NP : 2 définitions équivalentes

Par problème, on entend problème de décision, à réponse binaire oui/non (par exemple une formule en logique propositionnelle au format CNF est-elle satisfiable ?).

1. NP = les problèmes vérifiables en temps polynomial

Un problème *P* est dans NP si il existe un algorithme de vérification déterministe *A(i,c)*, où *i* est une instance et *c* un certificat (par exemple une solution candidate) tel que :

- *A(i,c) = 1* ssi *i* est une instance positive de *P*, validée par le certificat *c* (si *A(i,c) = 0* alors on ne peut rien conclure sur l'instance *i* : positive ou négative, on ne sait pas, par exemple *i* peut être une instance positive et *c* un certificat inadéquat) ;
- *A(i,c)* est calculable en temps polynomial en |*i*|.

2. NP = les problèmes décidables en temps polynomial par un algorithme non-déterministe

Un problème *P* est dans NP si il existe un algorithme non-déterministe *N(i)* polynomial en la taille de l'instance *i* tel que :

- si *N(i) = 1* alors *i* est une instance positive de *P* ;
- si *N(i) = 0* alors *i* est une instance négative de *P*.

Ces deux définitions sont *équivalentes* :

---

1 => 2

A partir d'un algorithme de vérification *A(i,c)*, on construit un algorithme *N(i)* comme suit :

- choisir de façon non-déterministe une solution candidate *c ;*
- retourner la valeur *A(i,c)*.

Par définition des algorithmes non-déterministes, si un des choix retourne 1, *N(i)* s'évalue à 1 et si tous les choix retourne 0, *N(i)* s'évalue à 0.
On constate que l'algorithme *N(i)* est non déterministe, polynomial en |*i*| et qu'il décide P.

---

2 => 1

A partir de *N(i)* non déterministe, polynomial en |*i*| et qui décide P, on construit un algorithme déterministe *A(i,c)* qui vérifie P en temps polynomial comme suit. Comme on a le choix pour le certificat, on décide que le certificat sera la séquence des choix effectués par l'algorithme non déterministe *N(i)* s'il retourne 1. Si *N(i)* retourne 0, alors le certificat est une séquence arbitraire de taille adéquate.
Voici l'algorithme *A(i,c)* :

- simuler *N(i)* en effectuant les choix indiqués par *c* ;
- si la simulation déterministe précédente retourne 1 alors retourner 1 sinon 0.

On constate que *A(i,c)* est un algorithme de vérification déterministe pour P de complexité polynomiale en |*i*|.