Mathématiques pour l'informatique 3
(notes de cours)

L2 – Université Paris-Est Marne-la-Vallée

Cyril Nicaud

8 février 2014

CHAPITRE I

Relations d'ordre

Ce chapitre traite des relations d'ordre. Après des rappels de notions abordées l'an dernier, on s'intéresse plus particulièrement aux "ordres bien fondés" qui permettent de généraliser le principe de récurrence.

I.1 Ordre et ordre strict

Définition (relation binaire) Soit E un ensemble. Une *relation binaire* $\mathcal{R}$ sur E est un sous-ensemble de $E \times E$. On note $x\mathcal{R}y$ pour signifier que $(x, y) \in \mathcal{R}$ et $x \not\mathcal{R} y$ pour signifier que $(x, y) \notin \mathcal{R}$.

Définition (classification) Soit $\mathcal{R}$ une relation binaire sur E. On dit que $\mathcal{R}$ est

- *réflexive* quand $\forall x \in E$, $x\mathcal{R}x$;
- *irréflexive* quand $\forall x \in E$, $x \not\mathcal{R} x$;
- *symétrique* quand $\forall x, y \in E$, $x\mathcal{R}y \Rightarrow y\mathcal{R}x$;
- *antisymétrique* quand $\forall x, y \in E$, $x\mathcal{R}y$ et $y\mathcal{R}x \Rightarrow x = y$;
- *transitive* quand $\forall x, y, z \in E$, $x\mathcal{R}y$ et $y\mathcal{R}z \Rightarrow x\mathcal{R}z$.

Définition (ordre) Une relation binaire est un *ordre* (ou une *relation d'ordre*) quand elle est réflexive, antisymétrique et transitive.

Définition (ensemble ordonné) Soit E un ensemble et $\preccurlyeq$ une relation d'ordre sur E. On dit que $(E, \preccurlyeq)$ est un *ensemble ordonné*.

▶ Attention, les définitions ci-dessus correspondent à l'ordre large mais pas à l'ordre strict : $(\mathbb{R}, \prec)$ où $\prec$ est l'ordre strict usuel sur les réels n'est pas un ensemble ordonné (c'est un ensemble strictement ordonné, voir en-dessous).

Définition (ordre strict) Une relation binaire est un *ordre strict* (ou une *relation d'ordre strict*) quand elle est irréflexive et transitive.

Théorème I.1 *Un ordre strict est toujours antisymétrique.*

Définition (ensemble strictement ordonné) Soit E un ensemble et $<$ une relation d'ordre strict sur E. On dit que $(E, \prec)$ est un *ensemble strictement ordonné.*

Définition (ordre total) Un ordre $\preccurlyeq$ sur E est dit *total* si deux éléments sont toujours comparables : $\forall x, y \in E,\ x \preccurlyeq y$ ou $y \preccurlyeq x$. Un ordre qui n'est pas total est dit *partiel.*

Définition (ordre strict total) Un ordre strict $\prec$ sur E est dit *strict total* si deux éléments différents sont toujours comparables : $\forall x, y \in E,\ x \neq y \Rightarrow x \prec y$ ou $y \prec x$.

► Sans surprise, on peut toujours passer d'un ordre à un ordre strict et réciproquement. Si $(E, \preccurlyeq)$ est un ensemble ordonné, la relation $\prec$ définie par $x \prec y$ ssi ($x \neq y$ et $x \preccurlyeq y$) est un ordre strict. Réciproquement, si $(E, \prec)$ est un ensemble strictement ordonné, la relation $\preccurlyeq$ définie par $x \preccurlyeq y$ ssi ($x = y$ ou $x \prec y$) est une relation d'ordre.

I.2 Minorants, majorants, minimaux et maximaux

Définition (minorant, plus petit élément) Soit $(E, \preccurlyeq)$ un ensemble ordonné et F une partie (c'est-à-dire un sous-ensemble) non vide de E. On dit que $x \in E$ est un *minorant* de F si tout élément de F est plus grand que x pour $\preccurlyeq$: $\forall y \in F,\ x \preccurlyeq y$. Si le minorant de F est un élément de F on dit que c'est *le plus petit élément* de F.

Définition (majorant, plus grand élément) Soit $(E, \preccurlyeq)$ un ensemble ordonné et F une partie non vide de E. On dit que $x \in E$ est un *majorant* de F si tout élément de F est plus petit que x pour $\preccurlyeq$: $\forall y \in F,\ y \preccurlyeq x$. Si le majorant de F est un élément de F on dit que c'est *le plus grand élément* de F.

► Il n'y a pas toujours un minorant ou un majorant : si l'ensemble c'est $\mathbb{Z}$ et la partie $\mathbb{P}$ l'ensemble des entiers relatifs divisibles par 2, il n'y a ni minorant ni majorant.

Théorème I.2 *Soit $(E, \preccurlyeq)$ un ensemble ordonné et F une partie non vide de E. F a au plus un plus petit élément et au plus un plus grand élément.*

Définition (élément minimal) Soit $(E, \preccurlyeq)$ un ensemble ordonné et F une partie non vide de E. Un élément $x \in F$ est un *élément minimal* de F quand aucun élément de F n'est strictement plus petit, pour $\preccurlyeq$, que x : $\forall y \in F,\ y \preccurlyeq x \Rightarrow y = x$.

Définition (élément maximal) Soit $(E, \preccurlyeq)$ un ensemble ordonné et F une partie non vide de E. Un élément $x \in F$ est un *élément maximal* de F quand aucun élément de F n'est strictement plus grand, pour $\preccurlyeq$, que x : $\forall y \in F,\ x \preccurlyeq y \Rightarrow y = x$.

I.3 Produit d'ordres

Définition (ordre produit simple) Soit $(E, \preccurlyeq_E)$ et $(F, \preccurlyeq_F)$ deux ensembles ordonnés. On définit l'*ordre produit simple* $\preccurlyeq_{\text{prod}}$ sur $E \times F$ par

$$\forall (x,y) \in E \times F, \forall (x',y') \in E \times F,\ (x,y) \preccurlyeq_{\text{prod}} (x',y') \Leftrightarrow x \preccurlyeq_E x' \text{ et } y \preccurlyeq_F y'.$$

Définition (ordre produit lexicographique) Soit $(E, \preccurlyeq_E)$ et $(F, \preccurlyeq_F)$ deux ensembles ordonnés. On définit l'*ordre produit lexicographique* $\preccurlyeq_{\text{prod}}$ sur $E \times F$ par, on notant $\prec_E$ et $\prec_F$ les ordres stricts associés à $\preccurlyeq_E$ et $\preccurlyeq_F$,

$$\forall (x,y) \in E \times F, \forall (x',y') \in E \times F,\ (x,y) \preccurlyeq_{\text{lex}} (x',y') \Leftrightarrow \begin{cases} x \prec_E x' \\ \text{ou} \\ x = x' \text{ et } y \preccurlyeq_F y'. \end{cases}$$

► Si $\preccurlyeq_E$ et $\preccurlyeq_F$ sont des ordres totaux, $\preccurlyeq_{\text{lex}}$ est aussi un ordre total ; ce n'est pas vrai pour $\preccurlyeq_{\text{prod}}$.

I.4 Ordre bien fondé

Définition (ordre bien fondé) Soit $(E, \preccurlyeq)$ un ensemble ordonné. On dit que $\preccurlyeq$ est *bien fondé* quand il n'existe pas de suite infinie strictement décroissante d'éléments de E.

► L'ordre usuel sur $\mathbb{N}$ est bien fondé, mais pas celui sur $\mathbb{Z}$, ni celui sur $\mathbb{R}^+$.

Théorème I.3 (caractérisation des ordres bien fondés) *Soit $(E, \preccurlyeq)$ un ensemble ordonné. L'ordre $\preccurlyeq$ est bien fondé si et seulement si tout partie non vide F de E admet au moins un élément minimal.*

Théorème I.4 *Le produit lexicographique de deux ordres bien fondés est bien fondé.*

► Le programme suivant se termine pour toutes entrées n et m dans $\mathbb{N}^*$:

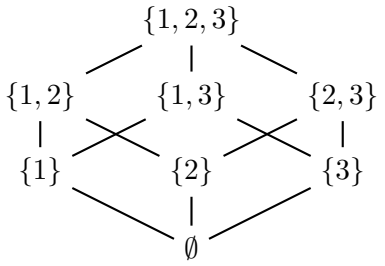
```
while (m != n) {
  if (m > n)
    m = m - n;
  else
    n = n - m;
}
```

En effet, si on considère l'ordre lexicographique naturel sur $\mathbb{N}^* \times \mathbb{N}^*$, on s'aperçoit qu'à chaque itération le couple (m, n) est strictement inférieur à celui d'avant. Comme l'ordre lexicographique sur $\mathbb{N}^* \times \mathbb{N}^*$ est bien fondé (d'après le théorème précédent), on ne peut effectuer qu'un nombre fini d'itérations dans le programme.

I.5 Ordres sur des ensembles finis, diagramme de Hasse

Quand $(E, \preccurlyeq)$ est un ensemble fini ordonné, on peut le représenter graphiquement par son *diagramme de Hasse*, qui est un graphe non-orienté avec les contraintes suivantes :
- les sommets sont les éléments de E ;
- si $x \prec y$ on place x plus bas que y sur le diagramme ;
- si $x \prec y$ et qu'il n'y a pas de $z \in E$ tel que $x \preccurlyeq z \preccurlyeq y$, on met une arête entre x et y.

► Le diagramme de Hasse pour l'ordre $\subset$ sur $E = \{1, 2, 3\}$ est :



Théorème I.5 *Soit $(E, \preccurlyeq)$ un ensemble fini ordonné. L'ordre $\preccurlyeq$ est bien fondé sur E.*

I.6 Ordres sur les mots

On se donne un ordre strict $\prec$ sur l'alphabet.

Définition (ordre lexicographique sur les mots) Soit A un alphabet fini. On définit l'ordre lexicographique sur les mots $\leq_{\text{lex}}$ par

$$\forall u, v \in A^*,\ u \leq_{\text{lex}} v \Leftrightarrow \begin{cases} u \text{ est préfixe de } v \\ \text{ou} \\ \exists w, s, t \in A^*,\ \exists a, b \in A, a \prec b \text{ et } u = was \text{ et } v = wbt \end{cases}$$

Définition (ordre militaire sur les mots) Soit A un alphabet fini. On définit l'ordre militaire sur les mots $\leq_{\text{mil}}$ par

$$\forall u, v \in A^*,\ u \leq_{\text{mil}} v \Leftrightarrow \begin{cases} |u| < |v| \\ \text{ou} \\ |u| = |v| \text{ et } u \leq_{\text{lex}} v \end{cases}$$

Théorème I.6 *Sur les mots, l'ordre militaire est bien fondé, mais pas l'ordre lexicographique.*

CHAPITRE II

Récurrence et induction

Définition (propriété) Une *propriété* sur un ensemble E est une application de E dans $\{\text{Vrai}, \text{Faux}\}$.

II.1 Récurrence

Théorème II.1 (principe de récurrence) *Soit P une propriété définie sur $\mathbb{N}$. Pour montrer que la propriété est toujours vraie, il suffit de montrer que*

(initialisation) la propriété est vraie en 0 : $P(0) = Vrai$;

(hérédité) pour tout $n \in \mathbb{N}$, si la propriété est vraie en n alors elle est encore vrai en $n+1$:

$$\forall n \in \mathbb{N},\ P(n) \Rightarrow P(n+1).$$

▶ Attention dans les preuves à ne pas supposer que P est vrai pour tout $n \in \mathbb{N}$ (il n'y a plus rien à démontrer si on suppose ça).

▶ Cela fonctionne aussi si on prend $\mathbb{N}_{\geq k} = \{k, k+1, \ldots\}$ l'ensemble des entiers supérieurs ou égaux à k, mais il faut faire l'initialisation en k.

▶

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}.$$

Théorème II.2 (principe de récurrence généralisée) *Soit P une propriété définie sur $\mathbb{N}$. Pour montrer que la propriété est toujours vraie, il suffit de montrer que*

(initialisation) la propriété est vraie en 0 : $P(0) = Vrai$;

(hérédité) pour tout $n \in \mathbb{N}^*$, si la propriété est vraie pour tout $k < n$ alors elle est encore vraie en n :

$$\forall n \in \mathbb{N}^*,\ (\forall k < n,\ P(k)) \Rightarrow P(n).$$

► C'est quasiment la même chose, à cause des propriétés structurelles de l'ensemble ordonné $(\mathbb{N}, \leq)$.

II.2 Deux exemples en informatique

► Si deux mots non vides commutent ($uv = vu$) alors ils sont puissance d'un même mot :

$$\exists w \in A^*,\ \exists k \geq 1,\ \exists \ell \geq 1,\ u = w^k \text{ et } v = w^\ell.$$

► Soit G un graphe orienté avec n sommets. S'il existe un chemin entre x et y alors il en existe un de longueur au plus $n-1$: par récurrence sur la longueur ℓ du chemin initial.

II.3 Induction bien fondée

Théorème II.3 (induction bien fondée) *Soit $(E, \preccurlyeq)$ un ensemble muni d'un ordre bien fondé. Soit P une propriété sur E. Si les deux hypothèses suivantes sont vérifiées :*

1. *$P(x) = Vrai$ pour tout élément minimal x de $(E, \preccurlyeq)$,*
2. *si pour tout élément non minimal x, on a la propriété :*

$$(\forall y \prec x, P(y) = Vrai) \ \Rightarrow\ P(x) = Vrai,$$

alors

$$\text{pour tout } x \in E,\ P(x) = Vrai.$$

► Soit $(E, \preccurlyeq)$ où $E = \mathbb{N} \setminus \{0, 1\} = \{2, 3, \ldots\}$ et $\preccurlyeq$ est l'ordre "divise" :

$$x \preccurlyeq y \Leftrightarrow \exists k \in \mathbb{N}^*,\ y = kx.$$

On a vu en TD que l'ordre "divise" est bien fondé et que ses éléments minimaux sur E sont les nombres premiers.

Soit P la propriété "s'écrit comme un produit de nombres premiers". On montre par induction bien fondée pour cet ordre que cette propriété est toujours vraie sur E.

CHAPITRE III

Structures Inductives

III.1 Introduction

L'objectif est de formaliser des définitions qui sont par nature récursive (on dira plutôt *inductive* dans un contexte mathématique) pour définir des objets, comme par exemple des arbres binaires qui seront définis de la façon suivante :

- $\square$ est un arbre binaire,
- si $\mathcal{A}$ et $\mathcal{B}$ sont des arbres binaires, $\overset{\square}{\underset{\mathcal{A}\ \mathcal{B}}{\wedge}}$ aussi.

Mais il faudrait rajouter un "il n'y a rien d'autre", ce qu'on va définir formellement dans la suite.

III.2 Définitions

On rappelle que si les A_i, pour $i \in \mathcal{I}$, sont des sous-ensembles d'un ensemble E, l'intersection de tous les A_i est l'ensemble

$$\bigcap_{i\in\mathcal{I}} A_i = \{x \in E \mid \forall i \in \mathcal{I},\ x \in A_i\}.$$

III.2.1 Fermeture inductive

Soit U un ensemble, on note $\mathcal{F}_k$ l'ensemble des fonctions d'arité k (c'est-à-dire des fonctions à k variables) partielles de $U^k \to U$. On note $\mathcal{F}$ toutes les fonctions partielles à une ou plusieurs variables :

$$\mathcal{F} = \bigcup_{k\geq 1} \mathcal{F}_k.$$

Définition (stabilité) Soit Ω un sous-ensemble de $\mathcal{F}$ et E un sous-ensemble de U. E est dit *stable* par Ω quand pour tout $f \in U$ d'arité k, et pour tout $(x_1, \ldots, x_k) \in E^k$, si $f(x_1, \ldots, x_k)$ existe alors $f(x_1, \ldots, x_k) \in E$.

Cela signifie donc que les images des k-uplets d'éléments de E sont dans E.

Définition Soit U un univers (un ensemble), $B \subset U$ la *base* et $\Omega \subset \mathcal{F}$ un ensemble de fonctions partielles. La *fermeture inductive* E de B par Ω est le sous-ensemble de U défini par :

$$E = \bigcap_{\substack{B \subset X \\ X \text{ stable par } \Omega}} X.$$

Théorème III.1 *La fermeture inductive de B par Ω est le plus petit sous-ensemble de U, pour l'inclusion qui vérifie*

- (base) $B \subset E$
- (induction) *E est stable par Ω.*

▶ L'ensemble $\mathcal{P}$ des mots bien parenthésés sont un sous-ensemble de $A = \{a, b\}$, où a correspond à une parenthèse ouvrante et b à une parenthèse fermante. L'univers est A^*. On a $\mathcal{P}$ défini inductivement par :

$$\begin{cases} B = \{\varepsilon\}, \\ \Omega = \{f\} \text{ avec } f(u, v) = aubv. \end{cases}$$

Théorème III.2 *Soit E la fermeture inductive de B par Ω. Un élément $x \in U$ est dans E si et seulement si :*

- *ou bien $x \in B$,*
- *ou bien il existe $f \in \Omega$ d'arité k et $x_1, \cdots, x_k \in E$ tels que $x = f(x_1, \cdots, x_k)$.*

III.2.2 Définition ascendante et descendante

La définition donnée pour la fermeture inductive est appelée *définition descendante* : E est obtenu par intersection infinie de tous les ensembles vérifiant (base) et (induction).

Soit E la fermeture inductive de B par Ω. On définit la suite des sous-ensembles de E définie par :

- $B_0 = B$,
- $B_{i+1} = \Omega(B_i) \cup B_i$, $\forall i \in \mathbb{N}$.

Théorème III.3 *On a $E = \cup_{i \geq 0} B_i$.*

▶ Cela signifie qu'on obtient les éléments de E en appliquant un nombre fini de fois des règles de Ω, en partant des éléments de la base.

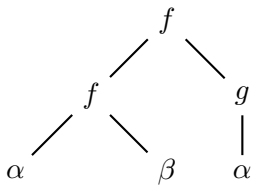
Définition (hauteur d'induction) Soit E la fermeture inductive de B par Ω. La *hauteur*, ou *hauteur d'induction*, de $x \in E$ est le plus petit entier h tel que $x \in B_h$.

III.2.3 Arbre syntaxique

Comme chaque élément x de la fermeture inductive E de B par Ω se construit en appliquant successivement les règles de construction, il est souvent utile de décrire par un arbre un procédé qui permet de construire x. Cet arbre s'appelle un *arbre syntaxique*.

On construit donc un arbre dont les feuilles sont étiquetés par des éléments de B et dont les nœuds internes sont étiquetés par des éléments de Ω : si une règle f d'arité k intervient dans le processus de construction de x, le nœud associé aura k descendants.

▶ Voilà l'arbre syntaxique de $x = f(f(\alpha, \beta), g(\alpha))$, pour $B = \{\alpha, \beta\}$ et f d'arité 2 et g d'arité 1.



▶ Il est important de remarquer qu'un x donné peut avoir plusieurs arbres syntaxiques : il peut très bien il y avoir plusieurs façons de construire x.

III.3 Principe d'induction structurelle

C'est l'équivalent de la récurrence pour les structures inductives. C'est donc un des théorèmes les plus utiles de ce chapitre.

Théorème III.4 *Soit E la fermeture inductive de B par Ω. Soit P une propriété définie sur E. Pour montrer que pour tout $x \in E$ la propriété $P(x)$ est vraie, il suffit de montrer que :*

- (base d'induction) $\forall x \in B$, $P(x) = Vrai$.
- (étape d'induction) *pour tout $f \in \Omega$ d'arité k et tout $x_1, \cdots, x_k$, si $y = f(x_1, \cdots, x_k)$ est définie, alors $P(y) = Vrai$.*

III.4 Définition de fonctions sur des ensembles inductifs

III.4.1 Schémas libres

Définition (uniquement constructible) Soit E la fermeture inductive de B par Ω. Un élément $x \in E$ est dit *uniquement constructible* quand une et une seule des deux propositions suivantes est vérifiée :

- $x \in B$,
- il existe une unique règle f, d'arité k, et un unique k-uplet $(x_1, \cdots, x_k)$ tel que $x = f(x_1, \cdots, x_k)$.

Définition (induction libre) Soit E la fermeture inductive de B par Ω. E est *libre*, ou *librement engendrée*, ou est un *schéma libre d'induction*, quand tous ses éléments sont uniquement constructibles.

Si f est une fonction d'arité k, de U^k dans U, et que X est un sous-ensemble de U, on note $f(X)$ l'ensemble des images des éléments de X :

$$f(X) = \{f(x_1, \ldots, x_k) \mid x_1, \ldots, x_k \in X \text{ et } f(x_1, \ldots, x_k) \text{ existe}\}.$$

On rappelle aussi que si f est une fonction de U^n dans U et que $E \subset U$, la restriction de f à E, notée $f_{|E}$ est la fonction de E^n dans U définie, pour tout $(x_1, \ldots, x_n) \in E^n$ du domaine de définition de f, par

$$f_{|E}\left((x_1, \ldots, x_n)\right) = f\left((x_1, \ldots, x_n)\right).$$

Théorème III.5 *Soit E la fermeture inductive de B par Ω. E est librement engendrée si et seulement si les trois points suivants sont vérifiés :*

- $\forall f \in \Omega,\ f(E) \cap B = \emptyset$.
- $\forall f \in \Omega,\ f_{|E}$ *est injective.*
- $\forall f, g \in \Omega,\ f(E) \cap g(E) = \emptyset$.

III.4.2 Fonctions

L'idée est de définir une fonction inductivement en donnant l'image des éléments de sa base et en expliquant comment transformer les règles.

Soit E la fermeture inductive **libre** de B par Ω. La définition inductive d'une application ϕ de E dans un ensemble F consiste en

- la donnée $\phi(x)$ pour tout élément $x \in B$.
- Pour toute règle $f \in \Omega$, d'arité k, la donnée de l'expression de $\phi(f(x_1, \cdots, x_k))$ en fonction des x_i et des $\phi(x_i)$.

► Si le schéma n'est pas libre il peut y avoir ambiguïté. Par exemple, pour E défini par

$$\begin{cases} \varepsilon, a, b \in E \\ f(u, v) = uv \in E \quad \forall u, v \in E \end{cases}$$

Si on définit la fonction ϕ de E dans $\mathbb{N}$ par

$$\begin{cases} \phi(\varepsilon) = \phi(a) = \phi(b) = 1 \\ \phi(f(u, v)) = \phi(u) + \phi(v) \quad \forall u, v \in E \end{cases}$$

On a $\phi(ab) = \phi(a) + \phi(b) = 2$ et $\phi(ab) = \phi(\varepsilon) + \phi(ab) = \phi(\varepsilon) + \phi(a) + \phi(b) = 3$.

III.4.3 Programmation

Les fonctions définies inductivement sont la plupart du temps transformables directement en fonction récursives. Il existe même des langages de programmation (prog. fonctionnelle) dont c'est le principe de base.

Le principe est de faire une fonction en deux parties. De la forme :

```
phi(x)
* Si x est dans B
  * retourner la valeur connue phi(x)
* Sinon
  * x = f(x1...xk)
  * retourner l'expression en fonction de x1...xk et phi(x1)..phi(xk)
```

▶ Il faut donc pouvoir "inverser" $x = f(x_1, \cdots, x_k)$, c'est à dire retrouver les x_i.

III.5 Un exemple : les expressions rationnelles

On définit inductivement l'ensemble des expressions rationnelles non vides sur l'alphabet A, noté $\mathcal{E}$ par (ce sont des mots sur l'alphabet $A \cup \{(,), \cup, \cdot, *\}$) :
- $\{a\} \in \mathcal{E}$ pour tout $a \in A$;
- $\{\varepsilon\} \in \mathcal{E}$;
- $(E_1 \cup E_2) \in \mathcal{E}$, pour tout $E_1, E_2 \in \mathcal{E}$;
- $(E_1 \cdot E_2) \in \mathcal{E}$, pour tout $E_1, E_2 \in \mathcal{E}$;
- $(E)^* \in \mathcal{E}$, pour tout $E \in \mathcal{E}$.

▶ On considère la fonction ϕ de $\mathcal{E}$ dans $\{0, 1\}$ définie par $\phi(\varepsilon) = 1$, $\phi(a) = 0$. Et $\phi(E_1 \cup E_2) = \phi(E_1) \vee \phi(E_2)$, etc. On a $\phi(E) = 1 \Leftrightarrow E$ reconnaît le mot vide. Cette fonction associe à une formule (l'expression) un langage (un ensemble de mots de A^*).

▶ On peut facilement faire des algorithmes récursifs sur l'arbre de l'expression, ce qui revient, mathématiquement, à définir des fonctions sur $\mathcal{E}$.

▶ Il y a beaucoup d'autres types d'expressions qui sont utilisées en informatique : formules logiques, expressions arithmétiques, etc. Toutes fonctionnent sur le même principe.