Algorithmique

CHAPITRE 3 : EXPRESSIONS ET OPÉRATIONS

Sommaire

1. Définitions

- Dans une instruction d'affectation, on trouve :
  - A gauche de la flèche, le nom de la variable et uniquement cela. Si jamais, on voit un jour à gauche d'une flèche d'affectation autre chose qu'un nom de variable on peut être certain à 100% qu'il s'agit d'une erreur.
  - A droite de la flèche, on retrouve ce qu'on appelle une **expression** qui est un ensemble de valeurs, reliées par des opérateurs et équivalent à une seule valeur. Par exemple dans $A <- 2 + 3$, on a A le nom de la variable et $2 + 3$ l'expression qui correspond à la valeur 5.

- Une expression n'est pas seulement constituée de numérique, on peut retrouver une expression du type $A <- -toto + 2$, avec $toto$ une variable.

- L'expression doit être du même type que la variable à laquelle on l'affecte (cas des langages typés). On ne peut pas mettre une expression de type texte dans une variable de type entier sinon cela déclenchera une erreur.

- **Un opérateur** est un signe qui relie deux valeurs, pour produire un résultat. Ainsi, dans l'expression $2 + 3$, on a l'opérateur $+$ qui relie les valeurs $2\ et\ 3$ pour produire le résultat 5.

Dans les langages non typés toute variable peut recevoir tout type d'expression.

2. Opérateurs numériques

- Les opérateurs numériques sont constitués des quatre opérateurs classiques :
  - $+$: pour l'addition
  - $-$: pour la soustraction
  - $*$: pour la multiplication
  - $/$: pour la division
- On peut retrouver aussi le ^ pour désigner la puissance. Par exemple 2 au carré s'écrira 2^2.
- On peut utiliser les parenthèses pour la priorité des opérations. En informatique, on utilise les mêmes règles de priorités qu'en mathématiques ainsi, la multiplication et la division ont priorité sur l'addition et la soustraction.

Exemple : Ecrire un algorithme qui effectue le calcul de $12 * (3 + 5)$

```
Algo: Calcul de "12*(3 + 5)"

Début
    Variables totalAddition, total en Entier
    totalAddition <- 3 + 5
    total <- 12 * totalAddition
Fin
```

3. Opérateurs alphanumérique

- Supposons qu'on veuille mettre le contenu suivant dans une variable "***Le résultat de 2 + 3 est valeur_du_resultat***", avec ***valeur_du_resultat*** le contenu d'une variable entière, comment procéder ?

- Pour associer une valeur chaine de caractères à une valeur entière, on va utiliser l'opération de **concaténation** dont l'opérateur varie d'un langage à l'autre.

- Pour ce cours nous utiliserons l'opérateur $+$ pour effectuer des concaténations.

Exemple :

```
somme <- 2 + 3
total <- "Le résultat de 2 + 3 =" + somme
```

- La variable *total* contiendra la chaine de caractères : "*Le résultat de* $2 + 3 = 5$".

- On peut aussi associer deux variables chaines par concaténation.

```
nom <- "GREEN"
prenom <- "OSCAR"
nomComplet <- nom + prenom // Ici nomComplet contiendra "GREENOSCAR" en un seul mot

nom <- "GREEN"
prenom <- "OSCAR"
nomComplet <- nom + " " + prenom // Ici nomComplet contiendra "GREEN OSCAR" en deux mots
```

4. Opérateurs logiques ou booléens

- Les opérateurs logiques nous permettent de relier des expressions pour produire un résultat qui soit $VRAI$ ou $FAUX$.
- On en dénombre 4 types qui sont : OU, ET, NON et XOR (appelé Ou exclusif).
- Pour comprendre les types booléens, nous allons les appliquer à des exemples:
  - $\boldsymbol{A < -(2 > 3)\ ET\ (4 < 5)}$, donnera comme résultat $FAUX$, car pour que se soit vrai il faut que chacune des expressions de part et d'autre soient vraies or $(2 > 3)$ est faux et $(4 < 5)$ est vrai. **Que donnera donc cette expression $\boldsymbol{B < -\big(6 = (5 + 1)\big)\ ET\ (6 < 8)\ ET\ (123 < 234)}$?**
  - $\boldsymbol{C < -(2 > 3)\ OU\ (4 < 5)}$, donnera comme résultat VRAI, pour l'opérateur OU, il suffit qu'une seule valeur soit vraie pour que le résultat de l'ensemble soit VRAI. Comme $(4 < 5)$ est vrai alors $(2 > 3)\ OU\ (4 < 5)$ est donc vrai**. Que donnera donc D $\boldsymbol{< -\big(6 = (5 + 1)\big)\ OU\ (6 < 8)\ OU\ (123 < 234)}$?**
  - L'opérateur $\boldsymbol{NON}$, est assez spécial, en fait il renvoie le résultat du contraire de l'expression. Prenons l'exemple de $\boldsymbol{E < -\ NON\ (2 > 3)}$, pour trouver le résultat de cette expression, on procède comme suit:
    - On vérifie l'expression sans le NON donc pour l'exemple on a : $2 > 3$ qui est FAUX.
    - Dès qu'on trouve le résultat de l'expression sans le NON, on prend comme résultat le contraire. Ainsi $NON(2 > 3)$ renverra VRAI car $NON(FAUX) = VRAI$.
- L'opérateur XOR, contrairement à l'opérateur OU ici toutes les valeurs ne doivent pas être vraies, c'est soit l'une ou l'autre pas les deux. Ainsi $A < -(2 < 3)\ XOR\ (4 < 5)$, renverra $\boldsymbol{FAUX}$ car $2 < 3$ est VRAIE ainsi que $4 < 5$.

4. Opérateurs logiques ou booléens

- Pour comprendre la différence entre le $\boldsymbol{OU}$ et le $\boldsymbol{XOR}$ (OU Exclusif), nous allons prendre l'exemple suivant :
  - Supposons un client dans une boutique qui dit : « **Je veux un habit rouge ou noir** ». Si le vendeur lui ramène un **habit rouge** il aura raison ainsi qu'un **habit noir** il aura aussi raison donc **VRAI**, même si le vendeur envoie au client les deux couleurs habit rouge et habit noir, il aura aussi raison. Pour une autre couleur ça sera du **FAUX**. C'est ça l'opérateur **OU.**
  - Par contre supposons que le client dise : « **Je veux soit un habit rouge, soit un habit noir** ». Là, la réaction n'est plus la même, si le vendeur envoie uniquement un **habit rouge** il sera dans le vrai, s'il envoie uniquement un habit noir il sera aussi dans le vrai. Par contre s'il envoie un habit rouge et un habit noir il sera dans le faux car le client veut soit l'un ou l'autre pas les deux. C'est le rôle de l'opérateur **XOR.**
- Supposons maintenant que le client dise : « **Je veux un habit rouge et un habit noir** ». Pour que le vendeur soit dans le vrai, il faut qu'il apporte deux habits l'un de couleur rouge et l'autre de couleur noire. Le client ici a besoin de deux habits pas d'un seul donc toute réponse à sa demande différente des couleurs rouge et noire est fausse. C'est le rôle de l'opérateur **ET.**
- Supposons que le client dise : « **Je ne veux pas un habit rouge** ». Ici, si le vendeur apporte un habit de couleur noire il aura raison car le client veut tout sauf le rouge. Donc tout ce qui est NON(rouge) est vrai. C'est le rôle de l'opérateur **NON.**
- L'on peut aussi combiner les opérateurs comme : $\big((5 < 10)\,OU\,(10 < 20)\big)\;ET\;(1 < 2)$.
- En pseudo code nous utiliserons : $\boldsymbol{OU/Ou; ET/Et; XOR/Xor; NON/Non.}$

5. Variables mathématiques et variables informatiques

- Bien que la notion de variable soit utilisée en mathématiques comme en informatique, il faut noter une différence de la notion de variable dans chacun.

- En mathématiques, une variable est généralement une inconnue, qui recouvre un nombre non précisé de valeurs.

  <u>Exemple</u>: Dans l'équation $y = 3x + 2$, les variables x et y valables pour l'équation existent en nombre infini. On a une infinité de valeurs de x et y qui résolvent l'équation.

- En informatique, une variable possède une valeur à un moment donné et une seule. Contrairement aux mathématiques où une variable est une inconnue, en informatique, la variable n'est pas une inconnue au contraire elle est connue et contient une valeur.

- Une autre différence est l'emploie du signe =, qui en mathématiques permet d'affirmer que deux valeurs sont identiques. En informatique, dans la quasi-totalité des langages de programmation, le signe = est utilisé pour affecter une valeur à une variable.

- Pour éviter toute confusion, nous utiliserons dans ce cours l'opérateur $<-$, pour l'affectation et l'opérateur $=$, pour la comparaison comme en mathématiques. Ainsi $\mathbf{5 = (4 + 1)}$ en mathématiques aura le même sens que $\mathbf{5 = (4 + 1)}$ en pseudo-code pour ce cours.

Avant de traduire un algorithme dans un langage de programmation, veuillez lire la documentation pour savoir quelles sont les règles à appliquer dans ce langage. Le pseudo-code n'est pas un langage de programmation car il peut varier d'un développeur à l'autre.

6. Exercices

Exercice 1

Que produit l'algorithme suivant ?

```
Début
    Variables A, B, C en Chaînes
    A <- "423"
    B <- "12"
    C <- A + B
Fin
```

Exercice 2

Que produit l'algorithme suivant ?

```
Début
    Variables A, B, C en Entiers
    A <- 423
    B <- 12
    C <- A + B
Fin
```

6. Exercices

Exercice 3

Que produit l'algorithme suivant ?

```
Début
    Variables phrase, nom en Chaînes
    Variable age en Entier
    nom <- "Oscar"
    age <- 12
    phrase <- nom + " a " + age + " ans"
Fin
```

7. Corrections

Exercice 1

Que produit l'algorithme suivant ?

```
Début
    Variables A, B, C en Chaînes
    A <- "423"
    B <- "12"
    C <- A + B
Fin
```

On exécute cet algorithme comme suit:

A <- "423" implique que la variable A contient la chaîne de caractères 423

B <- "12" implique que la variable B contient la chaîne de caractères 12

C <- A + B ici on affecte à la variable C la somme de A et B or ce sont des variables de type chaîne de caractères donc l'addition ici est une concaténation.

La variable C contient donc la chaîne de caractères 42312.

7. Corrections

Exercice 2

Que produit l'algorithme suivant ?

```
Début
    Variables A, B, C en Entiers
    A <- 423
    B <- 12
    C <- A + B
Fin
```

On exécute cet algorithme comme suit:

A <- 423 implique que la variable A contient le nombre 423

B <- 12 implique que la variable B contient le nombre 12

C <- A + B ici on affecte à la variable C la somme de A et B, comme A et B sont des entiers alors on effectue une addition.

La variable C contient donc le nombre 435.

7. Corrections

Exercice 3

Que produit l'algorithme suivant ?

```
Début
    Variables phrase, nom en Chaînes
    Variable age en Entier
    nom <- "Oscar"
    age <- 12
    phrase <- nom + " a " + age + " ans"
Fin
```

```
On exécute cet algorithme comme suit:
nom <- "Oscar" ici on affecte à la variable nom la chaîne de caractères Oscar
age <- 12 ici on affecte à la variable age le nombre 12
phrase <- nom + " a " + age + " ans " ici on effectue une concacténation de :
+ la variable chaîne nom
+ la valeur " a "
+ la variable entière age
+ la vaaleur " ans"
La variable phrase contient donc la chaine de caractères "Oscar a 12 ans"
```

Citations

« Compter en octal, c'est comme compter en décimal, si on n'utilise pas ses pouces. » Tom Lehrer

« A l'origine de toute erreur attribuée à l'ordinateur, vous trouverez au moins deux erreurs humaines. Dont celle consistant à attribuer l'erreur à l'ordinateur. » Anonyme

« Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Çà, se sont les caractéristiques de la magie. » Dave Small