Serge Burckel

Simplicité et Complexité.

Résumé. Nous présentons des éléments vers une réponse simple à une question difficile de complexité algorithmique concernant les classes P et NP.

La première simplification consiste à considérer ces classes dans un contexte “pragmatique”.

On commence par fixer un entier K suffisamment grand avec par exemple $K = 10$ ou $K = 10^{1000}$.

Définition. *Pour tout entier K fixé assez grand, $P^K = DTIME(n^K)$ et $NP^K = NTIME(n^K)$.*

Proposition. *Pour tout entier K fixé assez grand, $P^K \neq NP^K$.*

Preuve. Fixons un entier K assez grand. Etant donnée une machine de Turing M (déterministe ou non), on note $< M >$ un codage de M. Soit l'ensemble L des codes $< M >$ pour lesquels il existe un calcul rejetant de M à partir du mot $< M >$ de longueur n en au plus n^K étapes.

Nous allons montrer que le langage L est dans NP^K mais n'est pas dans P^K.

Supposons que $L \in P^K$. Il existe alors une machine de Turing déterministe M qui décide L en au plus n^K étapes sur des entrées de longueur n. Soit alors l'entrée $W =< M >$ de longueur N.

• Si $W \in L$ alors il existe un calcul rejetant de M à partir de W en au plus N^K étapes. Puisque M est déterministe, ce calcul est unique et il n'existe pas de calcul acceptant. Donc $W \notin L$.

• Si $W \notin L$ alors il n'existe pas de calcul rejetant de M à partir de W en au plus N^K étapes. Puisque ce nombre d'étapes est supposé suffisant pour cette machine particulière M, elle est en mesure de conclure et d'accepter W. Mais alors $W \in L$.

Dans les deux cas nous obtenons une contradiction.

Montrons que $L \in NP^K$. Considérons pour cela une machine de Turing non déterministe S qui étant donnée une entrée w réalise le programme non déterministe suivant :

Rejeter le mot w ou alors si w est un code $< M >$ de longueur n d'une machine de Turing M, deviner un calcul rejetant en au plus n^K étapes de M sur w et si une vérification par simulation montre que ce calcul est bien rejetant alors accepter w.

La machine S décide bien le langage L. Par ailleurs, elle ne tombe pas dans les contradictions précédentes pour $W =< S >$. On aura $W \in L$. En effet, nous avons défini S de telle sorte qu'elle puisse immédiatement rejeter tous les mots. Elle peut donc rejeter en particulier le mot W. Elle peut ainsi aisément le deviner et le vérifier et donc accepter W dans l'autre choix de calcul. Cette existence d'un calcul acceptant est la condition nécessaire et suffisante de décision pour une machine de Turing non déterministe comme S.

$\blacksquare$