Expressions et fonctions

Pour un système de calcul formel comme *Maple*, la question de la représentation des expressions symboliques est cruciale.

I. Expressions

Celles-ci doivent être représentées en interne de façon unique et non ambiguë, p. ex. afin de pouvoir juger de leur égalité.

À cet effet, *Maple doit* effectuer automatiquement certains regroupements.

Substitutions automatiques

Voici quelques remplacements que *Maple* effectue systématiquement[1].

expression	substitution	commentaire
$x + x$	$2x$	
$ax + bx$	$(a+b)x$	si a, b affectées
xx	x^2	
$x^a x^b$	$x^{(a+b)}$	si a, b affectées
$\sin(\pi/6)$	$1/2$	valeurs particulières
$\sqrt{20}$	$2\sqrt{5}$	

Ordre de calcul

Maple respecte généralement les conventions algébriques usuelles. On aura donc, du plus au moins prioritaire :

- les fonctions unaires (**exp**, **sin**, ...) ;
- les exponentiations **^** ;
- les multiplications ***** (et les divisions) ;
- les additions **+** (et les soustractions) ;
- les booléens (**=**, **<**, ...).

Ces derniers sont si bas dans la hiérarchie qu'ils ne non pas évalués automatiquement (il faut en forcer l'évaluation avec **evalb**) ([2]).

À niveau de priorité égal, les calculs sont faits de gauche à droite.

Une exception notable est **a^b^c**, qui déclenche une erreur[3].

Le type d'une expression est celui de son opérateur de plus bas niveau.

---

1 Il est pratiquement impossible d'empêcher *Maple* d'effectuer ces regroupements.
2 Cette disposition est indispensable pour pouvoir mettre en place des (in)équations.
3 Mathématiquement, cela aurait un sens (**a^(b^c)**) *contraire* à la convention "de gauche à droite" de *Maple*.

Conventions d'affichage

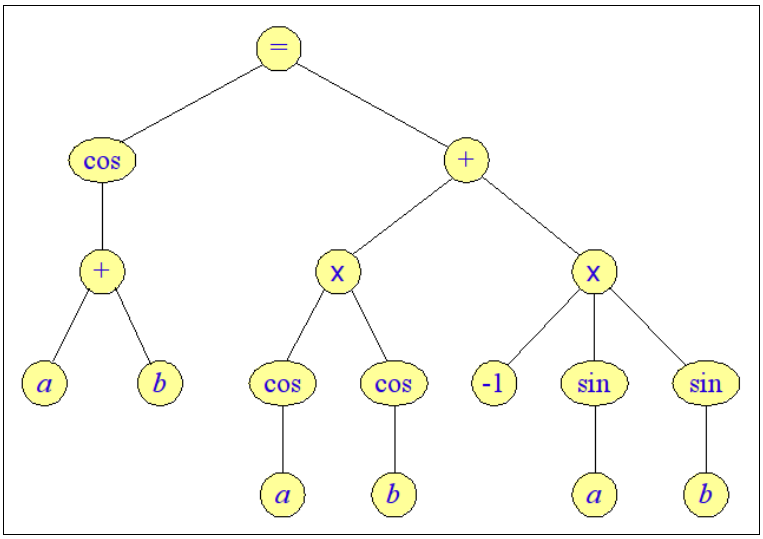
Parfois la représentation interne d'une expression n'est pas celle que *Maple* affiche :

affichage	représentation interne
$x - y$	$x + (-1) * y$
x / y	$x * y \wedge (-1)$
$\sqrt{x}$	$x \wedge (1/2)$

Représentation arborescente...

Considérons l'expression $\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$.

Nous disons que *Maple* la représente sous la forme d'un arbre :

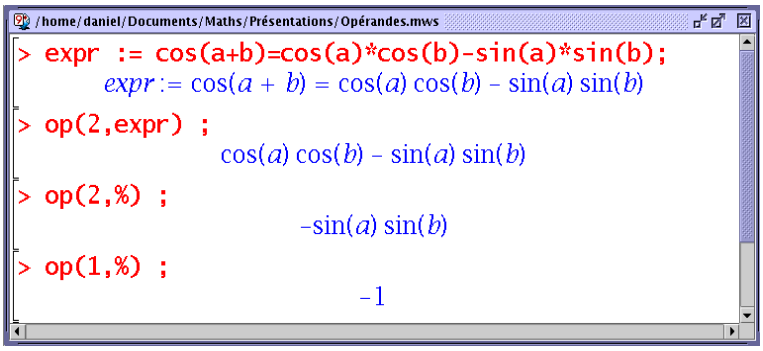


L'instruction **op**, qui sert à extraire les opérandes de toutes les expressions *Maple*, va nous aider à l'explorer.

Sa syntaxe générale est :

op(*indice* **,** *expression* **) ;**

Si l'on extrait ses opérandes successifs...



... on se rend compte de deux choses :

1. Les opérandes sont bien organisés en une arborescence descendante comme dans le schéma ci-dessus. (Ainsi, "– sin(*a*) sin(*b*)" est bien le deuxième opérande du deuxième opérande de *expr*.) ;
2. Le "-1" est le *premier opérande de* "– sin(*a*) sin(*b*)", confirmant les conventions d'affichage évoquées plus haut. (Il n'y a pas de "différence" dont "sin(*a*) sin(*b*)" serait le *deuxième* opérande.)

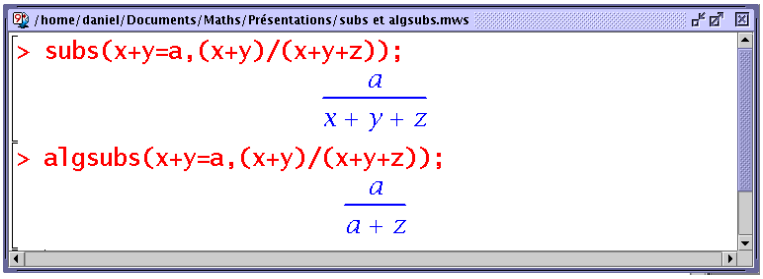
Substitution

L'instruction **subs** permet d'effectuer un remplacement dans une expression, selon la syntaxe :

subs(*motif* **=** *remplacement* **,** *expression* **) ;**

Attention, **subs** n'agit que sur les *opérandes* de l'expression (résultats de **op**).

Pour un examen plus détaillé (et plus mathématique) de l'expression, on pourra utiliser **algsubs**.



Dans l'exemple ci-dessus, "*x*+*y*" n'est effectivement présent *qu'une fois* dans la *représentation* de l'expression (*x*+*y*)/(*x*+*y*+*z*) ([4]).

Seul **algsubs** peut détecter le "*x*+*y*" dans "*x*+*y*+*z*".

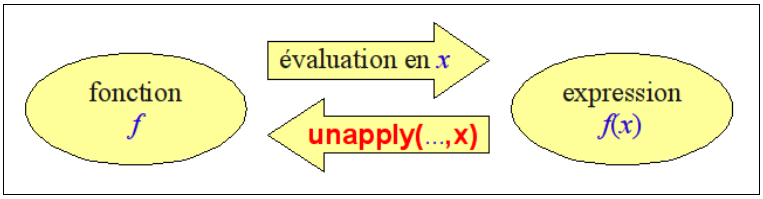
Plus de détails sur **subs** dans le poly. "Évaluation".

II. Fonctions

Dans tout système de calcul formel, la différence entre *fonction* et *expression* est cruciale.

- une *expression* est un objet algébrique inerte ;
- une *fonction* prend un ou plusieurs arguments et produit un résultat.

On convertit facilement l'une en l'autre :

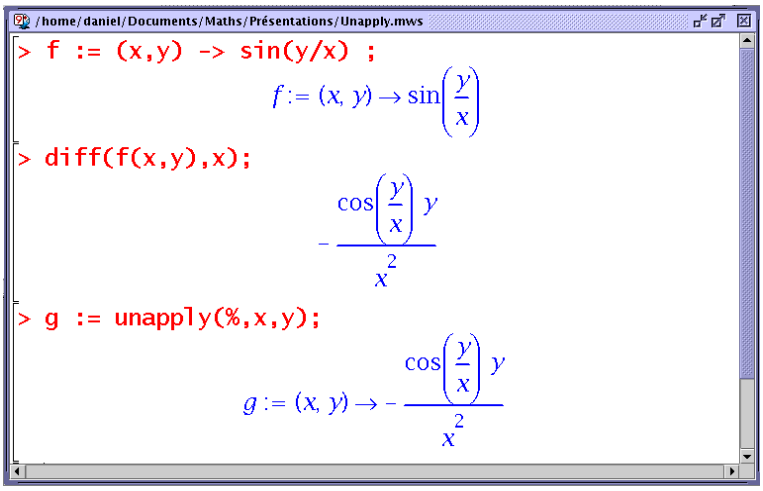


La commande fondamentale **unapply** fonctionne ainsi :

unapply(*expression* **,** *var1* { *var2* **,** ... **,** *varN* } **) ;**

4 En effet, l'opérande au dénominateur est *x*+*y*+*z*.

Par exemple :



Attention, **g := x - > %**, bien que tentant, n'aurait pas marché. En effet, dès que le symbole "**- >**" apparaît, il initie la définition d'une fonction (**g**). Dès lors, le caractère "**%**" désigne le dernier résultat calculé *dans le cours de la définition de* **g**. Il ne "sort" pas de cette dernière, et ne peut donc convenir pour faire référence à un résultat extérieur à **g**.

Opérations sur les fonctions

Maple permet d'opérer directement sur les fonctions. Résumons les correspondances avec la notation mathématique :

mathématiquement	traduction *Maple*
$f+g, f*g$	**f + g, f * g**
$f \circ g$ (composition)	**f @ g**
$f^n = \underbrace{f \circ \ldots \circ f}_{n \text{ fois "}f\text{"}}$	**f @@ g** (5)

Maple permet enfin d'utiliser des fonctions "anonymes" : **(x - > 1/(1 + x^2)) (1)** est correct. Cela peut servir à l'occasion, pour une fonction utilisée seulement ponctuellement.

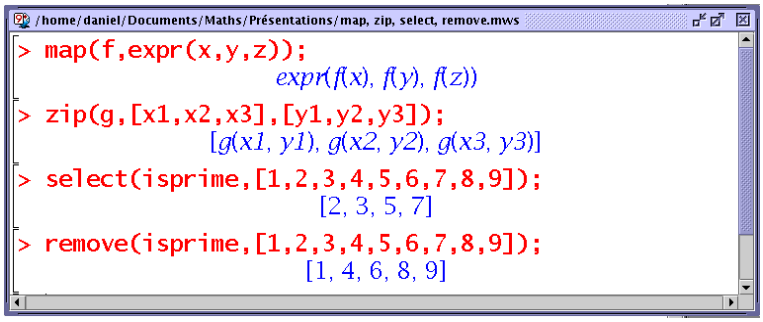
Fonctions d'ordre supérieur

Maple est assez pauvre en cette catégorie de "fonctions sur les fonctions". Citons

- **map**, qui applique une fonction à tous les opérandes d'une expression ;
- **zip**, qui s'applique à une fonction de *n* variables et *n* listes, et "enfile" la fonction à chaque $i^{\text{ème}}$ terme des *n* listes ;
- **select** et **remove**, qui filtrent une liste selon qu'un critère (booléen) est satisfait ou non.

On comprend mieux ce qui se passe avec quelques exemples :

5 Noter l'analogie avec la notation **x ** n** -- pour itérer la multiplication.



(**isprime(n)** est *true* ou *false* selon la primalité de l'entier *n*.)