

UE3 – Recherche

les variables en informatique contre les variables en mathématique

INTRODUCTION :

l'utilisation de lettres désignant des objets abstraits, que ce soit des nombres, des fonctions... , est une pratique courante depuis l'antiquité, par l'utilisation de la lettre π par exemples mais aussi de différentes couleurs par Brahmagupta en inde

néanmoins leur utilisation était limité et c'est au 16eme siècle que l'utilisation de variables en informatique devient « formel » par François Viète dans *logistique spéculaire* en utilisant des lettres pour désigner des entités inconnues dans des problèmes mathématiques.

De nos jours, les variables ont gardé en partie ce sens , en mathématique, de symbole représentant à priori un objet indéterminé.

Cette notion de variable est toutefois aussi utilisé en informatique, disciple postérieur à celle des mathématiques, et a un sens homologue mais néanmoins différent de celui des mathématiques.

Ce sont ces dites différences que nous allons voir ensemble

commençons par voir ce qu'est une variable en mathématique

Partie 1 : l'évolution de la notion de variable en mathématique et en informatique

étymologiquement, variable vient de *variābilis* avec "*vari(us)*" signifiant « changeant » et "*-ābilis*" signifiant « capable de », la notion désignerait donc un objet capable de changer

mais la notion de variable va changer au cours de l'histoire des mathématiques en fonction des besoins de l'époque

on peut retrouver très tôt cette notion de variable, dès 250 BC, *Diophantus* écrit *Arithmetica* dans lequel apparaît le symbole « ς » représentant une valeur inconnue

comme dit dans l'introduction, on retrouve par la suite des couleurs en inde néanmoins le concept est formalisé au 16eme siècle lorsque François Viète dans *logistique spéculaire* utilise des lettres pour désigner des entités inconnues dans des problèmes mathématiques

ensuite Euler introduit, après les travaux de newton sur les calculs infinitésimaux, la notation $y = f(x)$ pour une fonction, une variable x et sa valeur y , jusqu'à la fin du 19eme siècle, le mot variable sera beaucoup utilisé pour désigner les argument et valeurs d'une fonction

par la suite, dans la deuxième moitié du 19eme siècle, Karl Weierstrass formalise ce qu'est une limite par : $(\forall \epsilon > 0)(\exists \eta > 0)(\forall x) |x - a| < \eta \Rightarrow |L - f(x)| < \epsilon$,

où aucune des variables n'est considérer comme étant variable (qui peut varier) et à partir de ce moment, la variable va petit à petit arrêter de désigner seulement les arguments et valeurs d'une fonctions

de nos jours, une variable, au sens mathématique, désigne donc un symbole représentant un objet mathématique (nombre, fonction, ensemble....) qui est soit inconnu soit peut être remplacé par une valeur donnée par son ensemble de définition (les nombre entiers par exemple)

contrairement aux mathématique, la notion de variable en informatique n'a pas évolué, c'est un symbole qui associe un nom à une valeur
physiquement, c'est un emplacement mémoire réservé dont la taille dépend du type de la variable.

La définition de cette notion est donc similaire dans les deux contextes, et pour cause les deux contextes sont lié, les différences sont plus visibles dans l'utilisation et les attributs des variables dans les deux contextes, et c'est ce que nous allons voir dans cette deuxième partie

Partie 2: Les points communs et les différences

dans cette partie nous verrons plusieurs aspects liés aux variables dans un contexte ou l'autre, en commençant par un aspect fondamentale: les variables libres et variables muettes

Partie 2.1 : les variables libres et liées

les termes libre et lié sont courants quand on parle des variables en informatique, en mathématique on parlera plutôt de variable libre ou muette

Savoir si une variable est libre ou liée peut être fait de plusieurs façons

renommage :

La première de celles-ci est de renommer la variable à identifier et si le résultat obtenu a la même signification qu'avant le renommage, dans ce cas la variable est liée

substitution :

On peut aussi remplacer la variable à identifier par une valeur, si la phrase a encore du sens, alors c'est que la variable est libre

1. *Pour tout entier naturel k qui divise n , on a $k = 1$ ou $k = n$.*

si on remplace k par 42, on obtient :

Pour tout entier naturel 42 qui divise n , on a $42 = 1$ ou $42 = n$.

ce qui n'a pas de sens, c'est donc que la variable k est liée, or si on remplace n par 42, on obtient :

Pour tout entier naturel k qui divise 42, on a $k = 1$ ou $k = 42$.

cette phrase a un sens, c'est donc que la variable est libre

Mutificateur :

Une autre méthode est d'identifier un mutificateur dans une expression, si une variable est liée à un « mutificateur » c'est que celle-ci est liée, que ce mutificateur soit explicite (1) ou implicite (2)

1.
$$a = \sum_{n=1}^{10} \frac{b^n}{n!}$$

dans ce cas, le mutificateur est le symbole Σ qui lie la variable « n »

2. *Pour tout entier naturel k qui divise n , on a $k = 1$ ou $k = n$.*

dans ce cas, le mutificateur est « *Pour tout entier naturel d* » ce qui rend la variable « d » lié

On peut aussi identifier des mutificateurs dans des algorithmes, par exemple dans l'algorithme suivant :

1. pour chaque i de 1 à n faire :
2. $s \leftarrow s + i$

le mutificateur est « pour chaque i de 1 à n » qui lie la variable « n »

Partie 2.2 : l'état d'une variable

La principale différence entre les 2 notion est la notion d'état

En effet, un algorithme s'exécute à partir d'un état initial, évolue, puis se termine sur un état final. Les variables servent à désigner une valeur qui est fonction d'un état. Or cet aspect n'existe pas dans les proposition et démonstrations mathématiques. Pour représenter une valeur qui évolue en mathématiques, on utilise une suite ou une fonction qui liste explicitement les valeurs successive d'une variable, c'est d'ailleurs la méthode utilisé lors de la preuve d'un algorithme

cette notion d'état permet d'introduire celle d'environnement, l'ensemble des valeurs prise par les variables à un moment de l'exécution d'un programme

Partie 2.3 : les types

Une variable, peut importe le contexte est toujours typé, c'est à dire qu'on connaît exactement la nature de l'objet désigné par la variable

Un type en mathématique peut être exprimé de façon explicite :

1. *soit un entier n*

dans ce cas là, n est explicitement définie comme étant un entier

mais le type peut aussi être définie de façon implicite :

2. *k est pair si et seulement si il est un multiple de 2*

dans ce cas, k n'a à priori pas de type, mais son type peut être déduit par la reste de la phrase, un multiple de 2 est forcément un nombre, et non une fonction par exemples

Néanmoins, cette notion reste assez flou en mathématique, selon le contexte, un nombre peut être un type, d'autres fois on distinguera dans les nombre, les entiers positifs et négatifs

En revanche, en informatique la notion de type est très précise et le spectre des type est donnée par le langage de programmation utilisé

Un type correspond physiquement non seulement à la place nécessaire en mémoire pour stocker une valeur de ce type, un *long int* prendra 32 bits alors qu'un *short int* en prendra 16, mais il renseigne

aussi sur ce qui est codé, le nombre binaire «10000001» ne code pas le même chiffre en binaire (129) et en complément à deux (-127)

Le nom de la variable est très souvent accompagné par son type lors de la déclaration de la variable, en c on peut retrouver :

1. `int x = 2`

ici, le type de x est *int*

mais il existe des langages dit dynamique, ou le type d'une variable est définie par la valeur que va prendre celle ci ,on peut retrouver en python:

1. `x = 1`

2. `k = 1`

au premier abord, on peut penser que x et k partagent le même type, mais non, x est un *integer* alors que k est un *float*, le type d'un nombre peut alors évoluer au cours du programme, on peut très bien écrire :

1. `x = 'hello world '`

2. `x = 4`

x « change » du type *string*, au type *int*

Partie 2.4 : la portée

Une autre notion associée à celle de variable est la portée de celles-ci, c'est à dire leur « durée de vie » dans le discours, la partie du programme/de la démonstration dans laquelle le nom de la variable est définie, ce terme est très utilisé en informatique mais est absent du vocabulaire mathématique malgré la présence de cette notion

En programmation, la portée d'une variable est toujours bien définie due au caractère formel des langages de programmation, ainsi, si on prend la fonction en C suivante :

1. `Bool even(int x)`

2. `{`

3. `return x%2 == 0 ;`

4. `}`

la variable *x* est un argument de la fonction *even* , sa portée est donc celle de la fonction (l'ensemble des instructions compris entre les accolades, on parle de bloc), cela veut dire que en dehors de la fonction, le *x* de cette fonction n'existe pas à l'extérieur de celle-ci(il peut exister une variable *x* en dehors de la fonction, mais il n'y a aucune raison pour que celle-ci soit de même valeur/type que le *x* de cette fonction), alors que le nom *even* est définie en dehors des accolades(du bloc), on peut donc utiliser *even* à l'extérieur de celle-ci mais aussi à l'intérieur, c'est d'ailleurs le principe des fonction récursives

En mathématique, comme dit en introduction, la notion de portée est implicite, une variable est introduite par une phrase :

- soit k un entier...
- on a $m=k^2$...
- pour tout i inférieur à ...

la portée de ces différentes variables k, m, i, \dots est limité par les phrases dans lesquelles celles-ci sont contenues, dans la phrase :

1. Pour tous entiers a et b , pour tous entiers u et v , si $au + bv = 1$ alors a et b sont premiers entre eux et u et v sont premiers entre eux.

Les variables a, b, u, v existent dans le contextes de la phrase mais, comme en informatique, leurs noms pourront être réutiliser dans une autres sans garanties que celles-ci soit de même valeurs/types que celles de la phrase

Ces notions, notamment en informatique, ne sont souvent peu voir pas comprise par les élèves, c'est ce que j'ai pue voir lors du stage que j'ai fais, notamment à cause de la confusion avec la variable en mathématique sur certains concepts, surtout le type ainsi que l'état d'une variable

bibliographie :

Stage de logique par René Cori, les 3 et 4 décembre 2009

http://maths.ac-reunion.fr/IMG/pdf/Conference_Logique_Rene_Cori.pdf

Variables Explained Away Author(s): Willard V. Quine

<https://cpb-us-w2.wpmucdn.com/voices.uchicago.edu/dist/9/177/files/2008/05/quine-variables-explained-away.pdf>

Du numérique au littéral ,mars 2016 , eduscol

https://cache.media.eduscol.education.fr/file/Programmes/17/3/du_numerique_au_litteral_109173.pdf

Sur la notion de variable, entre mathématiques et informatique Emmanuel Beffara 9 décembre 2017

<http://www.i2m.univ-amu.fr/perso/emmanuel.beffara/docs/texte-variables.pdf>

Structure and Interpretation of Computer Programs by Harold Abelson and Gerald Jay Sussman with Julie Sussman

<https://mitpress.mit.edu/sites/default/files/sicp/full-text/book/book.html>

Structure and Interpretation of Computer Programs by Harold Abelson and Gerald Jay Sussman with Julie Sussman

<http://w.jarda.peregrin.cz/mybibl/PDFTxt/414.pdf>