

i. Le contexte `signs` pour les signes

Voici tous les éléments de syntaxe propres au contexte `signs`.

1. Le contexte `signs` est employable autant de fois que nécessaire.
2. L'expression dont on indique le signe doit être donnée via `mon_expr : ...` où les points de suspension spécifient les informations liées au comportement de l'expression (voir l'item suivant à ce sujet). **Il n'existe pas d'expression par défaut.**
3. Dans cet item, pour expliquer ce qui est attendu comme informations relatives au comportement d'une expression f , nous allons supposer que n valeurs pivots x_1, x_2, \dots, x_n ont été données via `bounds = x_1 , x_2 , ... , x_n`. Nous posons aussi $p = n - 1$ avec p pour « *p-récédent le naturel n* » .
 - A minima, il faut indiquer `f : s_1 s_2 ... s_p` où `s_1, s_2, ..., s_p` donnent des informations sur les p intervalles `]x_1 ; x_2[,]x_2 ; x_3[, ... ,]x_p ; x_n[` respectivement. Les valeurs possibles pour les `s_k` sont les suivantes.
 - (a) `+` indique une expression positive stricte sur l'intervalle concerné.
 - (b) `-` indique une expression négative stricte sur l'intervalle concerné.
 - (c) `z` indique une expression nulle sur l'intervalle concerné avec `z` pour **z-éro**.
 - (d) `u` indique une expression non définie sur l'intervalle concerné¹ avec `u` pour **u-ndefined** soit « *non défini* » en anglais.
 - On peut aussi indiquer le comportement d'une expression en certaines valeurs pivots. Ceci se fait à côté d'une information de type signe : par exemple, en gardant les notations de l'item précédent, nous avons les possibilités suivantes.
 - Si $n > 3$ alors $f(x_3) = 0$ s'indique via `... 0 s_3 ...` au milieu du code.
 - Si $n = 3$ alors $f(x_3) = 0$ s'indique via `... s_2 0` en fin de code.
 - $f(x_1) = 0$ s'indique via `0 s_1 ...` au début du code.

Les valeurs possibles pour le comportement éventuel en un pivot sont les suivantes.

- (a) `0` indique que l'expression s'annule au pivot concerné.
- (b) `!` indique que l'expression n'est pas définie au pivot concerné².
- (c) L'absence de valeur indique que l'expression ne change pas de signe au pivot concerné.

Note. Les espaces autour des doubles points et des informations codées ne sont pas obligatoires.

Voici deux codes fictifs illustrant les explications précédentes ; noter au passage que les espaces ignorés permettent d'obtenir un résultat humainement très clair.

```
% f > 0 sur ]x_1 ; x_2[
% f < 0 sur ]x_2 ; x_4[
% f = 0 en x_2

bounds =      x_1 , x_2 , x_3 , x_4 ;
signs   = f :      + 0 -      -
```

```
% Plus étrange...
% f = 0          sur {x_1} U ]x_2 ; x_3[
% f non définie sur ]x_1 : x_2] U {x_3}

bounds =      x_1 , x_2 , x_3 ;
signs   = f : 0 u ! z !
```

Dans le cadre de processus automatisés, il est possible de produire les horreurs suivantes qui aboutiront aux mêmes sorties que les codes correspondants ci-dessus.

```
bounds=x_1,x_2,x_3,x_4;signs=f:+0--
```

```
bounds=x_1,x_2,x_3;signs=f:0u!z!
```

1. Penser par exemple à l'expression $x\sqrt{x^2-1}$.

2. En France, le panneau de signalisation indiquant un danger est un triangle blanc au bords rouges et contenant un point d'exclamation.