Protocoles interactifs

Léo Gagnon

December 9, 2019

Université de Montréal

Qu’est-ce qu’une preuve?

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

- Il peut simplement essayer de vérifier la preuve directement.

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

- Il peut simplement essayer de vérifier la preuve directement.
- Il peut poser des questions au prouveur pour obtenir de l'information supplémentaire.

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

- Il peut simplement essayer de vérifier la preuve directement.
- Il peut poser des questions **aléatoires** au prouveur pour obtenir de l'information supplémentaire.

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

- Il peut simplement essayer de vérifier la preuve directement.
- Il peut poser des questions **aléatoires** au prouveur pour obtenir de l'information supplémentaire.
- Et autre choses...

Intuition

Quelles caractéristiques une procédure de preuve doit-elle avoir?

- Seulement une théorème vrai peut être prouvé
- Même si la preuve est très dificile, la vérification de la validité de la preuve doit être relativement façile.

Qu'est-ce que le vérificateur peut faire pour vérifier la preuve?

- Il peut simplement essayer de vérifier la preuve directement.
- Il peut poser des questions **aléatoires** au prouveur pour obtenir de l'information supplémentaire.
- Et autre choses...

On dira qu'un langage L est prouvable si $\forall x$ il est possible pour un prouveur tout-puissant de prouver $x \in L$ à un vérificateur ayant des capacités limités.

Exemple

Quelle est la classe de langages qu'un vérificateur peut prouver sans aucune interaction avec le prouveur?

Exemple

Quelle est la classe de langages qu'un vérificateur peut prouver sans aucune interaction avec le prouveur?

P

Exemple

Quelle est la classe de langages prouvables à un vérificateur fonctionnant en temps polynomial avec interaction?

Exemple

Quelle est la classe de langages prouvables à un vérificateur fonctionnant en temps polynomial avec interaction?

NP

Exemple

Quelle est la classe de langages prouvables à un vérificateur fonctionnant en temps polynomial probabiliste avec interaction?

Exemple

Quelle est la classe de langages prouvables à un vérificateur fonctionnant en temps polynomial probabiliste avec interaction?

IP !

Définition informelle : Les protocoles interactifs

Protocoles interactifs (IP) :

- Le prouveur est tout-puissant.

Définition informelle : Les protocoles interactifs

Protocoles interactifs (IP) :

- Le prouveur est tout-puissant.
- Le vérificateur peut calculer en temps polynomial et il a accès à des bits aléatoires secrets.

Définition informelle : Les protocoles interactifs

Protocoles interactifs (IP) :

- Le prouveur est tout-puissant.
- Le vérificateur peut calculer en temps polynomial et il a accès à des bits aléatoires secrets.
- Après un nombre polynomial de questions-réponses, le vérificateur doit correctement reconnaître le mot avec grande probabilité.

Définition formelle

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.
- Un vérificateur est une fonction $V : \Sigma^* \to \Sigma^*$ calculable en temps déterministe polynomial.

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.
- Un vérificateur est une fonction $V : \Sigma^* \to \Sigma^*$ calculable en temps déterministe polynomial.
- $V(x, r, z_1, ..., z_{k-1}) = y_k \in \Sigma^*$ est la k-ème question du vérificateur et $P(x, y_1, ..., y_k) = z_k \in \Sigma^*$ est la k-ème réponse du prouveur.

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.
- Un vérificateur est une fonction $V : \Sigma^* \to \Sigma^*$ calculable en temps déterministe polynomial.
- $V(x, r, z_1, ..., z_{k-1}) = y_k \in \Sigma^*$ est la k-ème question du vérificateur et $P(x, y_1, ..., y_k) = z_k \in \Sigma^*$ est la k-ème réponse du prouveur.
- La dernière question du vérificateur est interprété comme sa décision (0 ou 1).

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.
- Un vérificateur est une fonction $V : \Sigma^* \to \Sigma^*$ calculable en temps déterministe polynomial.
- $V(x, r, z_1, ..., z_{k-1}) = y_k \in \Sigma^*$ est la k-ème question du vérificateur et $P(x, y_1, ..., y_k) = z_k \in \Sigma^*$ est la k-ème réponse du prouveur.
- La dernière question du vérificateur est interprété comme sa décision (0 ou 1).
- On doit avoir $x \in L \implies$ il existe un prouveur qui fait accepter le vérificateur avec probabilité $\geq 2/3$

Définition formelle

Soit x le mot en entrée, L le langage et r l'aléat.

- Un prouveur est une fonction $P : \Sigma^* \to \Sigma^*$ sans contrainte de calculabilité.
- Un vérificateur est une fonction $V : \Sigma^* \to \Sigma^*$ calculable en temps déterministe polynomial.
- $V(x, r, z_1, ..., z_{k-1}) = y_k \in \Sigma^*$ est la k-ème question du vérificateur et $P(x, y_1, ..., y_k) = z_k \in \Sigma^*$ est la k-ème réponse du prouveur.
- La dernière question du vérificateur est interprété comme sa décision (0 ou 1).
- On doit avoir $x \notin L \implies$ pour tout prouveur, le vérificateur accepte avec probabilité $\leq 1/3$

Définition formelle

Protocole de s tours ($2s$ messages) :

$$\begin{aligned} y_1 &= V(x, r) \\ z_1 &= P(x, y_1) \\ &\ldots \\ y_i &= V(x, r, z_1, ..., z_{i-1}) \\ z_i &= P(x, y_1, ..., y_i) \\ &\ldots \\ 1_{x \in L} &= V(x, r, z_1, ..., z_s) \end{aligned}$$

Définition formelle

Définition :

$\mathrm{IP}[s(n)]$ est l'ensemble des langages L possédant un protocole à $s(n)$ tours. Alors $\mathrm{IP} = \bigcup_{k \in \mathbb{N}} \mathrm{IP}[n^k]$ est l'ensemble des protocoles interactifs ayant un nombre polynomial de tours.

Exemple : Isomorphisme de graphe

Figure 10.1 – Deux graphes isomorphes.

Exemple : Isomorphisme de graphe

FIGURE 10.1 – Deux graphes isomorphes.

- *entrée :* deux graphes non-orientés $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$;
- *question :* G_1 est-il isomorphe à G_2?

Exemple : Isomorphisme de graphe

Figure 10.1 – Deux graphes isomorphes.

Existe-t-il une permutation σ des sommest $\{1, ..., n\}$ de G_1 telle que $\forall (i, j)$,

$$(i, j) \in E_1 \iff (\sigma(i), \sigma(j)) \in E_2)$$

Pour l'exemple plus haut, $\sigma = (12543)$.
Donc $\mathrm{ISO} \in NP$. Qu'en est-il de $\mathrm{CO}\mathrm{ISO}$?

Exemple : non-Isomorphisme de graphe

Protocole pour coISO sur entrée (G_1, G_2) :

- V Tirer au hasard $i \in \{1, 2\}$. Appliquer une permutation aléatoire aux sommets de G_i, pour obtenir un nouveau graphe H. Envoyer H à P.
- P Identifier quel graphe G_j, pour $j \in \{1, 2\}$, a été permuté pour obtenir H. Envoyer j à V.
- V Accepter ssi $i = j$.

Exemple : non-Isomorphisme de graphe

Protocole pour $\mathrm{CO}\mathrm{ISO}$ sur entrée (G_1, G_2) (Valide cette fois):

- V Tirer au hasard $i, i' \in \{1, 2\}$. Appliquer une permutation aléatoire aux sommets de G_i pour obtenir un nouveau graphe H et de $G_{i'}$ pour obtenir H'. Envoyer H et H' à P.
- P Identifier quels graphes $G_j, G_{j'}$, pour $j \in \{1, 2\}$, ont étés permutés pour obtenir H et H' respectivement. Envoyer j et j' à V.
- V Accepter ssi $i = j$ et $i' = j'$.

IP = PSPACE (Shamir, 1990)

Structure de la preuve

- $\mathrm{IP} \subseteq \mathrm{PSPACE}$ car on peut simuler tout les déroulements possibles d'un protocole IP en espace polynomiale.
- Pour montrer $\mathrm{PSPACE} \subseteq \mathrm{IP}$ on va montrer qu'il existe un protocole interactif pour le langage PSPACE-complet QBF.

IP $\subseteq$ PSPACE

Idée de la preuve

Soit $L \in \mathrm{IP}$. Pour savoir si $x \in L$, on peut simuler tout les déroulements possibles du protocole et accepter ssi la probabilité maximale à laquelle le prouveur peut faire accepter le vérificateur est $\geq 2/3$.

IP $\subseteq$ PSPACE

On peut considérer tout les déroulements possibles du protocole comme un arbre où la racine est la première question du vérificateur et les feuilles sont les dernières question du vérificateur (1 ssi le vérificateur accepte). Remarquons que :

- La pronfondeur est polynomiale car c'est le nombre de tours.
- Chaque message a une taille polynomiale donc chaque noeud a au maximum 2^{n^c} enfants pour c une constante.

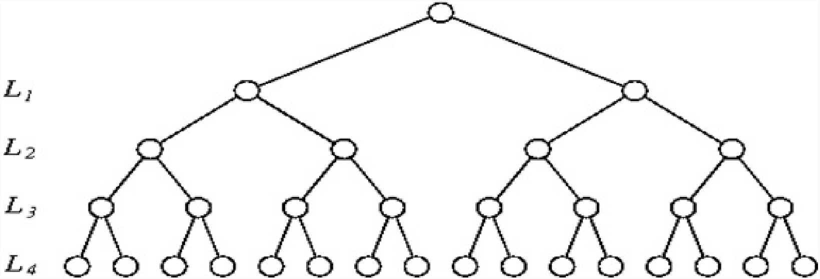
IP $\subseteq$ PSPACE

On peut assigner récursivement une valeur à tout les noeuds du graphe de la manière suivante :

- Les feuilles sont égales à 1 ssi le vérificateur accepte.
- Si un noeud intermédiaire représente une réponse de P sa valeur est égale au maximum de la valeur de ses enfants.
- Si un noeud intermédiaire représente une question de V sa valeur est égale à la moyenne pondérée de la valeur de ses enfants.

On doit garder en mémoire le nombre d'enfant d'un noeud pour faire la moyenne et on doit aussi se souvenir où on est dans la récursion. Ceci est faisable dans PSPACE.

IP $\subseteq$ PSPACE



IP $\subseteq$ PSPACE

Soit r la valeur à la racine. Donc r est la probabilité maximale avec laquelle un prouveur peut faire accepter un vérificateur. On a donc :

$$x \in L \iff r \geq 2/3$$

et on conlu

$$\mathrm{IP} \in \mathrm{PSPACE}$$

PSPACE $\subseteq$ IP

PSPACE $\subseteq$ IP

Idée de la preuve

Puisque $(A \leq_m^p B) \wedge (B \in IP) \implies A \in IP$ et que QBF est PSPACE-complet pour ces réductions alors on va donner un protocole IP pour QBF. Le prouveur devra donc prouver au vérificateur que sa formule booléene quantifiée est bien satisfaisable. La difficultée est de s'assurer que la taille des messages reste polynomiale et que le vérificateur n'ai pas de tâches trop difficiles à effectuer.

Définition : Formule booléence quantifiée

Une formule booléence quantifiée est une formule de la forme

$$Q_1 x_1 Q_2 x_2 ... Q_n x_n \varphi(x_1, ..., x_n)$$

où Q_i est un quantificateur $\exists$ ou $\forall$, et $\phi(x_1, ..., x_n)$ une formule booléenne sans quantificateurs sur les variables $x_1, ..., x_n$.

Définition : QBF

Le langage QBF est l'ensemble des formules booléennes quantifiées qui sont vraies.

PSPACE $\subseteq$ IP : Préliminaires

Arithmétisation

Soit ϕ une formule booléenne quantifiée. On peut supposer que

$$\phi = \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n)$$

où ψ est une formule sans quantificateurs en 3-CNF, n est pair et $a_i \in \{0, 1\}$.

PSPACE $\subseteq$ IP : Préliminaires

Arithmétisation

On peut associer à une formule en 3-CNF ψ un polynôme Q_ψ de la façon suivante :

- Transformer chaque clause de trois littéraux $(a_i \vee \neg a_j \vee a_k)$ en le polynôme $x_i + (1 - x_j) + x_k$
- Le polynôme Q_ψ est le produit de toutes les clauses.

On obtient donc

$$Q_\psi(a_1, ...a_n) > 0 \iff \psi(a_1, ..., a_n) = 1$$

PSPACE $\subseteq$ IP : Préliminaires

Arithmétisation

Selon le même raisonnement, on peut associer à

$$\phi = \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n)$$

le polynôme

$$Q_\phi = \sum_{a_1 \in \{0,1\}} \prod_{a_2 \in \{0,1\}} \sum_{a_3 \in \{0,1\}} ... \prod_{a_n \in \{0,1\}} Q_\psi(a_1, ..., a_n)$$

Arithmétisation

Selon le même raisonnement, on peut associer à

$$\phi = \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n)$$

le polynôme

$$Q_\phi = \exists_1 \forall_2 \exists_3 ... \forall_n Q_\psi(a_1, ..., a_n)$$

PSPACE $\subseteq$ IP : Préliminaires

Arithmétisation

Selon le même raisonnement, on peut associer à

$$\phi = \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n)$$

le polynôme

$$Q_\phi = \exists_1 L_1 \forall_2 L_1 L_2 \exists_3 ... L_1 ... L_{n-1} \forall_n L_1 ... L_n Q_\psi(a_1, ..., a_n)$$

Où

$$L_i(Q)(x_1, ..., x_n) = x_i Q_{x_i=1} + (1 - x_i) Q_{x_i=0}$$
$$\exists_i = \sum_{a_i \in \{0,1\}}$$
$$\forall_i = \prod_{a_i \in \{0,1\}}$$

PSPACE $\subseteq$ IP : Préliminaires

Arithmétisation

Selon le même raisonnement, on peut associer à

$$\phi = \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n)$$

le polynôme

$$Q_\phi = \exists_1 L_1 \forall_2 L_1 L_2 \exists_3 ... L_1 ... L_{n-1} \exists_n L_1 ... L_n Q_\psi(a_1, ..., a_n)$$

Tel que

$$Q_\phi(a_1, ...a_n) > 0 \iff \exists a_1 \forall a_2 \exists a_3 ... \forall a_n \psi(a_1, ...a_n) = 1$$

PSPACE $\subseteq$ IP : Préliminaires

Lemme

Soit p un nombre premier. $\mathbb{F}_p$ désigne le corps à p éléments (les entiers modulo p). Si $q(x)$ et $q'(x)$ sont deux polynômes distincts de degré $\leq d$ sur $\mathbb{F}_p$, alors

$$\underset{r \in \mathbb{F}_p}{Pr}(q(r) = q'(r)) \leq d/p$$

Preuve

$(q - q')$ est au plus de degré d donc a au plus d racines sur le corps $\mathbb{F}_p$. Il y a donc au plus d valeurs de r pour lesquelles $(q - q')(r) = 0$.

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{\lfloor Q_\phi \rfloor}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{\lfloor Q_\phi \rfloor}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q_0' et $l_{1,1}'$ et le premier p.

PSPACE $\subseteq$ IP : Protocole

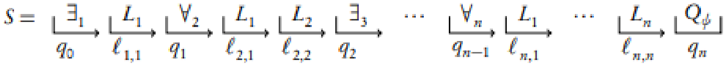


Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q_0' et $l_{1,1}'$ et le premier p.

V vérifie que $q_0' = \exists_1 l_{1,1}'$ et demande $q_1(x_1)$ à P

PSPACE $\subseteq$ IP : Protocole

$$S = \underbrace{\exists_1}_{q_0} \underbrace{L_1}_{\ell_{1,1}} \underbrace{\forall_2}_{q_1} \underbrace{L_1}_{\ell_{2,1}} \underbrace{L_2}_{\ell_{2,2}} \underbrace{\exists_3}_{q_2} \cdots \underbrace{\forall_n}_{q_{n-1}} \underbrace{L_1}_{\ell_{n,1}} \cdots \underbrace{L_n}_{\ell_{n,n}} \underbrace{Q_\psi}_{q_n}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q'_0 et $l'_{1,1}$ et le premier p.

V vérifie que $q'_0 = \exists_1 l'_{1,1}$ et demande $q_1(x_1)$ à P

P envoie q'_1 sensé être $q_1(x_1)$

PSPACE $\subseteq$ IP : Protocole

$$S = \underbrace{\exists_1}_{q_0}\ \underbrace{L_1}_{\ell_{1,1}}\ \underbrace{\forall_2}_{q_1}\ \underbrace{L_1}_{\ell_{2,1}}\ \underbrace{L_2}_{\ell_{2,2}}\ \underbrace{\exists_3}_{q_2}\ \cdots\ \underbrace{\forall_n}_{q_{n-1}}\ \underbrace{L_1}_{\ell_{n,1}}\ \cdots\ \underbrace{L_n}_{\ell_{n,n}}\ \underbrace{Q_\phi}_{q_n}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q_0' et $l_{1,1}'$ et le premier p.

V vérifie que $q_0' = \exists_1 l_{1,1}'$ et demande $q_1(x_1)$ à P

P envoie q_1' sensé être $q_1(x_1)$

V génère $r_1 \in_R \mathbb{F}_p$, vérifie que $l_{1,1}'(r_1) = L_1(q_1')(r_1)$ et demande $l_{2,1}(r_1, x_2)$ à P.

PSPACE $\subseteq$ IP : Protocole

$$S = \underbrace{\exists_1}_{q_0} \underbrace{L_1}_{\ell_{1,1}} \underbrace{\forall_2}_{q_1} \underbrace{L_1}_{\ell_{2,1}} \underbrace{L_2}_{\ell_{2,2}} \underbrace{\exists_3}_{q_2} \cdots \underbrace{\forall_n}_{q_{n-1}} \underbrace{L_1}_{\ell_{n,1}} \cdots \underbrace{L_n}_{\ell_{n,n}} \underbrace{Q_\psi}_{q_n}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q'_0 et $l'_{1,1}$ et le premier p.

V vérifie que $q'_0 = \exists_1 l'_{1,1}$ et demande $q_1(x_1)$ à P

P envoie q'_1 sensé être $q_1(x_1)$

V génère $r_1 \in_R \mathbb{F}_p$, vérifie que $l'_{1,1}(r_1) = L_1(q'_1)(r_1)$ et demande $l_{2,1}(r_1, x_2)$ à P.

P envoie $l'_{2,1}(x_2)$ sensé être $l_{2,1}(r_1, x_2)$

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{\underline{Q_\phi}}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie les coefficients de q_0' et $l_{1,1}'$ et le premier p.

V vérifie que $q_0' = \exists_1 l_{1,1}'$ et demande $q_1(x_1)$ à P

P envoie q_1' sensé être $q_1(x_1)$

V génère $r_1 \in_R \mathbb{F}_p$, vérifie que $l_{1,1}'(r_1) = L_1(q_1')(r_1)$ et demande $l_{2,1}(r_1, x_2)$ à P.

P envoie $l_{2,1}'(x_2)$ sensé être $l_{2,1}(r_1, x_2)$

V vérifie que $q_1'(r_1) = \forall_2 l_{2,1}'$. Génère $r_2 \in_R \mathbb{F}_p$, et demande $l_{2,2}(x_1, r_2)$ à P.

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\underbrace{Q_\psi}}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$

PSPACE $\subseteq$ IP : Protocole

$$S = \underbrace{\exists_1}_{q_0} \underbrace{L_1}_{\ell_{1,1}} \underbrace{\forall_2}_{q_1} \underbrace{L_1}_{\ell_{2,1}} \underbrace{L_2}_{\ell_{2,2}} \underbrace{\exists_3}_{q_2} \cdots \underbrace{\forall_n}_{q_{n-1}} \underbrace{L_1}_{\ell_{n,1}} \cdots \underbrace{L_n}_{\ell_{n,n}} \underbrace{Q_\psi}_{q_n}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$

V vérifie que $l'_{2,1}(r_2) = L_1(l'_{2,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $q_2(r_1, x_2)$ à P.

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\lfloor Q_\psi \rfloor}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$

V vérifie que $l'_{2,1}(r_2) = L_1(l'_{2,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $q_2(r_1, x_2)$ à P.

P envoie $q'_2(x_2)$ sensé être $q_2(r_1, x_2)$

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{\underline{Q_\phi}}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

- P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$
- V vérifie que $l'_{2,1}(r_2) = L_1(l'_{2,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $q_2(r_1, x_2)$ à P.
- P envoie $q'_2(x_2)$ sensé être $q_2(r_1, x_2)$
- V vérifie que $l'_{2,2}(r_1) = L_2(q'_2)(r_2)$. Génère un nouveau $r_2 \in_R \mathbb{F}_p$ et demande $l_{3,1}(r_1, r_2, x_3)$ à P.

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\xrightarrow{Q_\phi}}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

- P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$
- V vérifie que $l'_{2,1}(r_2) = L_1(l'_{2,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $q_2(r_1, x_2)$ à P.
- P envoie $q'_2(x_2)$ sensé être $q_2(r_1, x_2)$
- V vérifie que $l'_{2,2}(r_1) = L_2(q'_2)(r_2)$. Génère un nouveau $r_2 \in_R \mathbb{F}_p$ et demande $l_{3,1}(r_1, r_2, x_3)$ à P.
- P envoie $l'_{3,1}(x_3)$ sensé être $l_{3,1}(r_1, r_2, x_3)$

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{Q_\phi}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{2,2}(x_1)$ sensé être $l_{2,2}(x_1, r_2)$

V vérifie que $l'_{2,1}(r_2) = L_1(l'_{2,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $q_2(r_1, x_2)$ à P.

P envoie $q'_2(x_2)$ sensé être $q_2(r_1, x_2)$

V vérifie que $l'_{2,2}(r_1) = L_2(q'_2)(r_2)$. Génère un nouveau $r_2 \in_R \mathbb{F}_p$ et demande $l_{3,1}(r_1, r_2, x_3)$ à P.

P envoie $l'_{3,1}(x_3)$ sensé être $l_{3,1}(r_1, r_2, x_3)$

V vérifie que $q'_2(r_2) = \exists_3 l'_{3,1}$. Génère un nouveau $r_3 \in_R \mathbb{F}_p$ et demande $l_{3,2}(x_1, r_2, r_3)$ à P.

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\underbrace{Q_\psi}}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\underbrace{Q_\psi}}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

V vérifie que $l'_{3,1}(r_3) = L_1(l'_{3,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $l_{3,3}(r_1, x_2, r_3)$ à P.

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\underbrace{Q_\phi}}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

V vérifie que $l'_{3,1}(r_3) = L_1(l'_{3,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $l_{3,3}(r_1, x_2, r_3)$ à P.

P envoie $l'_{3,3}(x_2)$ sensé être $l_{3,3}(r_1, x_2, r_3)$

PSPACE $\subseteq$ IP : Protocole

$$S = \underbrace{\exists_1}_{q_0} \underbrace{L_1}_{\ell_{1,1}} \underbrace{\forall_2}_{q_1} \underbrace{L_1}_{\ell_{2,1}} \underbrace{L_2}_{\ell_{2,2}} \underbrace{\exists_3}_{q_2} \cdots \underbrace{\forall_n}_{q_{n-1}} \underbrace{L_1}_{\ell_{n,1}} \cdots \underbrace{L_n}_{\ell_{n,n}} \underbrace{Q_\phi}_{q_n}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

V vérifie que $l'_{3,1}(r_3) = L_1(l'_{3,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $l_{3,3}(r_1, x_2, r_3)$ à P.

P envoie $l'_{3,3}(x_2)$ sensé être $l_{3,3}(r_1, x_2, r_3)$

V vérifie que $l'_{3,2}(r_1) = L_2(l'_{3,3})(r_2)$. Génère un nouveau $r_2 \in \mathbb{F}_p$ et demande $q_3(r_1, r_2, x_3)$

PSPACE $\subseteq$ IP : Protocole

$$S = \underset{q_0}{\xrightarrow{\exists_1}} \underset{\ell_{1,1}}{\xrightarrow{L_1}} \underset{q_1}{\xrightarrow{\forall_2}} \underset{\ell_{2,1}}{\xrightarrow{L_1}} \underset{\ell_{2,2}}{\xrightarrow{L_2}} \underset{q_2}{\xrightarrow{\exists_3}} \cdots \underset{q_{n-1}}{\xrightarrow{\forall_n}} \underset{\ell_{n,1}}{\xrightarrow{L_1}} \cdots \underset{\ell_{n,n}}{\xrightarrow{L_n}} \underset{q_n}{\underline{\;Q_\psi\;}}$$

FIGURE 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

V vérifie que $l'_{3,1}(r_3) = L_1(l'_{3,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $l_{3,3}(r_1, x_2, r_3)$ à P.

P envoie $l'_{3,3}(x_2)$ sensé être $l_{3,3}(r_1, x_2, r_3)$

V vérifie que $l'_{3,2}(r_1) = L_2(l'_{3,3})(r_2)$. Génère un nouveau $r_2 \in \mathbb{F}_p$ et demande $q_3(r_1, r_2, x_3)$

...

PSPACE $\subseteq$ IP : Protocole

$$S = \xrightarrow[q_0]{\exists_1} \xrightarrow[\ell_{1,1}]{L_1} \xrightarrow[q_1]{\forall_2} \xrightarrow[\ell_{2,1}]{L_1} \xrightarrow[\ell_{2,2}]{L_2} \xrightarrow[q_2]{\exists_3} \cdots \xrightarrow[q_{n-1}]{\forall_n} \xrightarrow[\ell_{n,1}]{L_1} \cdots \xrightarrow[\ell_{n,n}]{L_n} \underset{q_n}{\underline{Q_\psi}}$$

Figure 10.2 – Polynômes successifs dans l'arithmétisation de φ.

P envoie $l'_{3,2}(x_1)$ sensé être $l_{3,2}(x_1, r_2, r_3)$

V vérifie que $l'_{3,1}(r_3) = L_1(l'_{3,2})(r_1)$. Génère un nouveau $r_1 \in_R \mathbb{F}_p$ et demande $l_{3,3}(r_1, x_2, r_3)$ à P.

P envoie $l'_{3,3}(x_2)$ sensé être $l_{3,3}(r_1, x_2, r_3)$

V vérifie que $l'_{3,2}(r_1) = L_2(l'_{3,3})(r_2)$. Génère un nouveau $r_2 \in \mathbb{F}_p$ et demande $q_3(r_1, r_2, x_3)$

...

V accepte ssi $q'_n(r_n) = Q_\psi(r_1, ..., r_n)$

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \notin \mathrm{QBF}$.

- Si P est honnête, $q'_0 = 0$ et V refusera.

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \notin \mathrm{QBF}$.

- Si P est honnête, $q'_0 = 0$ et V refusera.
- Au dernier tour du protocole, P envoie $q'_n(x_n)$ sensé être $q_n(r_1, ..., r_{n-1}, x_n) = Q_\psi(r_1, ..., r_{n-1}, x_n)$. Or P ne connaît pas le r_n que V va utiliser pour tester l'étalité. Ces deux polynômes sont de degré maximal m (le nombre de clauses) donc par le Lemme la probabilité que V trouve qu'ils sont égaux est $\leq m/p$.

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \notin \mathrm{QBF}$.

- Si P est honnête, $q_0' = 0$ et V refusera.
- Au dernier tour du protocole, P envoie $q_n'(x_n)$ sensé être $q_n(r_1, ..., r_{n-1}, x_n) = Q_\psi(r_1, ..., r_{n-1}, x_n)$. Or P ne connaît pas le r_n que V va utiliser pour tester l'étalité. Ces deux polynômes sont de degré maximal m (le nombre de clauses) donc par le Lemme la probabilité que V trouve qu'ils sont égaux est $\leq m/p$.
- Dans tout le reste du protocole, V teste la cohérence de P en évaluant des polynômes univariés de devré 1. Donc à chaque fois il se trompe avec probabilité $\leq 1/p$.

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \notin \mathrm{QBF}$.

- Si P est honnête, $q'_0 = 0$ et V refusera.
- Au dernier tour du protocole, P envoie $q'_n(x_n)$ sensé être $q_n(r_1, ..., r_{n-1}, x_n) = Q_\psi(r_1, ..., r_{n-1}, x_n)$. Or P ne connaît pas le r_n que V va utiliser pour tester l'étalité. Ces deux polynômes sont de degré maximal m (le nombre de clauses) donc par le Lemme la probabilité que V trouve qu'ils sont égaux est $\leq m/p$.
- Dans tout le reste du protocole, V teste la cohérence de P en évaluant des polynômes univariés de devré 1. Donc à chaque fois il se trompe avec probabilité $\leq 1/p$.
- La probabilité que V se trompe à tout coup est donc

$$q = \frac{n(n+3)}{2} \cdot \frac{1}{p} + \frac{m}{p}$$

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \notin \mathrm{QBF}$.

Soit

$$D = 3\left(\frac{n(n+3)}{2} + m\right), T = \alpha m^2 2^{2n}$$

Alors p(premier) $\in [D, D+T] \implies q \leq 1/3$

PSPACE $\subseteq$ IP : Analyse du protocole

Supposons que $\phi \in \mathrm{QBF}$. Si les contraintes suivantes sont respectés alors le vérificateur accepte avec certitude :

- P choisi un premier p tel que $Q_\phi \neq 0$ sur $\mathbb{F}_p$
- P envoie à chaque fois les "bons" polynômes.

PSPACE $\subseteq$ IP : Conclusion

PSPACE $\subseteq$ IP car pour décider de l'appartenance d'une mot x à un langage $L \in$ PSPACE on peut d'abord réduire à une instance ϕ de QBF et ensuite effectuer le protocole précédant. Si $x \in L$, alors $\psi \in$ QBF et le protocole acceptera. Si $x \notin L$ alors $\psi \notin$ QBF et le protocole refusera avec probabilité $\geq 2/3$.

Conclusion

Preuve

IP $\subseteq$ PSPACE $\wedge$ PSPACE $\subseteq$ IP $\implies$ IP=PSPACE

Fun facts

- Si on permet au vérificateur d'interagir avec plusieurs prouveurs qui ne peuvent pas communiquer ensemble alors MIP=NEXP

Fun facts

- Si on permet au vérificateur d'interagir avec plusieurs prouveurs qui ne peuvent pas communiquer ensemble alors MIP=NEXP
- Si les multiples prouveurs peuvent se partager un nombre arbitraire de qubits intriqués, alors on a NEEXP $\subseteq$ MIP* (peut-être RE)!