

Histoire de l'algorithmique

La notion d'algorithme remonte à l'antiquité. Cela s'est précisé dans le domaine des mathématiques par l'emploi de variables. L'algorithme au sens informatique apparaît avec l'invention des premières machines dotés d'automatismes.

Evolution

- [Origine du mot](#)
- [L'algèbre à l'origine des variables](#)
- [Les anciens](#)
- [Symboles, règles et paradoxes](#)
- [Première formalisation](#)
- [Post, l'espace de symboles](#)
- [Turing et le calculateur humain](#)
- [La méthode effective de Rosser](#)
- [La théorie algorithmique de Kleene](#)

Origine du mot

Le mot *algorithme* vient du nom du mathématicien perse du 9^{ème} siècle (AJC) Abu Abdullah Muhammad ibn Musa al-Khwarizmi. Le mot *algorisme* se référait à l'origine uniquement aux règles d'arithmétique utilisant les chiffres indo-arabes numéraux mais cela a évolué par la traduction en latin européen du nom Al-Khwarizmi's en *algorithme* au 18^{ème} siècle. L'utilisation du mot a évolué pour inclure toutes les procédures définies pour résoudre un problème ou accomplir une tâche.

L'algèbre à l'origine des variables

Le travail des anciens géomètres grecs, du mathématicien perse Al-Khwarizmi - souvent considéré comme le "père de l'algèbre", des mathématiciens chinois et européens de l'est ont atteint un point culminant avec Leibniz dans la notion de "calcul ratiocinateur", une algèbre logique.

Les anciens

Euclide a formalisé un algorithme auquel on a donné son nom. Cet algorithme qui sert à calculer le plus grand diviseur commun est le suivant:

- diviser a par b, on obtient le reste r
- remplacer a par b
- remplacer b par a
- continuer tant que c'est possible, sinon on obtient le pgcd.

L'algorithme d'Archimède donne une approximation du nombre Pi.

[Eratosthènes](#) a défini un algorithme pour retrouver les nombres premiers.

Averroès (1126-1198) utilise un procédé algorithmique pour ses calculs.

Au 12^{ème} siècle Adelard de Bath introduit le mot algorismus dérivé de Al Kwarizmi.

Symboles, règles, et paradoxes

Entre les années 1800 et jusqu'au milieu des années 1900:

- **George Boole** (1847) invente l'algèbre binaire, la base des ordinateurs. En fait il a unifié la logique et les calculs dans un symbolisme commun.

- **Gottlob Frege** (1879) et le langage de formules, qui est une *lingua characterica*, un langage écrit dans des symboles spéciaux, "pour la pensée pure", sans aucun embellissement rhétorique... construit avec des symboles spécifiques

manipulés selon des règles bien définies.

- **Giuseppe Peano** (1888) écrit *Les principes de l'arithmétique, présentés par une nouvelle méthode*, qui est la première tentative d'axiomatisation des mathématiques dans un langage symbolique.

- **Alfred North Whitehead** et **Bertrand Russell** dans leur *Principia Mathematica* (1910-1913) ont à leur tour simplifié et amélioré le travail de Frege.

- **Kurt Gödel** (1931) cite le paradoxe du menteur qui réduit les règles de récursion entièrement en nombres.

Première formalisation

Le concept a été formalisé en 1936 avec les machines d'Alan Turing et le calcul lambda d'Alonzo Church, ce qui a alors créé les fondations de l'informatique.

Post, l'espace de symboles

Emil Post (1936) a décrit les actions d'un "ordinateur" comme ceci:

"...deux concepts sont impliqués: celui d'un *espace de symboles* dans lequel le travail qui produit le problème à résoudre doit être accompli, et un *ensemble de directions* fixé de manière irrévocable."

Son espace de symboles pourrait être

"une séquence d'espaces ou de boîtes infinies à deux états... Le résolveur du problème ou le travailleur se déplace et travaille dans cet espace symbolique, il peut être et agir dans une boîte et une seule à la fois... une boîte peut admettre deux conditions possibles, être vide ou non marquée, ou avoir une seule marque, disons une barre verticale."

"Une boîte est mise à part et considérée comme le point de départ. (...) un problème donné est fourni sous forme symbolique par un nombre fini de boîtes - les entrées - marquées d'un trait. De même, la réponse - la sortie - sera donnée sous forme symbolique par une configuration de boîtes marquées..."

"Un ensemble de directions qui s'appliquent à un problème général définit un processus déterministe quand il s'applique à chaque problème spécifique. Ce processus se termine seulement quand il arrive à la direction de type *stop*."

Turing et le calculateur humain

Le travail d'**Alan Turing** (1936-1937) a précédé celui de Stibitz (1937); on ignore si Stibitz connaissait le travail de Turing. Le biographe de Turing pensait que l'utilisation par Turing d'un modèle inspiré par la machine à écrire avait une source dans son enfance: il rêvait d'inventer des machines à écrire quand il était petit garçon; Mrs. Turing avait une machine à écrire et il aurait pu commencer par se demander s'il y avait un moyen de rendre la machine à écrire capable de fonctionner seule de façon mécanique. Etant donné à l'époque la prévalence du code Morse et le télégraphe, des machines à bandes d'imprimeur, du Télétipe, on peut conjoncturer que tout cela l'a influencé.

Turing, dont le modèle de calcul est maintenant appelé la *machine de Turing*, a commencé, comme le fit Post, par une analyse d'un ordinateur humain réduit à des opérations élémentaires, les "états de pensée". Mais il va plus loin en créant un modèle de machine pouvant calculer avec des nombres:

"Calculer se fait normalement en écrivant certains symboles sur le papier. Nous pouvons imaginer le papier divisé en carrés comme le livre d'arithmétique des enfants... Je considère ensuite que le calcul est transporté du papier à une dimension sur une bande divisée en carrés. Je vais aussi supposer que le nombre de symboles qui peut être imprimé est infini..."

"Le comportement de l'ordinateur à tout moment est déterminé par les symboles qui lui apparaissent, et son "état de pensée" à ce moment. Nous pouvons supposer qu'il y a une limite B au nombre de symboles ou carrés que l'ordinateur peut voir à un moment donné. S'il désire en voir plus, il doit faire des observations successives. Nous allons aussi supposer que le nombre d'états de pensée à prendre en compte est fini..."

"Imaginons que les opérations accomplies par l'ordinateur sont divisées en opérations simples, qui sont si élémentaires qu'il n'est pas aisé d'imaginer qu'elles puissent être encore plus divisées".

Les réductions de Turing aboutissent à l'ensemble suivant:

Les opérations simples doivent donc inclure:

- 1) Le changement de symboles sur un des carrés observé.
- 2) Le changement d'un des carrés observé vers un autre carré, en passant par N carrés parmi les carrés précédemment observés."

"Il se peut que quelques uns de ces changements amènent un changement de l'état de pensée. L'opération simple la plus générale doit donc être prise parmi une des suivantes:

- 1) Un changement possible (a) d'un symbole en même temps qu'un changement possible d'état de pensée.
- 2) Un changement possible (b) de carrés observés, allant de pair avec un changement possible d'état de pensée.

Nous pouvons maintenant construire une machine pour faire le travail de cet ordinateur."

La méthode effective de Rosser

J. Barkley Rosser (1939) a audacieusement défini une méthode mathématique effective de la façon suivant:

"La méthode effective est utilisée ici dans le sens particulier d'une méthode dont chaque étape est précisément déterminée et qui est certaine de produire une réponse en un nombre fini d'étapes. Dans ce sens là, trois définitions précises ont été données à ce jour. (1) La plus simple d'entre elles (dûe à Post et Turing) dit en substance qu'**une méthode effective pour résoudre certains ensembles de problèmes existe si on peut construire une machine qui va résoudre tout problème de l'ensemble sans intervention humaine, autre que poser la question et (plus tard) lire la réponse.** Les trois définitions sont équivalentes, aussi peu importe laquelle on utilise. D'ailleurs, le fait que toutes trois sont équivalentes est un argument fort pour la validité de chacune."

(1) Rosser fait référence au travail de Church et Kleene et leur définition d' i-définabilité; Herbrand et Gödel dans leur utilisation de la récursion; Post et Turing avec leur modèle mécaniste de calcul.

La théorie algorithmique de Kleene

Stephen C. Kleene (1943) a défini une thèse maintenant fameuse et connue comme la "Thèse de **Church-Turing Thesis**". Dans ce contexte:

" **Théories algorithmiques**... En mettant au point une théorie algorithmique complète, ce que nous faisons, c'est décrire une procédure, applicable à chaque ensemble de valeurs des variables indépendantes, laquelle procédure se termine nécessairement et d'une façon telle que ce qui en résulte constitue une réponse définitive, "oui" ou "non," à la question, "la valeur du prédicat est-elle vraie?"