Chapitre 1

Chapitre 1 : Introduction à la décidabilité

1.1 Introduction

Il y a maintenant des dizaines de siècles que les mathématiciens décrivent et utilisent des méthodes de calcul permettant de résoudre leurs problèmes. Mais jusqu'à une date relativement récente (1934) ils ne savaient pas de façon précise ce qu'est une méthode de calcul.

La faculté étonnante des mathématiques à transformer en objets ce qui constitue leurs méthodes et leurs techniques, les rapproche de la philosophie. Cette faculté d'autoréflexivité permet par exemple de prendre les théories mathématiques pour objets d'études (et donc de théorèmes) comme les nombres entiers et les équations aux dérivées partielles sont objets d'études et de théorèmes.

Et effectivement, à propos du problème des méthodes de calcul, l'autoréflexivité des mathématiques a fonctionné, donnant naissance à une série de concepts et de résultats qui sont parmi les plus profonds et les plus féconds du vingtième siècle.

Exemples :
- Algorithmes,
- Fonctions calculables,
- Problèmes décidables et indécidables.

Aujourd'hui les mathématiciens possèdent une définition précise de ce qu'on doit appeler méthode de calcul, ils savent exactement quels sont les problèmes qu'elles peuvent traiter, et mieux encore, ils savent que certains problèmes ne sont pas "traitables" En effet, et c'est là l'un des aspects les plus extraordinaires de ces travaux, ils permettent d'établir des résultats négatifs, c'est-à-dire de la forme : pour tel problème non seulement aucune méthode n'est actuellement connue, mais aucune ne le sera jamais.

Ces problèmes pour lesquels on démontre qu'il n'existe aucune méthode de calcul adaptée s'appellent des problèmes indécidables.

Les premiers résultats d'indécidabilité des années 1930 semblaient artificiels, très vite on s'est aperçu que des problèmes assez simples entraient dans la classe des problèmes indécidables. En particulier en informatique, de nombreuses questions naturelles qui se posent aux programmeurs se révèlent après étude correspondre à des problèmes indécidables. Lorsqu'on démontre qu'un problème P est indécidable, on en déduit : - qu'on doit renoncer à le résoudre tel quel, et donc que, - qu'il faut trouver une version simplifiée de P, puis, soit réussir à la résoudre, soit à nouveau établir qu'elle est encore indécidable et donc la simplifier encore, etc (les fonctions récursives).

1.2 Petit historique

La notion informelle d'algorithme est extrêmement ancienne et les procédés que nous avons appris à l'école primaire pour additionner deux nombres écrits en base dix ou pour les multiplier sont des algorithmes.

En fait, au début du siècle les mathématiciens ne soupçonnaient pas qu'on pourrait arriver à préciser vraiment cette notion, ni encore moins, qu'on pourrait démontrer pour certains problèmes qu'aucun algorithme n'existe.

Le travail d'identification et de formulation de la notion d'algorithme fut effectué en plusieurs étapes entre 1931 et 1936 par les mathématiciens Church, Kleene, Turing et Gödel.

Ils introduisirent plusieurs classes différentes de fonctions dont ils montrèrent ensuite qu'elles coïncidaient, et qu'ils reconnurent alors comme la classe des fonctions calculables. Une fonction est calculable s'il existe une façon finie de la décrire qui permette effectivement d'en calculer toutes les valeurs. La définition précise de la notion de fonction calculable fixe en même temps celle d'algorithme, et on peut donc dire qu'en 1936 la formulation exacte de la notion d'algorithme était acquise. De toutes les définitions finalement équivalentes de la notion d'algorithme formulées dans les années 1930 et depuis, celle donnée par Turing en 1936 est la plus pratique pour les théoriciens, et on l'utilise encore aujourd'hui. C'est sur elle que nous allons nous appuyer plus loin pour définir la notion d'algorithme.

Chaque année des dizaines de résultats de décidabilité ou d'indécidabilité sont établis dans de nombreux domaines des mathématiques et on peut être certain, tant les questions ouvertes restent nombreuses, que ce mouvement n'est pas prêt de cesser. Ici nous nous intéresserons uniquement au problème de la possibilité d'un algorithme ou de son impossibilité, et non pas au problème de l'efficacité qui lui aussi est à l'origine d'un domaine de travail et de réflexion des plus vivants aujourd'hui.

1.3 Quelques problèmes décidables, semi-décidables et indécidables

Problème A.
Soient m et n deux entiers donnés > 1. m est-il un multiple de n ? On sait qu'il est vrai que : 12 est un multiple de 2, et qu'il est faux que : 16 est un multiple de 5. On sait même bien plus que cela, on sait comment s'y prendre pour déterminer pour tout n et tout m quand est vrai que "m est un multiple de n" et quand est faux que : "m est un multiple de n".

Il suffit en effet de :
- Faire la division de m par n,
- Regarder le reste obtenu, r,
- Si r = 0 alors il est VRAI que "m est un multiple de n",
- Sinon il est FAUX que "n est un multiple de m".

Ce procédé général et systématique de calcul, constitue un algorithme (informel). C'est un procédé absolument sûr, qui fonctionne en un temps fini pour tous les jeux de données possibles et conduit toujours à la bonne réponse : il montre que le problème A est décidable.

Remarquons bien que pour être précis quand on parle de problème décidable ou indécidable il faut indiquer ce que sont les paramètres du problème. Dans notre exemple n et m sont deux entiers > 1. En fait on ne cherche pas à résoudre un problème unique, on cherche à résoudre une classe infinie de problèmes, ici tous les problèmes :

2 est-il un multiple de 2 ?
3 est-il un multiple de 2 ?
....
10 est-il un multiple de 3 ?
....

Problème B.
Problème des nombres premiers. Soit n un entier donné > 1
n est-il un nombre premier ?

Problème C.
Il existe aussi des problèmes qui sont semi décidables.
Exemple : Soit le programme P suivant :
"3x + 1" Est-ce que le programme ci-dessous s'arrête sur un x donné ?

while x > 1 do
{ if (x mod 2 = 0) then x := x/2 ;
else x := 3x + 1 ; }

Réponse. Pour ce problème, on peut utiliser la procédure suivante : 'a partir de x donné exécuter ce programme, et s'il s'arrête retourner "OUI". Par contre, si pour un certain x le programme "3x + 1" boucle, cette procédure ne pourra jamais donner la réponse "NON". Une telle procédure s'appelle un semi-algorithme on peut dire aussi que c'est un programme semi décidable.

Il est inconnu s'il existe un algorithme de décision pour ce problème, qui pour chaque x donné renvoie une réponse correcte "OUI" ou "NON".

Problème D.
Il y a aussi des problèmes indécidables.

Exemple :
Soit l'algorithme A/P(x) { tant que AP(x) (si x est pair) alors x := x*2 }.
Il est évident que si on lui passe comme paramètre un x pair, cet algorithme ne se terminera jamais donc il est indécidable.