Complexité des classes randomisées

Jérôme Casse

19 décembre 2008

Résumé

Cet article traite des classes randomisées. Dans un premier temps, il étend la notion de machine de Turing à celle de machine de Turing randomisée ce qui permet ainsi d'introduire de nouvelles classes de complexité. Et, enfin, l'article finit par la présentation de théorèmes qui relient les classes de complexité introduites entre elles et à celles de la hiérarchie polynomiale.

Première partie
Les classes randomisées

Définition 1 (machine de Turing randomisée)**.** Une machine de Turing randomisée est une machine de Turing classique qui possède en plus une bande (bande d'aléa) en lecture seule qui contient des caractères déjà écris qui ont été tirés, préalablement, de manière aléatoire sur cette bande.

On notera $M(x,r)$ le résultat de la machine randomisée M sur l'entrée x avec la bande d'aléa qui contient le mot r.

Définition 2 (classe RTIME)**.** La classe RTIME$(f(n), l(n), \mathrm{acc}(n), \mathrm{rej}(n))$ est la classe des langages L reconnaissables par une machine de Turing randomisée en un temps $f(n)$ avec une longueur de bande d'aléa $l(n)$ et telle que :
- si $x \in L$, alors $P[M(x,r) = \text{Faux}] \leq \mathrm{acc}(n)$
- si $x \notin L$, alors $P[M(x,r) = \text{Vrai}] \leq \mathrm{rej}(n)$

Remarque 1.
- la probabilité est prise sur tous les aléas r de longeur $l(n)$.
- $l(n) \leq f(n)$ (car l'espace accessible est toujours inférieur au temps).

1 La classe RP : randomisée polynomiale

Définition 3 (classe RP)**.** RP $= \bigcup_{k=0}^{\infty} \mathrm{RTIME}(n^k, n^k, 0, 1/2)$ ie la classe des langages tels qu'il existe une machine de Turing M randomisée qui calcule en temps polynomial en la taille de l'entrée (avec une longueur de bande d'aléa également polynomiale) et tels que

– si $x \in L$, alors $P[M(x,r) = \text{Vrai}] \geq 1/2$
– si $x \notin L$, alors $M(x,r) = \text{Faux}$

Nous allons voir par le lemme suivant que le $1/2$ dans la définition à peu d'importance. En effet :

Lemme 4. *$\forall p$ polynôme, on a :*

$$RP = \bigcup_{n=0}^{\infty} RTIME(n^k, n^k, 0, 1/2^{p(n)})$$

Démonstration. Soit p un polynôme quelconque. Soit $L \in \text{RP}$. Soit M la machine reconnaissant ce langage en temps $t(n)$ avec longueur de bande d'aléa $l(n)$ (t et l sont deux polynômes). Soit la machine M' qui prend en entrée un mot x de taille n et dont la bande d'aléa est de longueur $p(n)l(n)$ (donc polynomiale). On découpe la bande d'aléa en $r_1 r_2 \dots r_{p(n)}$. Le résultat renvoyé par M' est $\bigwedge_{i=1}^{p(n)} M(x, r_i)$. M' calcule en $O(p(n)t(n))$ (donc en temps polynomial). Donc $L \in \text{RTIME}(O(p(n)t(n)), p(n)l(n), 0, 1/2^{p(n)}) \subseteq \text{RP}$. Donc $\text{RP} \subseteq \bigcup_{n=0}^{\infty} \text{RTIME}(O(p(n)t(n)), p(n)l(n), 0, 1/2^{p(n)}) \subseteq \text{RP}$. □

Corollaire 5. *$\forall \epsilon \in]0,1[$, on a :*

$$RP = \bigcup_{n=0}^{\infty} RTIME(n^k, n^k, 0, \epsilon)$$

Démonstration. Si $\epsilon \leq 1/2$, évident.

Sinon, en remarquant que le $1/2$ de la démonstration précédente peut être remplacé par ϵ, en prenant $p(n) = \lceil -1/\log_2 \epsilon \rceil$ (où $\lceil x \rceil$ désigne le plafond de x, c'est à dire $\lceil x \rceil = \inf\{n \in \mathbb{Z} | n \geq x\}$), on obtient :

$$\begin{aligned} \text{rej} &= \epsilon^{\lceil -1/\log_2 \epsilon \rceil} \\ &\leq (2^{\log_2 \epsilon})^{-1/\log_2 \epsilon} \\ &= 1/2 \end{aligned}$$

Cela est vrai à partir d'un certain rang. Mais cela suffit car avant ce rang, les mots du langages sont en nombre fini et donc décidables en temps polynomial. □

Définition 6 (co-RP)**.** La classe co-RP est la classes des langages dont le complémentaire est dans RP, on a évidemment : $\text{co-RP} = \bigcup_{k=0}^{\infty} \text{RTIME}(n^k, n^k, 1/2, 0)$

La classe RP est aussi appelé classe de *Monte Carlo*.

2 La classe ZPP

Définition 7 (classe ZPP)**.** $\text{ZPP} = \text{RP} \cap \text{co-RP}$

Le lemme suivant est un joli lemme optimiste. En effet, il énonce qu'un problème dans ZPP se résout en moyenne en temps polynomial.

Lemme 8. *ZPP est la classe des langages décidables en temps moyen polynomial à l'aide d'une machine de Turing qui ne fait aucune erreur.*

Démonstration. On dispose de 2 machines de Turing randomisées M_1 etM_2 qui reconnaissent respectivement L et L^c. On construit une machine de Turing M ainsi :

Input x
1.Tirer un aléa r pour M_1 (de manière uniforme)
2.Si $M_1(x,r) =$ Vrai alors
3. accepter
4.Sinon
5. Tirer un aléa r pour M_2
6. Si $M_2(x,r) =$ Faux alors
7. rejeter
8. Sinon
9. revenir en 1

La probabilité qu'elle s'arrête après k tours de boucle est $1-(1/2)^k$. Et son temps d'arrêt moyen est $\sum_{k=0}^{\infty} \frac{kp(n)}{2^n} = 2p(n)$. Il est donc polynomial en moyenne et son temps moyen est 2 fois celui de $p_1(n) + p_2(n)$.

La réciproque est vraie, mais fastidieuse et moins intéressante. □

C'est ce lemme qui justifie le nom ZPP (Zero Probality of error Polynomial time : en temps polynomial avec une probabilité nulle de se tromper). Cette classe est aussi appelée la classe des langages *Las Vegas*.

3 La classe BPP

Définition 9 (classe BPP). BPP $= \bigcup_{k=0}^{\infty}$ RTIME$(n^k, n^k, 1/3, 1/3)$ ie la classe des langages reconnaissables par une machine randomisée qui se trompe au plus une fois sur trois quand elle donne un résultat.

Lemme 10. *co-BPP=BPP*

Proposition 11. $\forall q$, *polynôme* $BPP = \bigcup_{k=1}^{\infty} RTIME(n^k, n^k, e^{-q(n)}, e^{-q(n)})$

Nous avons pour cela besoin de montrer le lemme suivant dû à Chernoff.

Lemme 12 (Chernoff). *Soient $X_1, \ldots, X_N$, N variables aléatoires indépendantes de Bernoulli de paramètre p ($P[X_1 = 1] = p$ et $P[X_1 = 0] = 1-p$). Alors, pour tout $\theta \in \mathbb{R}^+$,*

$$P[X_1 + \ldots + X_N \geq (1+\theta)pN] \leq e^{-c(\theta)pN}$$

où $c(\theta)$ est une fonction croissante de θ telle que, de plus, $c(\theta)/(1+\theta)$ est aussi croissante et $c(\theta) \geq \frac{\theta^2}{3}$ pour tout θ,$0 \leq \theta < 1$.

Démonstration. Par l'inégalité de Markov, on obtient :

$$\forall t, k > 0,\ P[e^{t(X_1+\ldots+X_N)} \geq kE(e^{t(X_1+\ldots+X_N)})] \leq \frac{1}{k}$$

En appliquant ce résultat à $k = \frac{e^{t(1+\theta)pN}}{E(e^{t(X_1+\ldots+X_N)})}$, on obtient :

$$P[e^{t(X_1+\ldots+X_N)} \geq e^{t(1+\theta)pN}] \leq \frac{E(e^{t(X_1+\ldots+X_N)})}{e^{t(1+\theta)pN}}$$

Comme ln est strictement croissante et que les X_i sont indépendants, l'inéquation devient :

$$\begin{array}{rcl} P[(X_1+\ldots+X_N) \geq (1+\theta)pN] & \leq & \prod_{i=1}^{N} E(e^{tX_i})e^{-t(1+\theta)pN} \\ & = & \prod_{i=1}^{N}(pe^t+(1-p))e^{-t(1+\theta)pN} \\ & = & (1+p(e^t-1))^N e^{-t(1+\theta)pN} \\ & \leq & e^{(e^t-1)pN}e^{-t(1+\theta)pN} \end{array}$$

Pour $t = \ln(1+\theta)$, on obtient :

$$P[(X_1+\ldots+X_N) \geq (1+\theta)pN] \leq e^{\theta pN}e^{-\ln(1+\theta)(1+\theta)pN} = e^{pN(\theta-(1+\theta)\ln(1+\theta))}$$

Posons $c(\theta) = -(\theta-(1+\theta)\ln(1+\theta))$. Vérifions que c vérifie les conditions.

- c est croissante car $c'(\theta) = -(1-(1*\ln(1+\theta)+\frac{1+\theta}{1+\theta}) = \ln(1+\theta) \geq 0$
- $c(\theta)/(1+\theta) = -\frac{\theta}{1+\theta}+\ln(1+\theta)$ est croissante car de dérivée $\frac{-1}{(1+\theta)^2}+\frac{1}{1+\theta} = \frac{\theta}{(1+\theta)^2} \geq 0$
- il reste à montrer que $c(\theta) \geq \frac{\theta^2}{3}$, ie $\theta-(1+\theta)\ln(1+\theta) \leq -\frac{\theta^2}{3}$ pour $0 \leq \theta < 1$. Comme $0 \leq \theta < 1$, on a :

$$\begin{array}{rcl} \ln(1+\theta) & = & \theta-\frac{\theta^2}{2}+\frac{\theta^3}{3}-\ldots \text{(developpement en série entière de ln)} \\ (1+\theta)\ln(1+\theta) & = & \theta+\theta^2(\frac{1}{2}-1)+\theta^3(\frac{1}{3}-\frac{1}{2})+\ldots \\ \theta-(1+\theta)\ln(1+\theta) & = & -\frac{\theta^2}{2}+\frac{\theta^3}{6}-\ldots \end{array}$$

Cette série converge (car elle converge normalement, le terme général étant en $O(1/n^2)$). De plus, elle est alternée, donc $\theta-(1+\theta)\ln(1+\theta) \leq -\frac{\theta^2}{2}+\frac{\theta^3}{6} \leq -\frac{\theta^2}{2}+\frac{\theta^2}{6} = -\frac{\theta^2}{3}$ (car $0 \leq \theta < 1$ et donc $\theta^3 \leq \theta^2$).

□

Démonstration. Supposons que L soit dans BPP. Soit q, un polynôme. Soit M la machine de Turing randomisée qui décide L en temps polynomial et en se trompant au plus une fois sur trois.

Construisons une machine M' qui décide L en temps polynomial et en se trompant au plus une fois sur $e^{q(n)}$.

Input x

1.On fait tourner M sur $N = 36q(n)$ bandes d'aléa $r_1, \ldots, r_N$ iid
(indépendantes et identiquement distribuées, la distribution étant la

distribution uniforme) avec en entrée x. On obtient donc une suite $X_1, \ldots, X_N$ de résultats iid (pour simplifier les notations, on notera $X_i = 1$ si $X_i =$ Vrai et $X_i = 0$ sinon).

2. On accepte si la majorité des résultats sont vrais (ie $\text{card}\{X_i = \text{Vrai}\} = \sum_{i=1}^{N} X_i \geq N/2$).

La machine ainsi construite s'exécute en temps polynomiale ($O(q(n)p(n))$). Montrons qu'elle se trompera au plus une fois sur $e^{q(n)}$. Supposons $x \notin L$ (pour $x \in L$, il suffit de prendre $X_i = 1 - X_i$), alors la probabilité que M' se trompe est $P[\sum_{i=1}^{N} X_i \geq N/2]$. Appliquons, donc le lemme de Chernoff pour $p \leq 1/3$ et $\theta = 1/2p - 1 \geq 1/2$, on obtient :

$$\begin{aligned}
P[\textstyle\sum_{i=1}^{N} X_i \geq N/2] &\leq e^{-c(\theta)pN} \\
&= e^{-\frac{c(\theta)}{1+\theta}(1+\theta)pN} \text{ (on remarque que } (1+\theta)p = 1/2) \\
&\leq e^{-\frac{c(1/2)}{1+1/2}\frac{N}{2}} \text{ (car } \theta \geq 1/2 \text{ et } \tfrac{c(\theta)}{1+\theta} \text{ est croissante)} \\
&\leq e^{-\frac{(1/2)^2}{3*3}N} \text{ (car } c(1/2) \geq \tfrac{(1/2)^2}{3} \text{ est croissante)} \\
&= e^{-\frac{N}{36}} \\
&= e^{-q(n)}
\end{aligned}$$

□

Cette classe de complexité est aussi appelée *Atlantic City*

Deuxième partie

Complexité de ces classes avec la hiérarchie polynomiale

4 Hiérarchie polynomiale

Définition 13 (Hiérarchie polynomiale d'ordre 0, 1 et 2)**.** Nous allons, ici, nous intéresser aux trois premières classes de la hiérarchie polynomiale : P, NP, co-NP, $\sum_2^p$ et $\prod_2^p$ (pour une analyse plus détaillée de la hiérarchie polynomiale nous renvoyons à l'exposé de Vincent). Nous prendrons pour définition de ces 5 classes :

- $\sum_0^p = \prod_0^p = \text{P}$
- $\sum_1^p = \text{NP}$ est la classe des langages tels que $x \in L \Leftrightarrow \exists y$ (de taille polynomiale en la taille de x) $(x, y) \in L'$ ($L' \in$ P) ie x appartient à L si étant donné un certificat y (de taille polynomiale en la taille de x), il existe une machine de Turing déterministe qui prend en entrée x et y et détermine en temps polynomial si x appartient à L.
- $\prod_1^p =$ co-NP est la classe des langages tels que $x \in L \Leftrightarrow \forall y$ (de taille polynomiale en la taille de x) $(x, y) \in L'$ ($L' \in$ P).
- $\sum_2^p$ est la classe des langages tels que $x \in L \Leftrightarrow \exists z\ \forall y$ (z et y de taille polynomiale en la taille de x) $(x, y, z) \in L'$ ($L' \in$ P).

– $\prod_2^p$ = co-$\sum_2^p$ est la classe des langages tels que $x \in L \Leftrightarrow \forall z\ \exists y$ (z et y de taille polynomiale en la taille de x) $(x, y, z) \in L'$ ($L' \in$ P).

5 Inclusions simples

Les premières de ces inclusions concernent les classes P, RP, co-RP et ZPP.

Proposition 14. *$P \subseteq RP$*

Démonstration. $\mathrm{P} = \bigcup_{k=0}^{\infty} \mathrm{RTIME}(n^k, 0, 0, 0) \subseteq \mathrm{RP}$ □

Par complémentation, on a les corollaires suivants :

Corollaire 15.
– *$P \subseteq$ co-RP*
– *$P \subseteq ZPP$*

Maintenant, regardons les relations entre RP, co-RP et BPP.

Proposition 16. *$RP \subseteq BPP$*

Démonstration. $\forall k, \mathrm{RTIME}(n^k, n^k, 0, 1/3) \subseteq \mathrm{RTIME}(n^k, n^k, 1/3, 1/3)$, donc :

$$\begin{array}{rcl} \mathrm{RP} & = & \bigcup_{k=0}^{\infty} \mathrm{RTIME}(n^k, n^k, 0, 1/3) \\ & \subseteq & \bigcup_{k=0}^{\infty} \mathrm{RTIME}(n^k, n^k, 1/3, 1/3) \\ & = & \mathrm{BPP} \end{array}$$

□

Par complémentation, on en déduit :

Corollaire 17. *co-$RP \subseteq BPP$*

Intéressons-nous, maintenant, à la classe NP, RP et co-NP et co-RP.

Proposition 18. *$RP \subseteq NP$*

Démonstration. Soit $L \in$ RP. Soit $x \in L$, alors il existe une bande d'aléa r telle que $M(x, r) =$ Vrai (car $P[M(x, r) = \text{Vrai}] \geq \frac{1}{2}$, donc la moitié des bandes d'aléa de taille $q(n)$ vérifie cela, donc au moins 1 si $q(n) \neq 0$). □

Par complémentation, on en déduit :

Corollaire 19. *co-$RP \subseteq$ co-NP*

6 Théorème et preuve de BPP $\subseteq (\sum_2^p \cap \prod_2^p)$

Théorème 20. *BPP* $\subseteq (\sum_2^p \cap \prod_2^p)$

Pour prouver ce théorème nous allons avoir besoin de notations et de plusieurs lemmes. Dans la suite, on considèrera un langage L de BPP et M sa machine de Turing randomisée qui s'exécute en temps polynomial $p(n)$ avec une bande d'aléa polynomiale de taille polynomiale $q(n)$ et qui se trompe au plus 1 fois sur 2^n (possible grâce à Chernoff).

On notera $m = q(n)$ pour simplifier la démonstration et $|x|$ la longueur de la bande d'aléa x si x est une bande d'aléa ou le cardinal de l'ensemble x si x est un ensemble.

Lemme 21. *Soit x une entrée de taille n. On note $R = \{r | r \in \{0,1\}^m et M(x,r) = Vrai\}$ (c'est l'ensemble des bandes d'aléa de taille m qui réponde Vrai pour l'entrée x sur la machine M). On a la propriété suivante :*

- *Si $x \in L$, alors $|R| \geq (1 - \frac{1}{2^n})2^m$*
- *Si $x \notin L$, alors $|R| \leq (\frac{1}{2^n})2^m$*

Démonstration. Si $x \in L$, alors $\forall r \in \{0,1\}^m$,

$$\begin{array}{rcl} P[M(x,r) = \text{Vrai}] & = & 1 - P[M(x,r) = \text{Faux}] \\ & \geq & 1 - \frac{1}{2^n} \text{ (car } P[M(x,r) = \text{Faux}] \leq \frac{1}{2^n}) \end{array}$$

Ainsi, $\frac{|R|}{|\{0,1\}^m|} \geq (1 - \frac{1}{2^n})$. Comme $|\{0,1\}^n| = 2^m$, on a $|R| \geq (1 - \frac{1}{2^n})2^m$.

Idem mutatis mutandis, si $x \notin L$. □

Définition 22 (ou exclusif)**.** Soit t et c, deux chaines de m bits. On définit le ou exclusif bit à bit de t et c par :

$$t \oplus c = (t_i \oplus c_i)_{i \in \{1,\ldots,m\}} \text{ (avec } \begin{array}{c|c|c} \oplus & 0 & 1 \\ \hline 0 & 0 & 1 \\ \hline 1 & 1 & 0 \end{array} \text{)}$$

On note $t \oplus R = \{t \oplus r | r \in R\}$.

La fonction $\begin{array}{ccc} \{0,1\}^n & \rightarrow & \{0,1\}^n \\ c & \rightarrow & t \oplus c \end{array}$ est bijective (car involutive). Ainsi, $|R| = |t \oplus R|$.

De plus, à r fixé, si t est distribué uniformément sur $\{0,1\}^n$, alors $t \oplus r$ aussi.

Lemme 23. *Avec les notations introduites, nous pouvons énoncer le lemme suivant :*

Si $x \in L$, alors il existe $t_0, \ldots, t_{\lceil m/n \rceil} \in \{0,1\}^m$ tel que :

$$\bigcup_{i=0}^{\lceil m/n \rceil} (t_i \oplus R) = \{0,1\}^m$$

Si $x \notin L$, alors pour tout $t_0, \ldots, t_{\lceil m/n \rceil} \in \{0,1\}^m$, on a :

$$\bigcup_{i=0}^{\lceil m/n \rceil} (t_i \oplus R) \neq \{0,1\}^m$$

où la notation $\lceil x \rceil$ désigne le plafond de x, c'est à dire $\lceil x \rceil = inf\{n \in \mathbb{Z} | n \geq x\}$

Démonstration. Pour $x \in L$, on va faire une preuve probabiliste.
Tirons $t_0, \ldots, t_{\lceil m/n \rceil}$ uniformément sur $\{0,1\}^m$. Soit $c \in \{0,1\}^m$. On a :

$$\begin{array}{rcll} P[r \notin \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)] & \leq & \prod_{i=0}^{\lceil m/n \rceil} \frac{1}{2^n} & (\text{car } \forall i \ P[(t_i \oplus r) \notin R] \leq \frac{1}{2^n}) \\ & = & (\frac{1}{2^n})^{\lceil m/n \rceil + 1} & \\ & \leq & \frac{1}{2^{m+1}} & (\text{car } n\lceil m/n \rceil \geq nm) \end{array}$$

On en déduit alors :

$$\begin{array}{rcl} P[\exists r | r \notin \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)] & = & \sum_{r \in \{0,1\}^n} P[r \notin \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)] \\ & \leq & \sum_{r \in \{0,1\}^n} \frac{1}{2^{m+1}} \\ & = & \frac{1}{2} \end{array}$$

Donc, par passage à la négation, on a :

$$P[\forall r | r \in \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)] \geq \frac{1}{2}$$

On en déduit donc que plus de la moitié des $(\lceil m/n \rceil + 1)$-uplets $(t_0, \ldots, t_{\lceil m/n \rceil})$ vérifient cette propriété. En particulier, il en existe un.

Pour $x \notin L$, $|R| \leq 2^{m-n}$. $\forall (t_0, \ldots, t_{\lceil m/n \rceil})$, on a :

$$|\bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)| \leq |R| * |\{t_i | i = 0, \ldots, \lceil m/n \rceil\}| = (1 + \lceil m/n \rceil)2^{m-n}$$

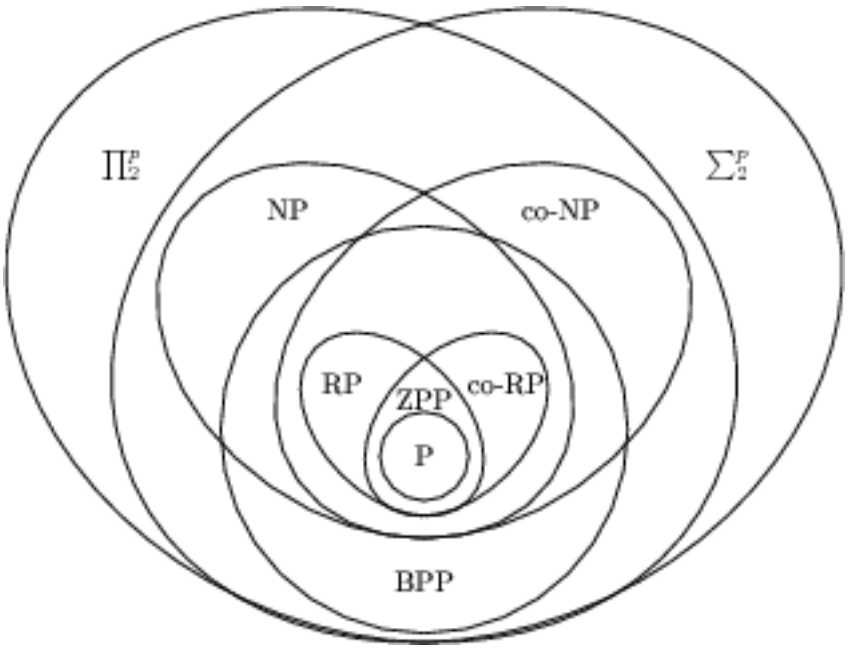
Ainsi la proportion des $c \in \{0,1\}^m$ tels que $c \in \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)$ est $(1 + \lceil m/n \rceil)/2^n = O(q(n)/n)$ (car $1 + \lceil m/n \rceil \leq 2 + m/n$).

Comme q est polynomiale en n, pour n assez grand, la proportion est plus petite que $1/2$, par exemple, et donc il existe $c \in \{0,1\}^{q(n)}$ tel que $c \notin \bigcup_{i=0}^{\lceil m/n \rceil}(t_i \oplus R)$ □

A partir ce cela nous pouvons, finalement, prouver l'inclusion cherchée.

Démonstration. On a prouvé par le lemme précédent que $x \in L$ ssi pour n assez grand, $\exists t_0, \ldots, t_{\lceil m/n \rceil} \in \{0,1\}^{q(n)}, \forall c \in \{0,1\}^{q(n)}, \bigvee_{i=0}^{\lceil q(n)/n \rceil} M(x, t_i \oplus c) = \text{Vrai}$.
Or $\bigvee_{i=0}^{\lceil q(n)/n \rceil} M(x, t_i \oplus c) = \text{Vrai}$ est calculable en temps polynomial en n, plus précisément en $(\lceil \frac{q(n)}{n} \rceil + 1)p(n) = O(q(n)p(n) + p(n))$.
Donc : $L \in \sum_2^p$ ce qui prouve que BPP $\subseteq \sum_2^p$. Par complémentarité, on en déduit BPP $\subseteq \prod_2^p$.
Ainsi : BPP $\subseteq (\sum_2^p \cap \prod_2^p)$ □

Nous pouvons réunir toute cette étude sur les classes de complexité randomisées dans ce dessin.



Quelques classes randomisées et leur relation avec les premières classes de la hiérarchie polynomiale

Références

[1] *http ://www.lsv.ens-cachan.fr/ goubault/Complexite/nl.pdf*, Jean GOUBAULT-LARRECQ

[2] *http ://www.lsv.ens-cachan.fr/ goubault/Complexite/pcp.pdf*, Jean GOUBAULT-LARRECQ