarrow PSPACE \subseteq IP$

7) $IP \subseteq PSPACE$. [$IP = PSPACE$]
Prover can be simulated in PSPACE.

Thm:- [LFKN'89] $\#3SAT \in IP$.

$\varphi := (x_1 \vee x_3 \vee \bar{x}_5) \wedge (x_2 \vee x_4 \vee \bar{x}_3) \wedge \dots$

#variables $= n$ #clauses $= m$

Given φ, a number K,

Prover has to prove that

φ has K satisfying assignment

1) arithmetization.

$$(1-x_1)\cdot(1-x_3)\, x_5$$

$C_1 = \text{False} \iff (1-x_1)(1-x_3)x_5 = 1$

$C_1 = \text{True} \iff (1-x_1)(1-x_3)x_5 = 0$

$$C_1 = [1-(1-x_1)(1-x_3)x_5]$$

$C_1 = \text{True} \iff$ the above expression $= 1$.

associate such expression with every clause.

$$P_\varphi(x_1, \dots, x_n) = [1-(1-x_1)(1-x_3)\cdot x_5]\cdot$$
$$[1-(1-x_2)(1-x_4)\cdot x_3]\cdot$$
$$\text{-- -- -- --}$$

= product of expression associated with clauses.

$$\deg(P_\varphi) \le 3m$$

$$P_\varphi =: P$$

Satisfying assignments of φ

$$= \sum_{x \in \{0,1\}^n} P(x_1, \dots, x_n)$$

want to prove that

$$\sum_{x \in \{0,1\}^n} P(x_1, \dots, x_n) = K.$$

$$\Rightarrow \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} P(x_1, \dots, x_n) = K \quad (1)$$

Consider $q_1(x_1) = \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} P(x_1, \ldots, x_n)$

$\deg q_1(x_1) \leq m$ (assuming a variable occurs only at most once in each clause)

$\leq 3m.$

$\therefore$ q_1 is a low-degree Univariate polynomials.

$\boxed{q_1(0) + q_1(1) = K} \rightarrow$ to verify.

Verifier :- 1) It asks prover to send a prime λ between 2^{n+1} and 2^{2n}

(1) holds $\Leftrightarrow$ (1) holds (mod λ)

2) It asks prover to send the polynomial $q_1(x_1)$.

$q_1(x_1) \pmod{\lambda}$

message length is $O(m \cdot n)$

Prover :- sends q_1' claiming it is q_1.

Verifier :- $\boxed{q_1(0) + q_1(1) = K}$

$\hookrightarrow$ want to verify.

Checks $q_1'(0) + q_1'(1) = K$ ✓

It is a possibility that $q_1' \neq q_1$

but $q_1'(0) + q_1'(1) = K \pmod{p}$

So, verifier picks a random number α between $[0, p-1]$

and tries to verify that.

$q_1'(\alpha) = q_1(\alpha) \pmod{p}$

if $q_1' \neq q_1$ then there is a very low prob. of passing the last equality test $\leq \frac{3m}{p}$.

$$(2) \quad q_1'(\alpha) = \sum_{x_2=0}^{1} \sum_{x_3=0}^{1} \cdots \sum_{x_n=0}^{1} P(\alpha, x_2, \ldots, x_n) \pmod{p}$$

Verifier can evaluate this

Verifer recurses by writing as a polynomial in x_2.

$$q_2(x_2) = \sum_{x_3=0}^{1} \cdots \sum_{x_n=0}^{1} P(\alpha, x_2, x_3, \ldots, x_n)$$

need to verify $q_2(0) + q_2(1) = q_1'(\alpha)$

Prover :- sends q_2' claiming it as q_2.

Verifier :- Checks $q_2'(0) + q_2'(1) = q_1'(\alpha)$

if the check passes then

Verifier pickey random $\beta \in [0, \lambda-1]$

$$q_2'(\beta) = q_2(\beta)$$
$$= \sum_{x_3 \in \{0,1\}} \sum_{x_4 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} P(\alpha, \beta, x_3 \ldots x_n)$$

error prob. $\leq n \cdot \frac{3m}{\lambda}$

$$\simeq \frac{\text{poly}(n)}{2^{O(n)}}$$