

Abderrahmane Fadil

Algorithmique et développement Python

Cours et exemples d'applications



Chapitre I. Représentation des données et algèbre de Boole

1 Généralités sur les algorithmes

1.1 Introduction

Depuis la machine de Turing et la procédure de déchiffrement des messages codés, le mot *algorithme* est associé aux traitements automatiques, aux stratégies d'optimisation et à l'informatique moderne. Actuellement on l'associe aux méthodes d'analyse et de traitement des données, à l'analyse prédictive, au datamining et encore à l'apprentissage, au *deep learning* et à l'Intelligence Artificielle.

Aujourd'hui et demain, chaque fois qu'il s'agit de mettre en place un traitement, automatique, autonome, qui évolue intelligemment dans des contextes connus ou nouveaux, on fera appel aux algorithmes et à l'Intelligence Artificielle.

L'algorithme est un terme d'origine arabe qui fait référence au mathématicien Al Khawarizmi (780 - 850) auteur d'un ouvrage décrivant des méthodes de calculs algébriques. A l'origine du mot **algorithme** en français (*algorithm* en anglais, *algorithmus* en allemand, *algorithmo* en espagnol, *aloritmul* en roumain, ...) on trouve le mot *Algorismus* qui est la forme latine du nom Khwarizmi.

1.2 Définition 1

Un algorithme est une méthode de résolution de problèmes énoncée sous la forme d'une série d'instructions ou de commandes, c'est-à-dire des opérations à effectuer.

1.3 Exemples

Voici une liste, évidemment non exhaustive, de situations ou problèmes à résoudre et dont la résolution consiste en amont à procéder par méthode et à effectuer une séquence d'opérations dans un ordre cohérent menant ainsi à la ou à une solution du problème posé :

- préparer une recette de cuisine ;
- calculer puis préparer une ration, équilibrée, nutritionnelle équine ;
- récompenser la fidélité d'un client dans le cadre d'une gestion de la relation client GRC ;

- stocker et diffuser les données d'un réseau de capteurs météorologiques implantés sur une parcelle ou un territoire ;
- déterminer la meilleure implantation géomarketing d'une enseigne marchande ;
- déterminer la meilleure implantation géoéconomique et environnementale d'un silo à grain ;
- déterminer en temps réel la présence d'un individu dans une foule d'individus ;
- déterminer la probabilité et le niveau de risque d'irruption d'un volcan.
- En analyse de données, calculer l'inverse, lorsqu'il existe, d'une matrice carrée d'ordre 100 ;
- décoder l'ADN d'un être vivant (algorithmes génétiques) ;
- caractériser des services de l'Internet des objets connectés (Internet of Things-IoT) ;
- déterminer le chemin le plus court ou le moins coûteux dans un réseau de relais d'information ;
- reconnaître et comptabiliser des ravageurs en agriculture connectée en utilisant des algorithmes d'apprentissage automatique que l'on appelle le Deep Learning ;
- etc.

La mise en œuvre de l'algorithme consiste en l'écriture de ces instructions dans un langage appelé « langage de programmation » et constitue alors la brique de base d'un programme informatique.

1.4 Définition 2

Un algorithme est une suite finie d'opérations élémentaires, à appliquer dans un ordre déterminé, à des données. Sa réalisation permet de résoudre un problème donné.

La *figure 1*. est une représentation systémique d'un algorithme. En effet, une séquence d'instructions définit et met en œuvre des fonctions de traitement de données. Ces fonctions agissent sur des données en entrée e_1, e_2, \dots, e_n où n est le nombre de ces données en entrée, c'est-à-dire les données qui alimentent l'algorithme. Le traitement utilise ces données pour en produire d'autres.

A la fin du traitement, les données produites sont les données en sortie de l'algorithme s_1, s_2, \dots, s_m où m est le nombre de ces données en sortie et correspondent aux résultats escomptés du traitement.

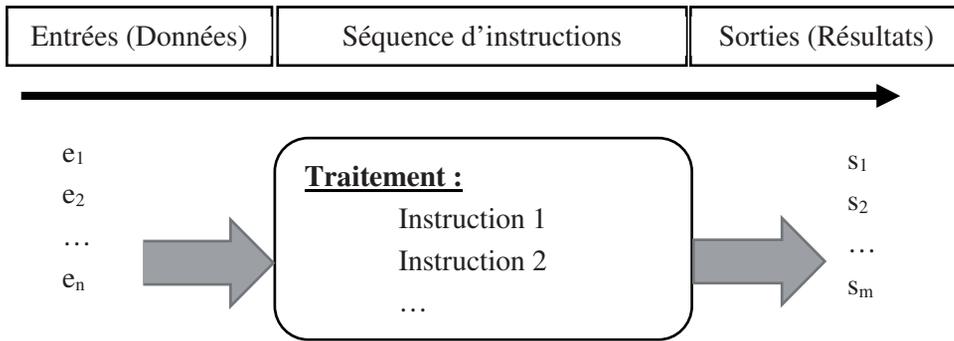


Figure 1. Modèle algorithmique

Prenons comme exemple (figure 2.), le calcul du prix toutes taxes comprises (pttc) d'un article étant donné son prix hors taxes (pht) et sa tva :

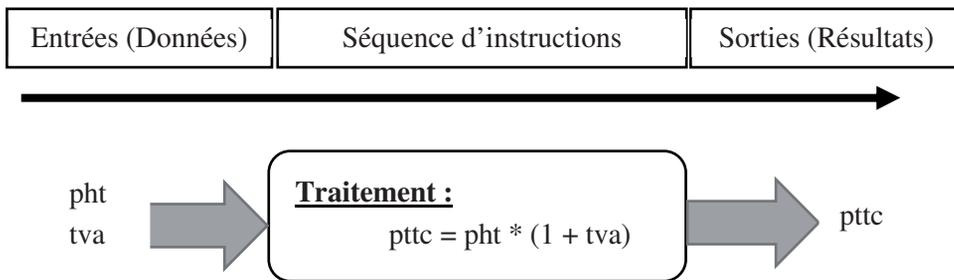


Figure 2. Algorithme calcul pttc

1.5 Ecriture syntaxique

Il y a plusieurs conventions d'écriture d'un algorithme. Le but d'une convention algorithmique est de :

- écrire un algorithme dans un langage proche du langage courant en adoptant une syntaxe d'écriture (une grammaire) ;
- permettre une sorte de standardisation d'écriture, ainsi toutes les personnes avisées puissent faire la même lecture de l'algorithme ;
- faciliter la traduction du langage algorithmique en langage de programmation ;
- de prouver formellement la performance et la convergence d'un algorithme.

Un algorithme peut être considéré comme l'écriture d'un *modèle abstrait de traitement* (c'est-à-dire modèle + méthode de résolution) d'une ou de plusieurs instances d'une situation réelle (figure 3.).

Voici un exemple d'écriture conventionnelle de l'algorithme précédent de calcul du prix pttc :

Algorithme Calcul_PTTC

Variables

pht, tva, pttc : réels

Traitement

Saisir(pht) # Utiliser le # pour un commentaire

Saisir(tva) # comprise strictement entre 0 et 1

pttc = pht * (1 + tva)

Imprimer(pttc)

Fin Algorithme

1.6 Remarques

Pour fonctionner, un algorithme doit donc contenir uniquement des instructions compréhensibles par celui qui devra l'exécuter : *Homme* ou *Machine*.

La taille d'un algorithme en nombre d'instructions ne conditionne pas en soi sa complexité : de longs algorithmes peuvent être finalement assez simples, et de petits algorithmes peuvent être très 'complexes'

1.7 Les instructions fondamentales

Les instructions fondamentales utilisées dans un algorithme sont :

- l'affectation ;
- la lecture / écriture ;
- les tests ;
- les boucles ou répétition.

2 Algorithmes et programmation informatique

L'informatique est la science du traitement automatique de l'information. Pour cela il faut :

- modéliser cette information ;
- définir à l'aide d'un formalisme strict les traitements dont elle fera l'objet ;
- et enfin traduire ces traitements dans un langage informatique compréhensible par un ordinateur et/ou son système d'exploitation.

Les deux premiers points aboutissent à un modèle abstrait de traitement en algorithmique, alors que le dernier le dernier point relève de ce que l'on nomme la programmation informatique.

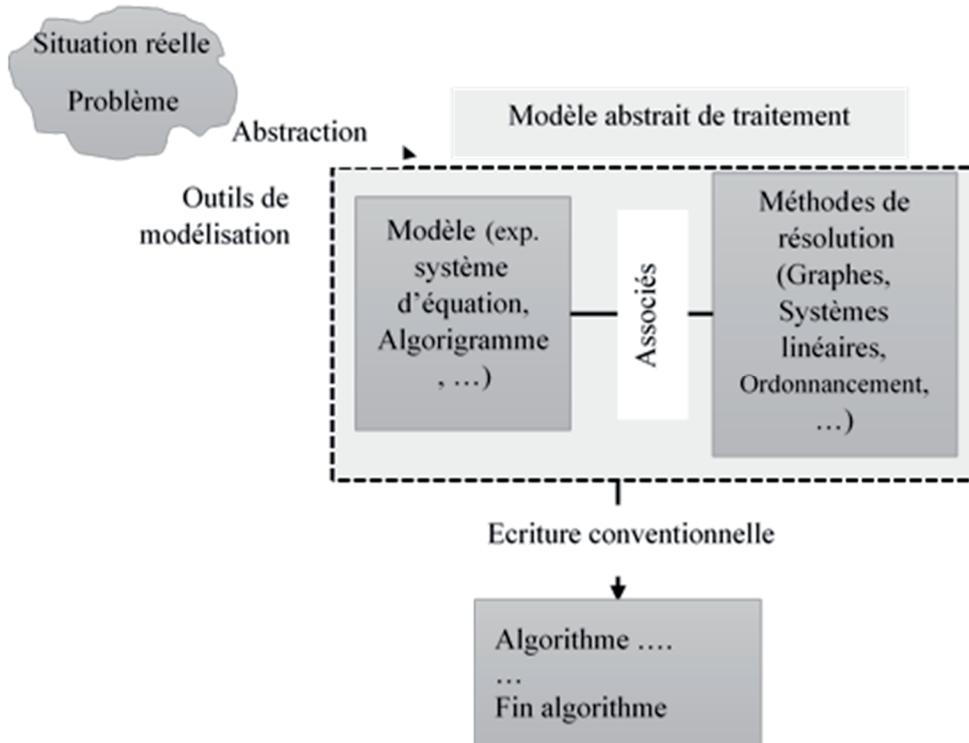


Figure 3. Modèle abstrait de traitement

2.1 Rappels et définitions

L'informatique désigne les sciences et techniques du traitement automatisé de l'information, c'est à dire des données (informations structurées).

Proposer une application ou une solution informatique pour la résolution d'un problème qu'il soit ponctuel ou répétitif, dans un contexte constant ou changeant, revient à :

1. définir quelle information à traiter : ce qui implique une maîtrise de l'encodage et de la représentation des données (systèmes de numérotation de données). Nous abordons ce thème dans la suite de ce chapitre ;
2. définir comment traiter l'information : ce qui implique la définition d'un algorithme de traitement performant c'est-à-dire le meilleur (optimisation du temps de réponse, du temps de résolution, de la mémoire intermédiaire

requis, ...) et le plus adéquat possible (réponse attendue). Les algorithmes en particulier et l'informatique en général sont adossés à des bases mathématiques à savoir l'algèbre de Boole. Nous abordons ces thèmes (c'est-à-dire Algèbre de Boole et Algorithmique) de façon plus étendue dans la suite de ce chapitre ;

3. faire exécuter un algorithme par une machine : ceci relève de la programmation informatique. Nous aborderons également ce thème les chapitres suivants en vue d'applications algorithmiques. A ce titre nous utiliserons le langage de programmation Python qui est un langage de script.

Dans une démarche d'informatisation (automatisation d'une résolution de problème), il y a donc trois phases à caractériser :

1. représentation des données ;
2. algorithme (de traitement) ;
3. programmation (automatisation).

Mais d'ores et déjà, et pour une maîtrise avisée de l'environnement algorithmique, nous expliquons ci-dessous le processus d'exécution d'un algorithme par une machine (un processeur).

2.2 Processus d'exécution d'un algorithme

Dans ce paragraphe (2.2), nous utiliserons exclusivement en texte et bibliographie les productions de *France-IOI* comme référence.

2.2.1 Définition

Un langage informatique est destiné à décrire l'ensemble des actions consécutives qu'un ordinateur doit exécuter. Ceci implique, l'apprentissage, la mise en œuvre puis la pratique du langage choisi, sa syntaxe, son écriture, son débogage et son exécution machine.

2.2.2 Langage machine

Le langage machine, le seul compréhensible par la machine, est le langage utilisé par le processeur (qui utilise le code binaire, suite de 0 et de 1, peu compréhensible par un humain). Ceci implique la connaissance avisée du codage en langage machine (alphabet $\{0, 1\}$), son algèbre (Algèbre de Boole) puis les modes de raisonnement (Algèbre des propositions).

2.2.3 Langage assembleur

Le langage assembleur est un langage intermédiaire, proche du langage machine, mais plus « lisible » par un humain. Le langage assembleur est dépendant du processeur et donc il est donc non portable.

Cette acquisition n'est pas vraiment utile pour les non informaticiens.

2.2.4 Langage informatique

Un langage informatique est par définition différent du langage machine. Un langage informatique dispose et est régi par une grammaire syntaxique.

Un programme informatique ou encore un code source correspond à la traduction d'un ou plusieurs algorithmes par un programmeur ou développeur (qu'il soit Homme et/ou Machine) en utilisant comme traducteur un éditeur de langage informatique.

Les mots d'un langage machine sont écrits à l'aide de l'alphabet {0 ; 1} et sont codés dans le système de numérotation binaire et leur codage est conforme à l'algèbre de Boole.

2.2.5 Langage interprété

Il faut donc le traduire pour le rendre intelligible du point de vue du processeur. Un programme écrit dans un langage interprété a besoin d'un programme auxiliaire (l'interpréteur) pour traduire au fur et à mesure les instructions du programme.

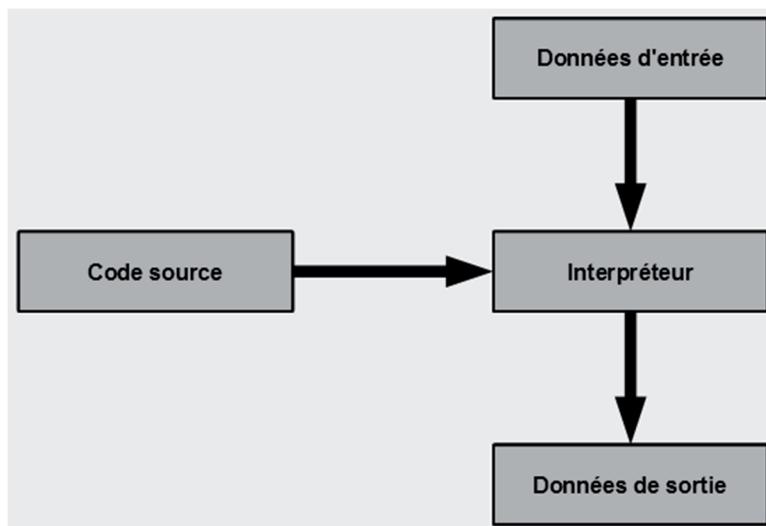


Figure 4. Langage Interprété

En effet, le code source (celui que vous écrivez) est interprété, par un logiciel qu'on appelle *interpréteur*. Celui-ci va utiliser le code source et les données d'entrée pour calculer les données de sortie. L'interprétation du code source est un processus « pas à pas » : l'interpréteur va exécuter les lignes du code une par une, en décidant à chaque étape ce qu'il va faire ensuite.

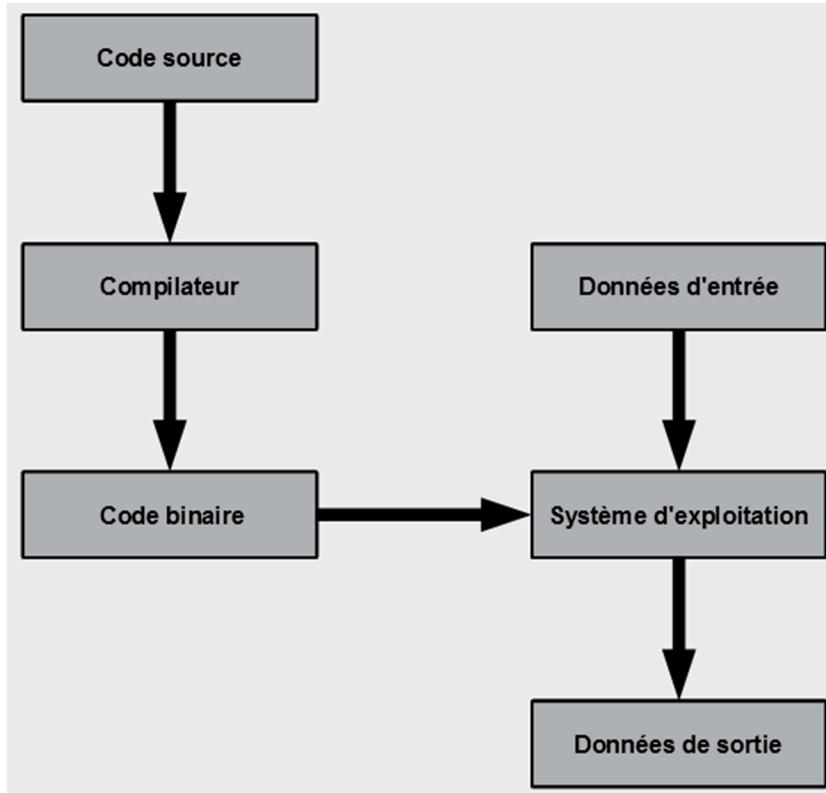


Figure 5. Langage Compilé

2.2.6 Langage compilé

Un programme écrit dans un langage dit compilé va être traduit une fois pour toutes par un programme annexe, appelé compilateur, afin de générer un nouveau fichier qui sera autonome, c'est-à-dire qui n'aura plus besoin d'un programme autre que lui pour s'exécuter ; on dit d'ailleurs que ce fichier est exécutable.

Un programme écrit dans un langage compilé a comme avantage de ne plus avoir besoin, une fois compilé, de programme annexe pour s'exécuter. De plus, la traduction étant faite une fois pour toute, il est plus rapide à l'exécution.