

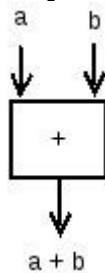
Connexions

par Jacques Bailhache

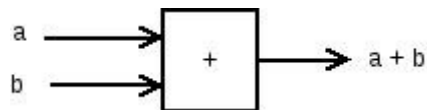
Juillet 2017

Représentation graphique des fonctions

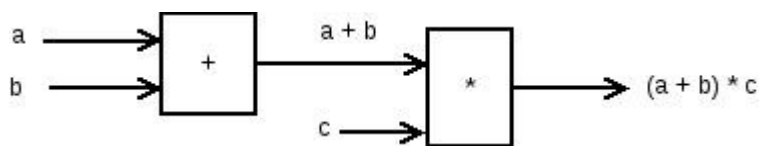
On peut représenter les opérations ou fonctions par des « boîtes » avec des entrées pour les arguments et une sortie pour le résultat, par exemple $a + b$ peut être représenté par :



ou :



Ces « boîtes » peuvent être interconnectées pour représenter des expressions plus complexes, par exemple :

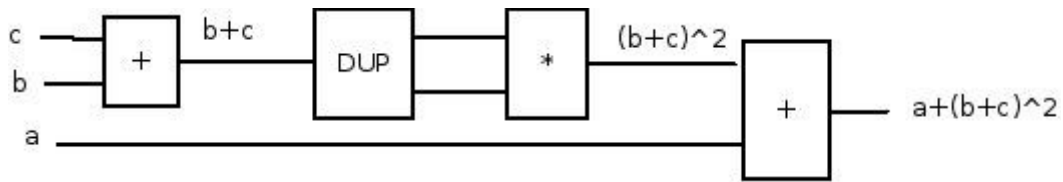


Expressions en notation postfixée

Considérons par exemple le programme FORTH :

```
+ DUP * +  
a b c - - a + (b + c) ^ 2
```

Ce programme peut aussi être représenté graphiquement :

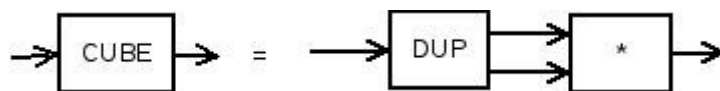


et peut être vu comme un produit tensoriel basé sur les opérations « et » et « ou » logiques de tenseurs dont les indices sont des nombres et les valeurs des booléens.

La suite d'opération « DUP * » correspond au cube que l'on peut définir en Forth par :

: CUBE DUP * ;

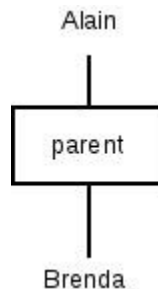
ce qui peut se représenter graphiquement par :



Relations

On peut également représenter des relations de la même façon.

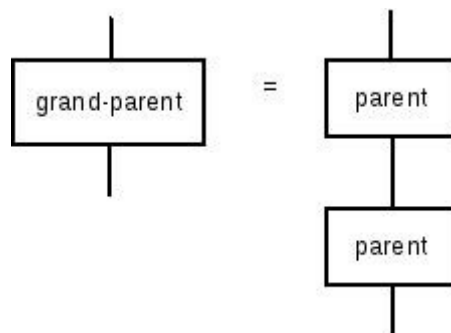
Par exemple « Alain est parent de Brenda » peut être représenté par :



On peut également combiner les relations par interconnexion, par exemple :

« X est grand-parent de Z si x est parent de Y et Y est parent de Z »

peut se représenter par :



Bases de données relationnelles

Une relation telle que la relation « parent » peut être représentée par une table dans une base de données relationnelle, par exemple :

```

CREATE TABLE parents(parent VARCHAR(30), enfant VARCHAR(30));
INSERT INTO parents(parent,enfant) VALUES('Alain','Bertrand');
INSERT INTO parents(parent,enfant) VALUES('Aude','Bertrand');
INSERT INTO parents(parent,enfant) VALUES('Alain','Brenda');
INSERT INTO parents(parent,enfant) VALUES('Aude','Brenda');
INSERT INTO parents(parent,enfant) VALUES('Ben','Charles');
INSERT INTO parents(parent,enfant) VALUES('Brenda','Charles');
INSERT INTO parents(parent,enfant) VALUES('Charles','Daniel');
INSERT INTO parents(parent,enfant) VALUES('Cecile','Daniel');

```

On peut chercher les grands-parents de Charles par cette requête :

```

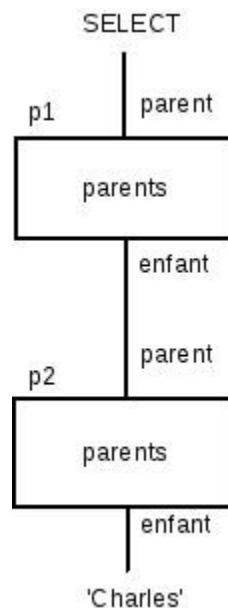
SELECT p1.parent
FROM parents p1
JOIN parents p2 ON p2.parent = p1.enfant
WHERE p2.enfant = 'Charles';

```

qui donne le résultat :

Alain
Aude

et que l'on peut représenter graphiquement par :



Calcul matriciel et tensoriel

Un vecteur peut être représenté par un tableau à une dimension d'éléments appartenant à un certain ensemble (par exemple les nombres réels ou les nombres complexes).
Il peut aussi être vu comme une fonction qui à un nombre entier (appelé indice) associe par exemple un nombre réel.

Exemples :

Vecteur X :

3,5
-1
2

Ce vecteur peut être vu comme une fonction qui à 1 associé 3,5, à 2 associe -1 et à 3 associe 2.

Vecteur Y :

2
1,5
-3

Transposition

La transposition consiste à permuter lignes et colonnes.

Par exemple, le transposé du vecteur colonne X est le vecteur ligne 'X :

3,5	-1	2
-----	----	---

Produit scalaire

Le produit scalaire de 2 vecteurs peut être obtenu en multipliant les éléments de même indice et en faisant la somme des produits obtenus :

$$X \cdot Y = 3,5 * 2 + (-1) * 1,5 + 2 * -3 = -0,5$$

D'une façon générale :

$$X \cdot Y = \sum_{i=1}^n x_i y_i$$

Notation matricielle

$$X \cdot Y = {}^t X Y = \sum_{i=1}^n x_i y_i$$

Par convention, on note en indice les indices des vecteurs lignes et en exposant les indices des vecteurs colonnes.

Notation d'Einstein

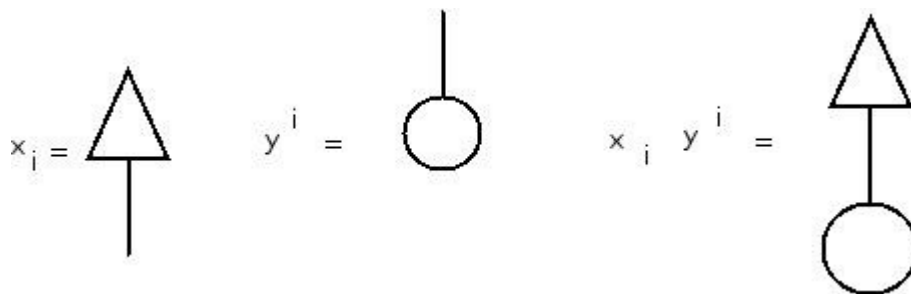
On simplifie la notation en omettant les Σ , les sommations se faisant implicitement sur les indices et exposants correspondant.

Exemple : $x_i y^i$

Notation diagrammatique de Penrose

Dans cette notation que Roger Penrose expose dans son livre « A la découverte des lois de l'univers » (12.8 p 232), les indices sont représentés par des lignes tracées vers le bas (« pattes ») et les exposants vers le haut (« bras »). Penrose utilise des formes géométriques différentes pour représenter les différents tenseurs.

Exemple :



Autre notation graphique

Personnellement, je préfère représenter les tenseurs par des rectangles contenant la lettre qui les désigne, et représenter les indices par des liaisons vers la droite et les exposants par des liaisons vers la gauche, ce qui permet d'avoir une disposition similaire à celle de la notation matricielle.

Exemple : produit d'un vecteur-ligne $X (x_i)$ et d'un vecteur-colonne $Y (y^i)$:



Matrices

Une matrice est un tableau à 2 dimensions qui associe un nombre (réel ou complexe) à 2 indices.

Exemple :

Matrice A :

1	0	2
-1	3	1

0	1	0
---	---	---

Produit d'un vecteur par une matrice :

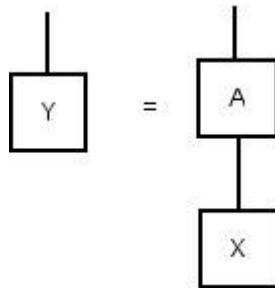
$$y^i = \sum_{j=1}^3 a_j^i x^j$$

7,5
-4,5
-1

En notation d'Einstein : $y^i = a_j^i x^j$

En notation matricielle : $Y = A X$

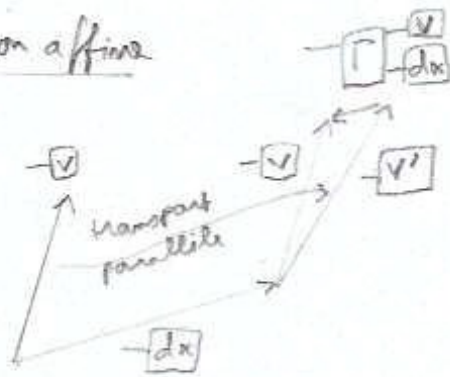
En notation graphique :



Ces notations (sauf la notation matricielle limitée aux tableaux bidimensionnels) se généralisent à des « tenseurs » qui représentés par des tableaux à un nombre quelconque de dimensions.

Exemples de tenseurs en notation graphique :

Connexion affine



dérivée partielle

$$\boxed{v}$$

$$\boxed{v}$$

dérivée covariante

$$\boxed{v} \circlearrowleft = \boxed{v} + \Gamma \boxed{v}$$

$$\boxed{v} \circlearrowright = \boxed{v} - v \Gamma$$

changement de base

$$\boxed{v} = \boxed{P} \boxed{v'}$$

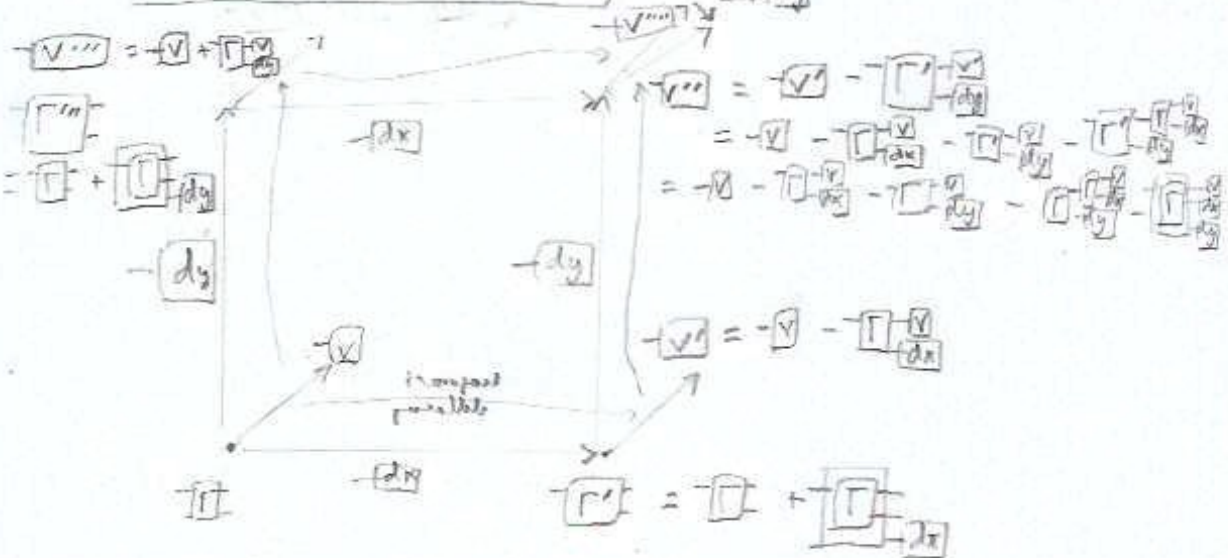
$$\boxed{A} = \boxed{P} \boxed{A'} \boxed{P}^{-1}$$

$$\boxed{\Gamma} = \boxed{P} \boxed{\Gamma'} \boxed{P}^{-1} + \boxed{P} \boxed{P}^{-1}$$

courbure

$$\boxed{R} \begin{matrix} \boxed{v} \\ \frac{d}{dx} \\ \boxed{v} \end{matrix} = \boxed{v''} - \boxed{v''''}$$

$$\boxed{R} = \boxed{\Gamma} \boxed{\Gamma} - \boxed{\Gamma} \boxed{\Gamma} + \boxed{\Gamma} \boxed{\Gamma} - \boxed{\Gamma} \boxed{\Gamma}$$



Rapport entre la notation graphique des relations et la notation diagrammatique de Penrose des tenseurs

Considérons par exemple une table avec une seule colonne de type entier sur 8 bits (-128 à 127), définie en SQL par :

```
CREATE TABLE X(i TINYINT) ;
```

et remplie par :

```
INSERT INTO X(i) VALUES(-1) ;  
INSERT INTO X(i) VALUES(3) ;  
INSERT INTO X(i) VALUES(4) ;
```

Cette table peut être vue comme un vecteur d'un espace à 256 dimensions représenté par un tableau d'une colonne et 256 lignes numérotées de -128 à 127 où les lignes numéro -1, 3 et 4 contiennent la valeur « Vrai » et les autres la valeur « Faux ».

Définissons une autre table à 2 colonnes de type TINYINT :

```
CREATE TABLE A(j TINYINT, k TINYINT);
```

```
INSERT INTO A(j,k) VALUES(5,1);  
INSERT INTO A(j,k) VALUES(6,3);  
INSERT INTO A(j,k) VALUES(8,3);  
INSERT INTO A(j,k) VALUES(8,4);  
INSERT INTO A(j,k) VALUES(9,4);  
INSERT INTO A(j,k) VALUES(10,5);
```

Cette table pourrait être représentée par un tableau à 256 lignes et 256 colonnes numérotées de -128 à 127 où les cases correspondant aux valeurs présentes contiendraient la valeur « Vrai » et les autres la valeur « Faux ».

La jointure :

```
SELECT DISTINCT A.j FROM A JOIN X ON A.k = X.i;
```

que l'on peut aussi écrire :

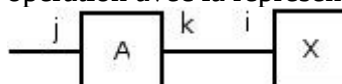
```
SELECT DISTINCT A.j FROM A, X WHERE A.k = X.i;
```

donne les résultats :

6
8
9

Cette opération peut être vue comme le produit de A considéré comme matrice de booléens par X considéré comme vecteur colonne de booléens, en remplaçant la multiplication par le « et » logique et l'addition par le « ou » logique.

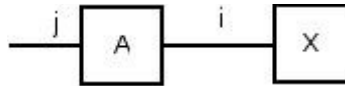
On peut également représenter cette opération avec la représentation graphique :



Conventions de simplification et de clarification des représentations graphique des requêtes

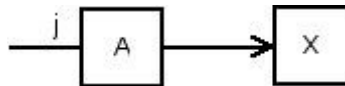
Dans le cas où les colonnes sur lesquelles on fait la jointure ont le même nom dans les deux tables, on peut simplifier la représentation :

```
SELECT DISTINCT A.j FROM A JOIN X ON A.i = X.i ;
```



Si la colonne sur laquelle se fait la jointure correspond à l'identifiant d'une des tables, on peut la représenter par une flèche dirigée vers cette table :

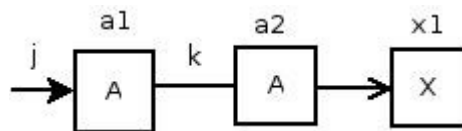
```
SELECT DISTINCT A.j FROM A JOIN X ON A.id_x = X.id_x ;
```



Si une table apparaît plus d'une fois dans une jointure, on doit nommer chaque occurrence, par exemple :

```
SELECT DISTINCT a1.j FROM A a1 JOIN A a2 ON a1.k = a2.k JOIN X x1 ON a2.id_x = x1.id_x ;
```

Dans ce cas, on peut indiquer le nom de l'occurrence de la table au-dessus de celle-ci :



Autres exemples de conventions graphiques pour représenter les requêtes SQL



Table

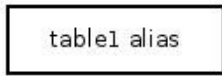


Table avec alias



Table à droite d'un LEFT JOIN

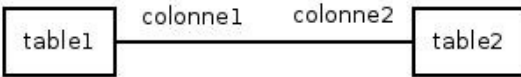


table1 JOIN table2 ON table1.colonne1 = table2.colonne2

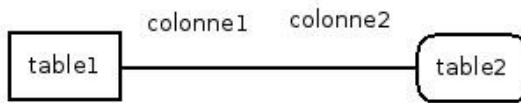


table1 LEFT JOIN table2 ON table1.colonne1 = table2.colonne2

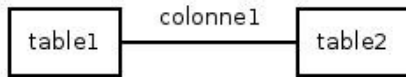


table1 JOIN table2 ON table1.colonne1 = table2.colonne2

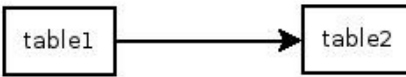
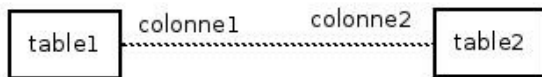
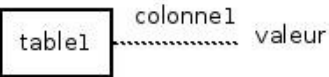


table1 JOIN table2 ON table1.id_Table2 = table2.id_Table2



WHERE table1.colonne1 = table2.colonne2



WHERE table1.colonne1 = valeur