$IP = PSPACE$

Robin Morisset

14 janvier 2010

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?
- On retrouve NP car le déterminisme limite à un round d'interaction

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?
- On retrouve NP car le déterminisme limite à un round d'interaction
- Rendre partiellement aléatoire une des machines permet des progrès

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?
- On retrouve NP car le déterminisme limite à un round d'interaction
- Rendre partiellement aléatoire une des machines permet des progrès
- On obtient une nouvelle classe de complexité : IP.

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?
- On retrouve NP car le déterminisme limite à un round d'interaction
- Rendre partiellement aléatoire une des machines permet des progrès
- On obtient une nouvelle classe de complexité : IP.
- Où se situe-t-elle par apport aux autres ?

- 2 Machines de Turing qui communiquent sont elles plus "fortes" qu'une seule ?
- On retrouve NP car le déterminisme limite à un round d'interaction
- Rendre partiellement aléatoire une des machines permet des progrès
- On obtient une nouvelle classe de complexité : IP.
- Où se situe-t-elle par apport aux autres ?
-

$$IP = PSPACE$$

Sommaire

Machine de Turing probabiliste

Machine de Turing probabiliste

Probabiliste : la machine reçoit une suite infinie de bits aléatoires en entrée supplémentaire.
Ceci revient à dire qu'elle peut choisir à chaque configuration entre 2 transitions via le tirage d'une pièce à pile ou face.

Machine de Turing probabiliste

Machine de Turing probabiliste

Probabiliste : la machine reçoit une suite infinie de bits aléatoires en entrée supplémentaire.
Ceci revient à dire qu'elle peut choisir à chaque configuration entre 2 transitions via le tirage d'une pièce à pile ou face.

BPP

Langage reconnu par une machine probabiliste, polynômiale en temps, avec une probabilité d'erreur $p < 1/2$.

Interactive Proofs

Définition

IP

On généralise BPP, en permettant à la machine de Turing (appelée $\mathcal{V}$: le vérificateur) de communiquer avec une autre (appelée $\mathcal{P}$: le prouveur).

Interactive Proofs

Définition

IP

On généralise BPP, en permettant à la machine de Turing (appelée $\mathcal{V}$: le vérificateur) de communiquer avec une autre (appelée $\mathcal{P}$: le prouveur).
$L \in IP$ si $\exists \mathcal{V}$ une machine de Turing probabiliste, polynômiale en temps telle que :

Interactive Proofs

Définition

IP

On généralise BPP, en permettant à la machine de Turing (appelée $\mathcal{V}$: le vérificateur) de communiquer avec une autre (appelée $\mathcal{P}$: le prouveur).
$L \in IP$ si $\exists \mathcal{V}$ une machine de Turing probabiliste, polynômiale en temps telle que :

- si $x \in L$, $\exists \mathcal{P}$ telle que en interagissant avec $\mathcal{P}$, $\mathcal{V}$ refuse x avec probabilité $p < 1/2$

Interactive Proofs

Définition

IP

On généralise BPP, en permettant à la machine de Turing (appelée $\mathcal{V}$: le vérificateur) de communiquer avec une autre (appelée $\mathcal{P}$: le prouveur).
$L \in IP$ si $\exists \mathcal{V}$ une machine de Turing probabiliste, polynômiale en temps telle que :

- si $x \in L$, $\exists \mathcal{P}$ telle que en interagissant avec $\mathcal{P}$, $\mathcal{V}$ refuse x avec probabilité $p < 1/2$
- si $x \notin L$, $\forall \mathcal{P}'$, en interagissant avec $\mathcal{P}'$, $\mathcal{V}$ accepte x avec probabilité $p < 1/2$

Interactive Proofs

Remarques sur la définition

$\mathcal{P}$

Aucune contrainte n'est imposée sur la complexité de $\mathcal{P}$. La difficulté n'est pas de trouver si $x \in L$, mais d'en convaincre $\mathcal{V}$.

Interactive Proofs

Remarques sur la définition

$\mathcal{P}$

Aucune contrainte n'est imposée sur la complexité de $\mathcal{P}$. La difficulté n'est pas de trouver si $x \in L$, mais d'en convaincre $\mathcal{V}$.

p

On peut itérer l'exécution de $< \mathcal{P}, \mathcal{V} > (x)$ un nombre constant de fois sans changer la complexité asymptotique de $\mathcal{V}$.
Ceci fournit un nouveau p' qui est une puissance de p.
En itérant suffisament de fois, on peut donc prendre p arbitrairement petit.

Interactive Proofs

Lien avec les autres classes de complexité

Théorème

$$IP = PSPACE$$

Interactive Proofs

Lien avec les autres classes de complexité

Théorème

$$IP = PSPACE$$

Ainsi, il ne sert en particulier à rien d'accroître la puissance de $\mathcal{P}$ au-delà de *PSPACE*

Interactive Proofs

Lien avec les autres classes de complexité

Théorème

$$IP = PSPACE$$

Ainsi, il ne sert en particulier à rien d'accroître la puissance de $\mathcal{P}$ au-delà de *PSPACE*

La démonstration se fait en démontrant les deux inclusions réciproques.

Sommaire

Preuve de $IP \subseteq PSPACE$

Soit $L \in IP$.
On va construire $\mathcal{M}$ une machine de Turing non déterministe en espace polynômiale acceptant L.
Puisque $PSPACE = NPSPACE$, on aura montré $IP \subseteq PSPACE$.

Preuve de $IP \subseteq PSPACE$

Soit $L \in IP$.
On va construire $\mathcal{M}$ une machine de Turing non déterministe en espace polynômiale acceptant L.
Puisque $PSPACE = NPSPACE$, on aura montré $IP \subseteq PSPACE$.

On simule $\mathcal{V}$ sur tous les bits aléatoires possibles (complexité en espace polynômiale, puisque $\mathcal{V}$ est en temps polynômial).
Quand $\mathcal{V}$ interroge $\mathcal{P}$, on "devine" la suite de réponse de $\mathcal{P}$ qui maximise la probabilité que $\mathcal{V}$ accepte.

Preuve de $IP \subseteq PSPACE$ (2)

On retient le nombre de fois où $\mathcal{V}$ accepte. $\mathcal{M}$ accepte si et seulement si $\mathcal{V}$ aurait accepté avec probabilité $> 1/2$.
Par définition de IP, $\mathcal{M}$ accepte si et seulement si $x \in L$.

Preuve de $IP \subseteq PSPACE$ (2)

On retient le nombre de fois où $\mathcal{V}$ accepte. $\mathcal{M}$ accepte si et seulement si $\mathcal{V}$ aurait accepté avec probabilité $> 1/2$.
Par définition de IP, $\mathcal{M}$ accepte si et seulement si $x \in L$.

On en conclue que

$$IP \subseteq PSPACE$$

Sommaire

Principe de la preuve de $PSPACE \subseteq IP$

On va montrer que $QSAT \in IP$.
$QSAT$ étant $PSPACE$-complet, cela suffit à démontrer que $PSPACE \subseteq IP$.

Principe de la preuve de $PSPACE \subseteq IP$

On va montrer que $QSAT \in IP$.
$QSAT$ étant $PSPACE$-complet, cela suffit à démontrer que $PSPACE \subseteq IP$.

On transforme les formules quantifiées en formules arithmétiques assez facilement en temps polynômial.
L'intérêt est de pouvoir les évaluer sur des valeurs autres que 0 et 1.

Principe de la preuve de $PSPACE \subseteq IP$ (2)

On instaure ensuite un protocole par lequel $\mathcal{P}$ va aider $\mathcal{V}$ à "éplucher" les quantificateurs un à uns.
Pour ce faire il va envoyer à chaque fois à $\mathcal{V}$ un polynôme correspondant à la formule où on ne fait varier que la variable considérée.
$\mathcal{V}$ vérifie la cohérence par rapport aux polynômes précédemment échangés, et fixe pour la suite cette variable à un nombre aléatoire tiré entre 1 et k (k une constante).

Principe de la preuve de $PSPACE \subseteq IP$ (3)

Puisque $\mathcal{P}$ ne peut pas prédire les nombres tirés par $mathcalV$, plus k est grand et moins il y a de chances pour que la valeur du polynôme qu'un $\mathcal{P}'$ malhonnête envoie coïncide avec celle du polynôme voulu.
k étant arbitrairement choisi, il suffit de le prendre suffisament grand par rapport à n pour maintenir $p < 1/2$.

Principe de la preuve de $PSPACE \subseteq IP$ (3)

Puisque $\mathcal{P}$ ne peut pas prédire les nombres tirés par $mathcalV$, plus k est grand et moins il y a de chances pour que la valeur du polynôme qu'un $\mathcal{P}'$ malhonnête envoie coïncide avec celle du polynôme voulu.
k étant arbitrairement choisi, il suffit de le prendre suffisament grand par rapport à n pour maintenir $p < 1/2$.

Donc

$$QSAT \in IP$$

Remarque sur la preuve

Complexité en temps de $\mathcal{V}$

La complexité en temps polynômiale de $\mathcal{V}$ dépend du degré des polynômes manipulés.
Malheureusement, si on procède naïvement, celui-ci explose exponentiellement.
Il faut donc "réduire" la formule entre chaque quantificateur en remarquant que sur $\{0,1\}$, $x_i^k = x_i$ pour $k \geq 1$.

Références

IP = PSPACE : Simplified Proof de A. Shen (ACM, mai 1992)

Notes on complexity theory de Jonathan Katz, chapitres 10 et 11 (novembre 2005)