

PCP = Probabilistically Checkable Proof

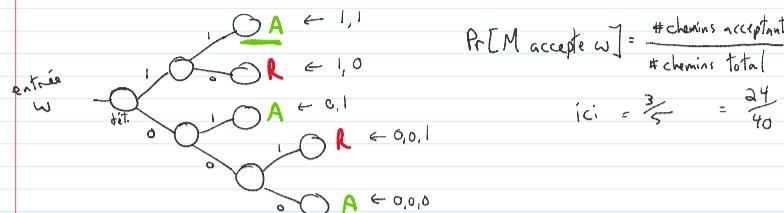
MT probabiliste: MT qui peut lancer une pièce p telle que
 $\Pr[p=0] = \frac{1}{2}$
 $\Pr[p=1] = \frac{1}{2}$

La MT peut prendre une décision selon le résultat

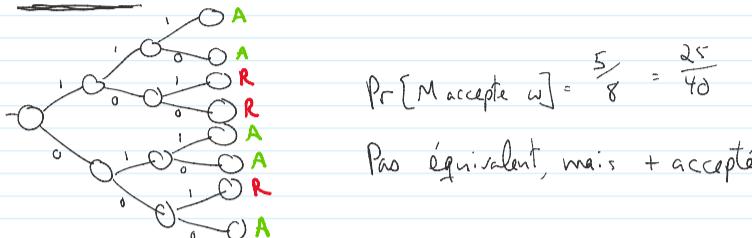
Fonctionnement interne

La MT M a accès à un oracle qui, en temps $O(1)$, écrit 0 ou 1 à l'emplacement de la tête

chacun avec prob. $\frac{1}{2}$



Pour simplifier, on suppose que chaque chemin a la m^e longueur.



Vérificateur probabiliste pour un langage L .

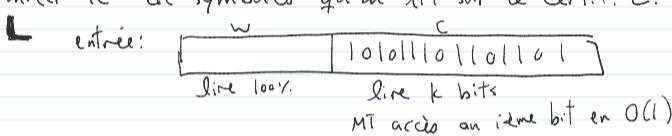
On dit qu'une MT prob. V est un vérif. polynomial prob. pour L si:

- ① $w \in L \Rightarrow \exists c \text{ t.g. } \Pr[V \text{ accepte } \langle w, c \rangle] = 1$ (NP)
- ② $w \notin L \Rightarrow \forall c \quad \Pr[V \text{ accepte } \langle w, c \rangle] \leq \frac{1}{2}$ (O)

Vérificateur PCP

Vérif. prob. avec limites

- limiter le # de lancer de pièces permis
- limiter le # de symboles qu'on lit sur le certif. c .



Déf: vérificateur PCP



Un langage L admet un vérif. $(r(n), g(n))$ -PCP

s'il existe une MT probabiliste M qui satisfait:

- ✓ ① sur entrée $\langle w, c \rangle$, M s'exécute en temps $O((|w|+|c|)^k)$
- ✓ ② M utilise $r(n)$ lancers de pièces, $n = |w|$
- ✓ ③ M lit $g(n)$ symboles de c , $n = |w|$
- ? ✓ ④ M est un vérificateur prob. pour L
 - si $w \in L$, $\Pr[M \text{ accepte } \langle w, c \rangle] = 1$ pour un certain c
 - si $w \notin L$, $\Pr[M \text{ accepte } \langle w, c \rangle] \leq \frac{1}{2} \quad \forall c$

Déf: On dit que $L \in \text{PCP}(r(n), g(n))$ s'il existe des constantes c, d tq. L admet un vérificateur $(c \cdot r(n), d \cdot g(n))$ -PCP.

On écrit parfois $L \in \text{PCP}(O(r(n)), O(g(n)))$.

Le théorème PCP: $NP = \text{PCP}(\log n, 1)$ O(1) bits de certificat

↑ O(log n) lancer de pièce

~1990: >100 pages | preuve ultra-profonde

~2000: >25 pages | preuve ultra-profonde

ex: $\text{PCP}(0, \log n) = P$

O aléatoire ↑ certif. ↑ $O(\log n)$

Thm: $\text{PCP}(0, \log n) = P$

- ① $P \subseteq \text{PCP}(0, \log n)$: soit $L \in P$. On montre que $L \in \text{PCP}(0, \log n)$.

Puisque $L \in P$, $\exists M$ en temps $O(n^k)$ qui décide L , $k \in \mathbb{N}$.

Il s'avère que M est aussi un vérificateur $\text{PCP}(0, O(\log n))$.

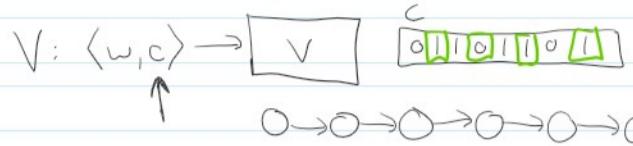
Soit $c = \emptyset$. Sur entrée $\langle w, c \rangle$, M roule en temps poly, utilise O bits aléatoires, lit $< O(\log n)$ bits de c .

Soit $c = \emptyset$. Sur entrée $\langle w, c \rangle$, M renale en temps poly, utilise 0 bits aléatoires, lit $\leq O(\log n)$ bits de c , et

- si $w \in L$, alors $\Pr[M \text{ accepte } \langle w, c \rangle] = 1$
- si $w \notin L$, alors $\Pr[M \text{ accepte } \langle w, c \rangle] = 0 \leq \frac{1}{2}$. Donc M satisfait toutes les cond. d'un vérif. PCP($O, O(\log n)$) $\Rightarrow L \in \text{PCP}(O, \log n)$.

(2) $\text{PCP}(O, \log n) \subseteq P$: soit $L \in \text{PCP}(O, \log n)$. On veut $L \in P$.

Vérif. \exists pour L qui utilise 0 aléatoire et lit $\leq d \log n$ bits d'un certificat passé.



Puisque V a 0 bits aléatoires, on peut supposer que les mêmes $d \log n$ bits de certificat lus sont tous les mêmes.

On peut donc supposer que tout certif. c a une taille $\leq d \log n$.

Il y a donc

$\leq 2^{d \log n}$ certificats possibles que V utilise.

Puisque $2^{d \log n} = (2^{\log n})^d = n^d$ est polynomial, on peut énumérer tous les certifi.

On peut décider L de façon déterministe comme suit:

sur entrée w :

pour chaque $c \in \{0, 1\}^{d \log n}$ //énumérer les $\leq n^d$ certif.

simuler V sur $\langle w, c \rangle$

si V accepte, accepter

sinon rejeter

// si: $w \in L$

// $\Pr[V \text{ accepte}] = 1$

// sinon $\Pr[V \text{ accepte}] \leq \frac{1}{2}$

Ceci montre que $L \in P$. ■

Notation: $\text{PCP}(f(n), \text{poly}(n)) = \bigcup_k \text{PCP}(f(n), n^k)$

Ex: $\text{PCP}(O, \text{poly}(n)) = \text{NP}$

(1) $\text{NP} \subseteq \text{PCP}(O, \text{poly}(n))$.

Soit $L \in \text{NP}$. Alors \exists vérif. V pour L tel que

V utilise aucun bit aléatoire (car V est une MT dét.)

et lit tout un certificat de taille $\text{poly}(n)$.

C'est donc un vérificateur $\text{PCP}(O, \text{poly}(n))$.

(2) $\text{PCP}(O, \text{poly}(n)) \subseteq \text{NP}$.

Soit $L \in \text{PCP}(O, \text{poly}(n))$. Vérif. V pour L qui utilise 0 aléatoire (par déf.) et qui lit $\text{poly}(n)$ bits d'un certificat. C'est donc c'est un vérificateur pour NP.

Ex: $\text{PCP}(\log n, 1) \subseteq \text{NP}$.

Soit $L \in \text{PCP}(\log n, 1)$. Donc \exists vérif. prob. V qui utilise

$\leq d \log n$ bits aléatoires et $\leq q$ bits de certificat sur chaque exécution possible, $d, q \in \mathbb{N}$, et tel que

$\rightarrow w \in L \Rightarrow \exists c \text{ tq. } \Pr[V \text{ accepte } \langle w, c \rangle] = 1$
 $w \notin L \Rightarrow \forall c \text{ tq. } \Pr[V \text{ accepte } \langle w, c \rangle] \leq \frac{1}{2}$.

V peut prendre un chemin d'exéc. différent pour chaque série de lancer de $d \log n$ piées. Il y a donc $\leq 2^{d \log n}$ chemins d'exécution possibles, un pour chaque $r \in \{0, 1\}^{d \log n}$.

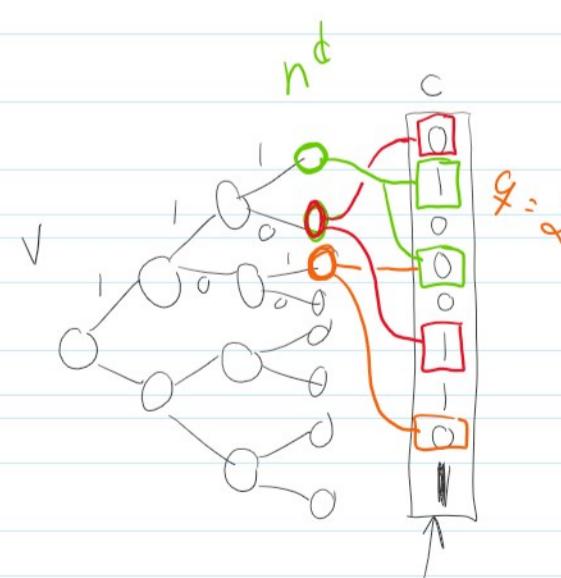
Ce nb de chemins est donc borné par $2^{d \log n} = n^d$.

Chaque chemin lit $\leq q$ bits du certif. c passé

\Rightarrow il y n^d chemins $\cdot q$ bits lus \Rightarrow on peut supposer que $|c| \leq q \cdot n^d$ (poly)

On peut faire un MT non-dét., pour L comme suit:

sur entrée $w \in \Sigma^*$



Un peu plus loin pour la suivante

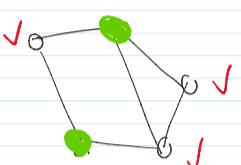
sur entrée $w \in \Sigma^*$

- deviner le certificat c qui fait que V accepte $\langle w, c \rangle$ si $w \in L$.
(c'est possible car $|c| \in O(n^d)$)
- pour chaque $r \in \{0, 1\}^{d \log n}$, simuler V sur entrée $\langle w, c \rangle$
- si V a tjs accepté $\langle w, c \rangle$, alors accepter w // $\Pr[V \text{ accepte}] = 1$
sinon rejeter // $\Pr[V \text{ accepte}] \leq \frac{1}{2}$ &c.

$\Rightarrow L \in NP$.

Pistes sur devoir

4a)



① borne triviale sur CPT

$\rightarrow OPT \geq ?$

② algo qui retourne une sol. APP

tg. APP \leq ratio vu du * bonne
 $(k+1) \cdot$ bonne

4b: (facile)

min clauses insatif.

indice: SAT est NP-difficile

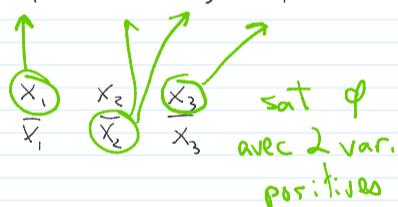
4c: IND-SET n'a pas de α -approx

pour une certaine constante α

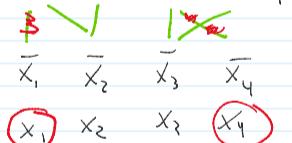
(conséquence de l'inapprox
de MAX-CLIQUE)

MAT-TRJE-SAT

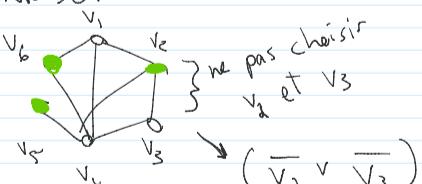
$$\varphi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$



$$\varphi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$



IND-SET



5.a) trivial

Une phrase suffit à me convaincre que vous avez compris

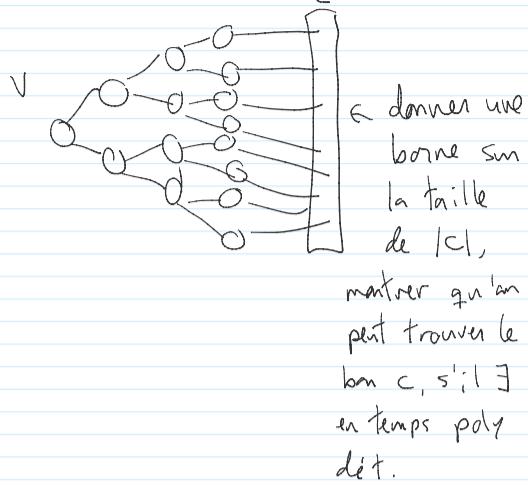
b) Considérez énumérer les

$2^{d \log n}$ chaînes de bits aléatoires possibles.

c) si SAT admet un vérif. prob.

qui utilise $\log(\log n)$ bits aléatoires, et lit q bits d'un certif. alors

qui utilisent au moins bits aléatoires, et lit q bits d'un certif. alors on peut résoudre SAT en temps polynomial déterministe.



Lien entre PCP et inapprox.

<1990

Thm: si $NP = PCP(\log n, 1)$, alors le problème Max- q -CSP n'a pas de $\frac{1}{2}$ -approx si $P \neq NP$, $q \in \mathbb{N}$.

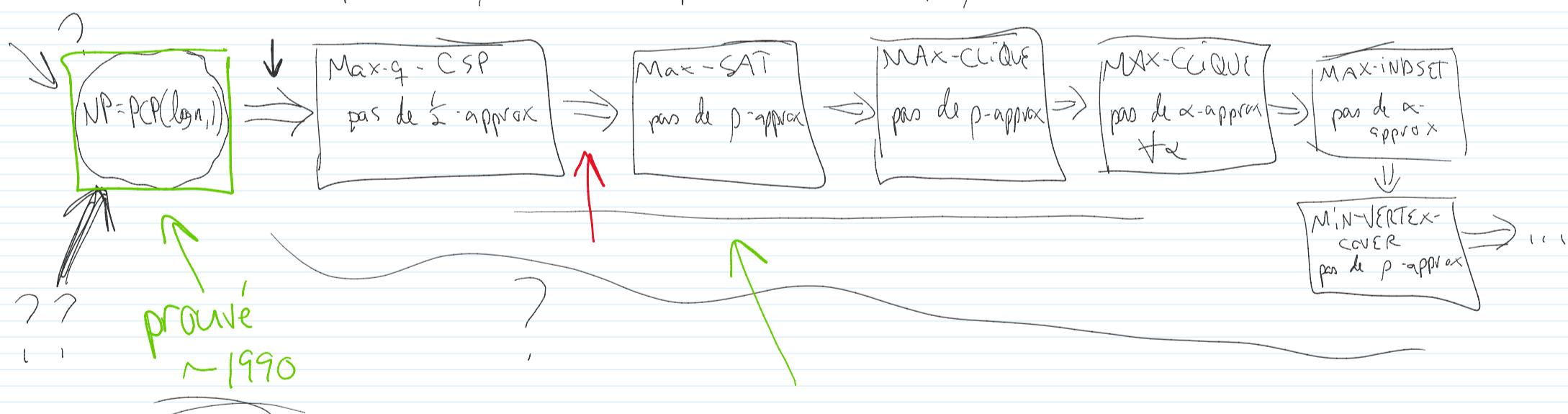
Max- q -CSP: entrée $\varphi = \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \dots \wedge \varphi_n$ (φ_i : clauses)

où chaque φ_i est une formule booléenne arbitraire, et chaque φ_i contient $\leq q$ variables.

ex: $\varphi_i = (x_1 \wedge (\bar{x}_2 \vee x_3) \oplus x_4)$ dans SAT: $\varphi_i = (x_1 \vee \bar{x}_2 \vee x_3)$

Bnf: assignation des x_i qui max. # φ_i satisfaites

Thm: si Max- q -CSP n'a pas de $\frac{1}{2}$ -approx, alors MAX-SAT n'a pas de p -approx pour une constante p .



Thm: si $NP = PCP(\log n, 1)$, alors Max- q -CSP n'a pas de $\frac{1}{2}$ -approx

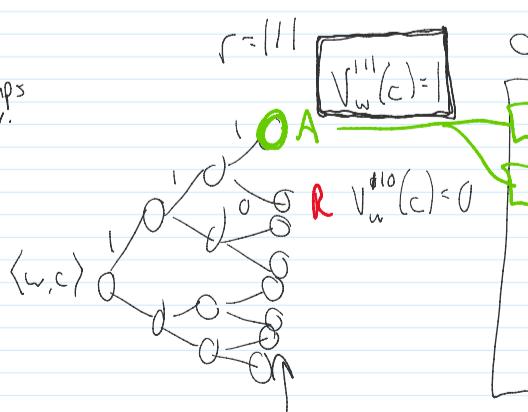
Soit $L \in NP$. Si Max- q -CSP avait une $\frac{1}{2}$ -approx, on déciderait L en temps poly.

Puisque $L \in NP$, alors $L \in PCP(\log n, 1)$ (par l'égalité).

Donc L a un vérif. pr.($d \log n, q$)-PCP appelé V .

Soit $w \in \Sigma^*$.

Pour $r \in \{0, 1\}^{d \log n}$. Soit V_w^r une fonction booléenne telle que $V_w^r(c) = \begin{cases} 1 & \text{si } V \text{ accepte } (w, c) \text{ quand ces bits aléatoires sont } r \\ 0 & \text{sinon} \end{cases}$



Puisque V consulte $\leq q$ bits de c , alors V_w^r dépend de seulement q bits d'entrée du certificat. Donc

→ V_w^r est une fct booléenne qui utilise q variables booléennes.

ne servement à être un casse au certificat pour

→ V_w^r est une fct booléenne qui utilise q variables booléennes.

→ On peut voir V_w^r comme une formule bool φ_w^r sur q variables.

Soyent $r_1, r_2, \dots, r_{2^{\log n}} = \{0, 1\}^{d \log n}$ et

$$\varphi = V_w^{r_1} \wedge V_w^{r_2} \wedge \dots \wedge V_w^{r_{2^{\log n}}} = \varphi_w^{r_1} \wedge \varphi_w^{r_2} \wedge \dots \wedge \varphi_w^{r_{2^{\log n}}}$$

φ est une instance de MAX-q-CSP.

Si $w \in L$, alors $\exists c \text{ t.q. } \Pr[V \text{ accepte } \langle w, c \rangle] = 1$

⇒ T'assignation t.q. $V_w^{r_1} = 1, V_w^{r_2} = 1, V_w^{r_3} = 1, \dots$

⇒ on peut satisfaire chaque φ_i .

Si $w \notin L$, alors $\forall c \quad \Pr[V \text{ accepte } \langle w, c \rangle] \leq \frac{1}{2}$

⇒ T'assignation, $\leq \frac{1}{2}$ des $V_w^{r_i}$ pourront être satisfait.

Dmc, une $\frac{1}{2}$ -approx nous permettrait de distinguer si $w \in L$ ou non.

gap ↴