Théorie de la complexité

11 octobre 2010

Table des matières

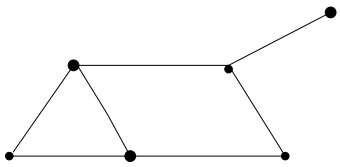
Chapitre 1

Introduction

1.1 Approche semi-formelle

- Factorisation des entiers :
  - instances du problème (données)
  - question

  $n \longrightarrow \bigcirc \longrightarrow p_1, k_1, \ldots, p_m, k_m$ (o $p_1^{k_1} \ldots p_m^{k_m} = n$)
- Calcul dans des graphes



$G = (S, A)$

S fini, $|S| = n$, $A \subset$ { paires de sommets }

Questions :

1. Existe-t-il un "triangle" ?
2. Quel est le plus court chemin de a vers b ?
3. Existe-t-il un circuit hamiltonien ? (tous les sommets une seule fois)

- Calculer un PGCD
- Calculer un exposant RSA

L'existence d'un algorithme efficace qui donne la réponse revient à dire que le problème est "facile".

"Définition" :

Un problème de décision est un problème dont la réponse est oui ou non.
But : Faire la différence entre :

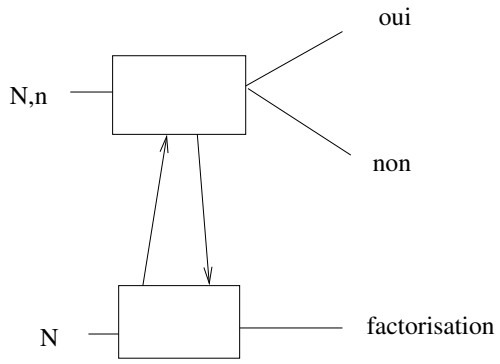
- Problèmes "faciles" : algorithme rapide
- Problèmes "difficiles" : pas d'algorithme rapide

entrée de taille n $\longrightarrow \bigcirc \longrightarrow$ sortie après un temps $\leqslant f(n)$
Un problème facile a un temps $\leqslant n^k$ avec k fixé (temps polynomial) et un problème difficile n'a pas de k qui vérifie cette condition.

Quelle est la taille de l'entrée ?

Elle est exprimée en nombre de bits (symboles).
Un graphe peut être représenté par sa matrice d'adjacence de taille n^2 mais si le graphe a peu d'arêtes on peut le représenter par des listes de voisins qui ont une taille de $n^2 \log n$ mais s'il y a peu d'arêtes on a en réalité $n \log n$.
On peut toujours se ramener à un problème de décision.

Exemple : la factorisation des entiers. On a en entrée N, M entier $\leqslant N$ de taille $n = \log N$.
Question : Existe-t-il un diviseur d de N tel que $1 < d \leqslant M$?



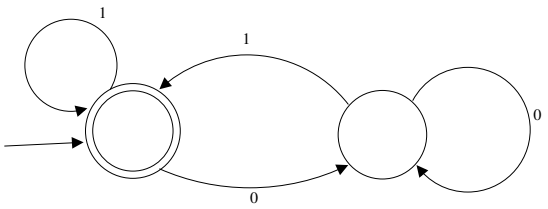
On utilise la dichotomie pour résoudre le problème initial en prenant $M = \frac{N}{2}$. On a $\log n$ questions à l'oracle pour d.

1.2 Passage du semi-formel au formel

Un problème de décision est un langage. Une instance est un élément de Σ^* où Σ^* est un alphabet fini (par exemple $\Sigma = \{0, 1\}$). Le langage $L \subset \Sigma$ est l'ensemble des instances "positives" (dont la réponse est oui). On doit reconnaitre L avec une machine.

Qu'est-ce qu'une machine ?

Une machine simple est un automate. Par exemple, l'automate qui suit accepte les mots se finissant par 1.



Un langage reconnaissable par un automate est un langage rationnel (régulier).

Machine de Turing :

-3	-2	-1	0	1	2	3	4
b	b	b	b	x_1	x_2	x_3	b

alphabets : Σ et $\Gamma \supset \Sigma \cup \{b\}$

Opération élémentaire (transition) :
- La machine est dans l'état q_i et lit le symbole $s \in \Gamma$.
- Elle écrit t (au lieu de s), $t \in \Gamma$.
- Elle se déplace de $\{-1, 0, 1\}$.
- Elle change d'état $q_i \to q_j$, $Q = \{q_0, \ldots, q_n, q_A, q_R\}$.

Fonction de transition $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 0, 1\}$
Calcul : entre : $x_1, x_2, \ldots, x_m$, curseur en position 1, état q_0. La suite de transition s'arrête si on a q_A ou q_R.
Temps de calcul : $t_m(x)$ = nombre de transitions jusqu'à q_A ou q_R (on peut avoir ∞). Le langage reconnu par M est : $L_M = \{x \in \Sigma^*, M \text{ accepte } x\}$.
Tableau :

	b	0	1
q_0	b,R,0	0,q_0,+1	1,q_1,+1
q_1	b,A,0	0,q_0,+1	1,q_1,+1

En résumé :
- Algorithme = machine de Turing
- Problème = Langage ($\subset \Sigma^*$)

On a t_M qui est la complexité de la machine M (en temps)

Définition : machine de Turing universelle (ordinateur)
La machine U prend comme entrée $< M >$ qui est la description de la machine M et x et U retourne ce que retourne M sur l'entrée x.

Théorème :

Il existe des machines de Turing universelles.
On a $t_U(< M >, x) \leqslant c|x|t_M(x)$ avec $|x|$ longueur de x.

Chapitre 2

Complexité en temps

2.1 Problèmes faciles ?

Définition :
La classe P est l'ensemble des langages polynomiaux.

Définition :
L est un langage polynomial si $\exists$ M et un polynôme $P(X) \in \mathbb{Z}[X]$ tels que $\forall x \in \Sigma^*,\ t_M(x) \leqslant P(|x|)$ et M reconnait L.

2.2 Problèmes difficiles ?

Exemples :
- Circuit hamiltonien (CH) :
  I : graphe
  Q : existe-t-il un circuit hamiltonien dans ce graphe ?
- Satisfaisabilité (SAT) :
  I : formule booléenne : $f = (x_1 \cup x_2 \cup \bar{x_4}) \cap (x_2 \cup \bar{x_5})$
  Q : $\exists?(a_1, \ldots, a_n) \in \{0,1\}^n,\ f(a_1, \ldots, a_n) = 1$ (vrai)
- FBQ :
  I : $\forall x_1, \exists x_2, \forall x_3, x_4,\ f(x_1, \ldots, x_n) = 1$
  Q : Vrai ou faux ?
- Clique
  I : graphe G, $k \in \mathbb{N}$
  Q : existe-t-il une clique (sous graphe complet) de taille k ?
- 10ème problème de Hilbert :
  I : un polynôme $P(X_1, \ldots, X_n) \in \mathbb{Z}[X_1 \ldots X_n]$
  Q : $\exists(a_1, \ldots, a_n) \in \mathbb{Z}^n,\ P(a_1, \ldots, a_n) = 0$?

- Voyageur de commerce (VC) :
  I : graphe G, $p_1, \dots p_m \in \mathbb{N}, k \in \mathbb{N}$ avec p_i poids de a_i
  Q : Existe-t-il un parcours de tous les sommets du graphe où la somme des longueurs des arêtes parcourues est $< k$?
- k-coloration :
  I : graphe, $k \in \mathbb{N}$
  Q : Peut-on colorier les sommets avec k-couleurs ?

Définition :
Un langage L est dit certifiable en temps polynomial si $\exists M$ et $P(X)$ tel que pour tout $x \in L$, $\exists y \in \Sigma^*$ tel que sur l'entrée (x, y), M aboutit à l'état q_A en temps :

$$t_M(x, y) \leqslant P(|x|)$$

D'où, la classe NP est l'ensemble des langages certifiables en temps polynomial.
Conjecture : $P \neq NP$.

Réduction polynomiale
$\varphi : \Sigma^* \longrightarrow \Sigma^*$ est une réduction polynomiale de L vers L' si :
- φ est calculable en temps polynomial
- $l \in L \Longleftrightarrow \varphi(l) \in L'$

Définition :
$L \propto L'$ si $\exists \varphi$ réduction de L vers L'.
Exemple : CH $\propto$ VC, $\varphi : G \rightarrow (G, p_1 = \ldots = p_n = 1, k = n)$

Définition : Complétude
Λ est NP-complet si :
- $\Lambda \in NP$
- $\forall L \in NP,\ L \propto \Lambda$

Théorème :
Il existe un problème NP-complet.

Exemple d'un tel problème : Arrêt en temps limité
I : $< M >,\ x \in \Sigma^*,\ 1^t = 11 \dots 1$ (t fois)
Q : Est qu'il existe $y \in \Sigma^*$ tel que M s'arrête sur q_A sur l'entrée (x, y) en temps $\leqslant t$?

Théorème :
L'arrêt en temps limité est NP-complet.

Preuve :

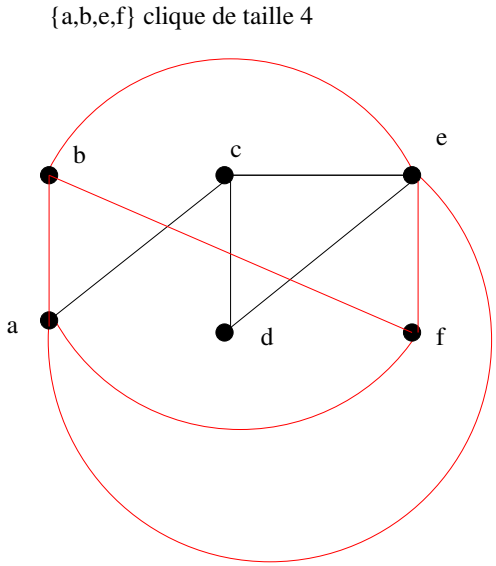
- Problème dans NP ?
  Le certificat est y. Soit U une machine de Turing universelle, U exécute le déroulement de M sur l'entrée (x, y) et 1^t.
- $\forall L \in NP,\ L \propto \Lambda$?
  L vient avec M_L, P un polynôme, tels que $x \in L \iff \exists L,\ M_L$ accepte (x, y) en temps $\leqslant P(|x|)$

$$x \longmapsto < M_l >,\ x,\ 1^{P(|x|)}$$

  $x \in L \iff \exists y$ tel que M_L s'arrête sur (x, y) en temps $\leqslant P(|x|)$
  $x \in L \iff f(x) \in \Lambda$

Théorème de Cook
SAT est NP-complet.

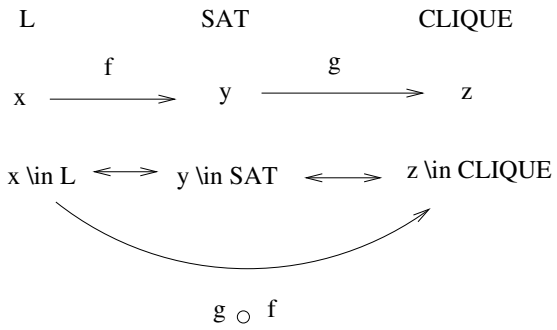
Exemple de réduction polynomiale
Le problème CLIQUE :
I : G graphe, $k \in \mathbb{N}$
Q : existe-il une clique de G (sous graphe complet) de taille k ?



Proposition :
CLIQUE est NP-complet.

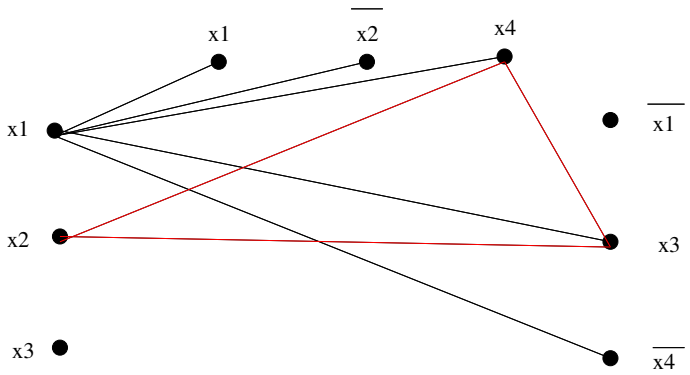
Preuve :

$$SAT \propto CLIQUE$$



Proposition :
Si Λ est NP-complet et si $\Lambda \propto L$ et $L \in NP$ alors L est NP-complet. (car si $L \propto L'$ et $L' \propto L''$ alors $L \propto L''$)

Exemple : $f = (x_1 \cup x_2 \cup x_3) \cap (x_1 \cup \bar{x_2} \cup x_4) \cap (\bar{x_1} \cup x_3 \cup \bar{x_4})$ donne un graphe G avec #sommets = longueur de f = #litéraux.



Il faut des arêtes partout sauf :
– entre sommets issus d'une même clause
– entre 2 sommets issus de x_i et $\bar{x_i}$

$k = \#$clauses (ici $k = 3$)
f satisfaisable $\Leftrightarrow$ G admet une k-clique.
$\Rightarrow$: Si f est satisfaisable, il existe dans chaque clause, un litéral de valeur 1. Cela donne une clique dans G.
$\Leftarrow$: Si on a une k-clique, il est possible de mettre un terme dans chaque clause à 1.

2.3 Classe NP et fonctions à sens unique

Définition : Fonction à sens unique
$f : \Sigma^* \longrightarrow \Sigma^*$
- $Dom(f)$ est infini
- f est calculable en temps polynomial
- $\not\exists M$ machine, polynome P tel que $\forall y \in Im(f)$, M calcule x tel que $f(x) = y$ en temps $\leqslant P(|y|)$.
- $\exists P$ polynome, $\forall y \in Im(f)$, il existe x, $|x| \leqslant P(|y|)$ tel que $f(x) = y$ car sinon on peut prendre $f(x) = \log|\log|x||$

Théorème :
Il existe f fonction à sens unique $\Longleftrightarrow P \neq NP$

$\Rightarrow$: On construit le problème suivant :
I : $x, y \in \Sigma^*$, P polynome
Q : $\exists z \in \Sigma^*$, $f(xz) = y$? et $|z| \leqslant P(|y|)$
- problème dans NP ? Oui, z est le certificat
- Si f à sens unique, le problème $\notin P$. Sinon, on calcule rapidement un antécédent de y. On prend y et on pose Q avec $x = 0$ si non $x = 1$ si oui, $x = 10$ ou $x = 11$? Et ainsi de suite...
  $\#$appels au problème de décision est de $2p(|y|)$.

$\Leftarrow$ Soit $\varphi : \Sigma^* \to \Sigma^* \times \Sigma^*$ une bijection, calculable en temps polynomial.

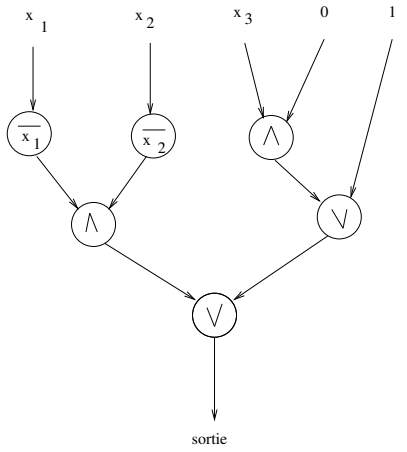
$$\{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$$

Par exemple : $E \subset \{0,1\}^* \to \mathbb{N}$ écriture en base 2 $\leftarrow n$
$L \in NP$ avec $(M_L, p_L())$, $z \in \Sigma^* : z \longmapsto \varphi(z) = (\varphi_1(z), \varphi_2(z)) = (x, y)$
$f(z) = \begin{cases} 1) x \in L, y \text{ certificat de x alors } f(z) = x \\ 2) \text{ sinon f n'est pas définie en z} \end{cases}$

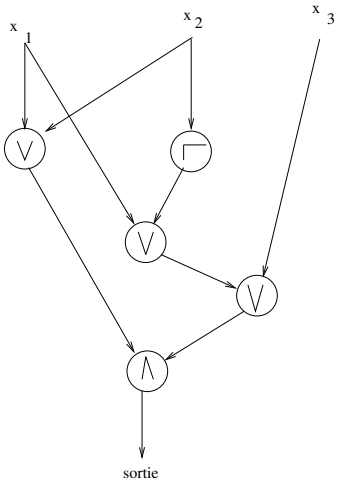
1. f calculable
2. f à sens unique car si $\exists M$ tel que $x \mapsto z$ tel que $f(z) = x$ alors $x : M \mapsto z \mapsto \varphi(z) = (x, y)$ avec y certificat de $x \in L$. A partir de x,

on trouverait le certificat, M fabriquerait donc des certificats ce qui est impossible.

Variantes de SAT $SAT \propto 3-SAT$ (et $3-SAT \propto SAT$)
Qu'en est -il de $2-SAT$? $\longrightarrow 2-SAT \in P$
Autre variante : SATC (Satisfaisabilité en circuit)



Proposition :
$SAT \propto SATC$
$(x_1 \cup x_2) \cap (x_1 \cup \bar{x_2} \cup x_3) \longrightarrow C$



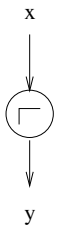
Proposition :
$SATC \propto SAT$

Réduction :

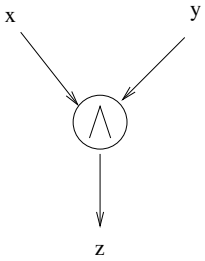
Il faut ajouter des variables auxiliaires, autant que d'arêtes (arcs) issues des portes du circuit.

Porte Non

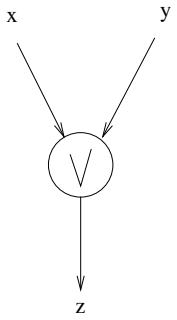
$y = \bar{x_1} \Leftrightarrow (x_1 \cup y) \cap (\bar{x_1} \cup \bar{y}) = 1$



Porte Et $z = x \cap y \Leftrightarrow (\bar{x} \cup \bar{y} \cup z) \cap (x \cup \bar{y} \cup \bar{z}) \cap (\bar{x} \cup y \cup \bar{z}) \cap (x \cup y \cup \bar{z}) = 1$



Porte Ou $z = x \cup y \Leftrightarrow (x \cup y \cup \bar{z}) \cap (x \cup \bar{y} \cup z) \cap (\bar{x} \cup y \cup z) \cap (\bar{x} \cup \bar{y} \cup z) = 1$



Théorème de Cook :

SAT est NP-complet.

Preuve :

Il suffit de montrer que SATC est NP-complet.

Stratégie :

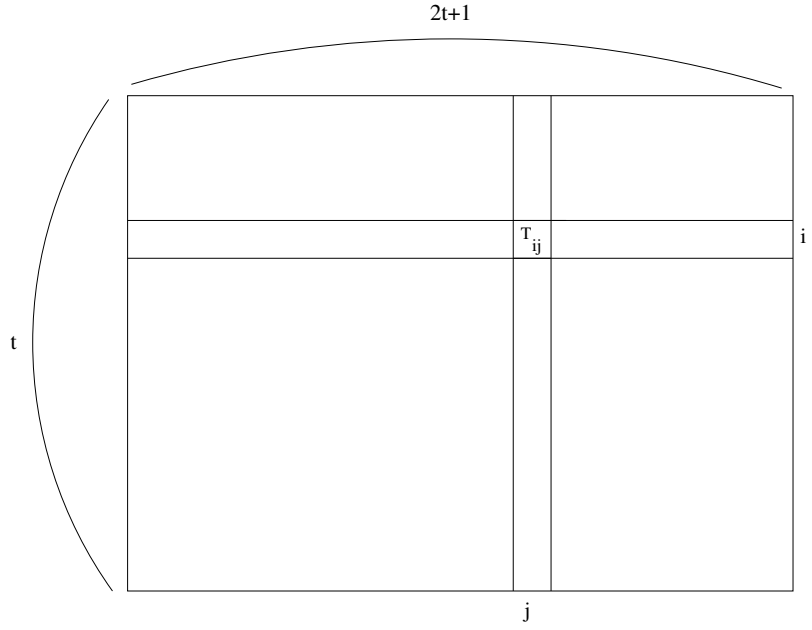
Le circuit a pour entrées $x_1 \dots x_n y_1 \dots y_m$ et le circuit renvoie 1 si $y_1 \dots y_m$ est un certificat de x. Si $x \in L$, $\exists y$ tel que $|y| \leqslant p_L(|x|)$, M accepte (x, y) en temps $\leqslant p_L(|x|)$.

Tableau de calcul :

Paramètre : $t = p_L(|x|)$

$-t$	$\dots$	$\dots$	$\dots$	-1	0	1	2	$\dots$	n	$\dots$	t
$\dots$	$\dots$	$\dots$	$\dots$	y_1	b	x_1	x_2	$\dots$	x_n	$\dots$	$\dots$

On construit le tableau $(T_{ij})_{i=1\dots t, j=-t\dots t}$ de taille $t \times 2t+1$.

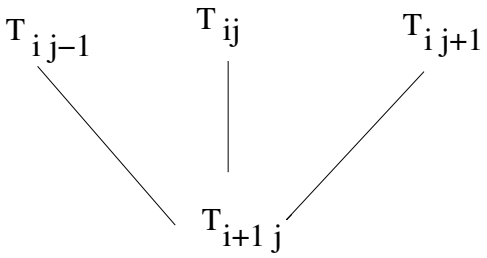


On a $T_{ij} = (a_{ij}, h_{ij}, q_{ij})$ avec :

- a_{ij} = contenu de la case j au temps i
- $h_{ij} = \begin{cases} 1 \text{ si le curseur est en j au temps i} \\ 0 \text{ sinon} \end{cases}$
- $q_{ij} = \begin{cases} \text{état dans lequel est M au temps i si } h_{ij} = 1 \\ 0 \text{ sinon} \end{cases}$

Point cl :

$T_{i+1\ j}$ est fonction déterministe de $T_{i\ j-1}, T_{ij}, T_{i\ j+1}$

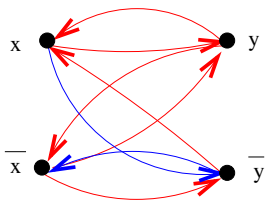


Circuit :
entrée : $x_1, \ldots, x_n$ (constantes), $y_1, \ldots, y_m \in \Sigma \cup \{b\}$.
Toute fonction $F \to F$ est représentable par un circuit fini. A la fin, la sortie est 1 s'il existe q_A dans la dernière ligne.

2-SAT :

Proposition : 2-SAT est polynomial.
I : $f = C_1 \cap C_2 \cap \ldots \cap C_k$ avec $C_i = x \cup y$
Q : f satisfaisable ?

Exemple : $(x \cup y) \cap (\bar{x} \cup y) \cap (x \cup \bar{y}) \cap (\bar{x} \cup \bar{y})$
Idée : graphe orienté avec pour sommets les litéraux. On a : $a \cup b$ devient $\bar{a} \to b$ et $\bar{b} \to a$



Proposition :
f est satisfaisable si et seulement si $\nexists x$ tel que $x \to \bar{x}$ et $\bar{x} \to x$.

2.4 Pspace, RP, BPP

2.4.1 Pspace

Au dessus de NP, on a Pspace, espace polynomial.

Définition : Pspace
Pspace est la classe des langages L tels que $\exists M$ et un polynome P tels que sur l'entrée $(x_1, \ldots, x_n)$, $\forall x \in \Sigma^*$, M décide si $x \in L$ en temps $\leqslant 2^{q(|x|)}$ avec q un polynome. $|\Gamma|^{2m+1}|\text{états}|(2m+1)$ majore le nombre d'itérations (sinon la machine boucle).

Proposition : $NP \subset Pspace$

Preuve :
Soit $L \in NP$. L'algorithme est de passer en revue tous les certificats y possibles.

Conjecture :
$P \subsetneq NP \subsetneq Pspace$

Définition :
Un langage L est dit Pspace-complet si :
- $L \in$ Pspace
- $\forall L' \in$ Pspace, $L' \propto L$

Théorème :
FBQ est Pspace-complet.
I : $\forall x_1,\ \exists x_2, x_3,\ \forall x_4,\ f(x_1, x_2, \ldots, x_n) = 1$
Q : vrai ou faux ?

2.4.2 Algorithmes probabilistes polynomiaux

Exemple : Test de (non)-primalité
Solovay - Strassen :
On a en entrée n, avec a aléatoire, a-t-on $a^{\frac{n-1}{2}} = \left(\frac{a}{n}\right) \mod n$? Si non, n n'est pas premier. Si n n'est pas premier, on a :

$$\frac{\#\{a \in (\mathbb{Z}/n\mathbb{Z})^*,\ a^{\frac{n-1}{2}} \neq \left(\frac{a}{n}\right) \mod n\}}{\varphi(n)} \geqslant \frac{1}{2}$$

Définition : RP
Soit L un langage, $\exists M$ et p un polynome tels que pour toute entrée x, il existe un ensemble de certificats Y, de taille $\leqslant p(|x|)$ tel que pour tout (x, y) donnés à M :
- si $x \notin L$, M rejette (x, y) avec une probabilité 1 (pour tout $y \in Y$)

– si $x \in L$, M accepte (x, y) avec une probabilité $\geqslant \frac{1}{2}$ (pour une moitié des y).

M calcule en temps polynomial.

Proposition :
$RP \subset NP$ et $P \subset RP$

Définition : BPP
Soit un langage L, $\exists M$ et un polynome p tel que pour toute entrée x, il existe un ensemble de certificats Y, de taille $\leqslant p(|x|)$, tel que pour tout (x, y) :

– si $x \notin L$, M rejette (x, y) avec une probabilité $> \frac{2}{3}$

– si $x \in L$, M accepte (x, y) avec une probabilité $> \frac{2}{3}$

On a : $RP \subset BPP$ et $RP \subset NP$