

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 signe de l'expression (voir l'item suivant).
3. Dans cet item, nous allons supposer que `bounds = x_1 , x_2 , x_3` a été utilisé pour donner les valeurs pivots `x_1` , `x_2` et `x_3` . Les cas suivants couvrent tout ce qu'il est possible de faire.
 - `f : + -` indique que $f > 0$ sur $]x_1 ; x_2[$, $f < 0$ sur $]x_2 ; x_3[$, sans donner d'information particulière sur f en `x_1` , `x_2` et `x_3`¹.
 - `f : + 0 -` complète ce qui précède en indiquant que $f = 0$ en `x_2` .
 - `f : + ! -` permet d'indiquer une valeur interdite en `x_2`².
 - `f : 0 + ! - !` est autorisé pour indiquer aussi le comportement de f en `x_1` et `x_3` .
 - `f : u z` indique f n'est pas définie sur $]x_1 ; x_2[$ et aussi que $f = 0$ sur $]x_2 ; x_3[$ ³.

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

Voici deux codes fictifs reprenant les indications précédentes. Noter que les espaces ignorés permettent d'obtenir un résultat humainement clair.

```
% f > 0 sur ]x_1 ; x_2[
% f < 0 sur ]x_2 ; x_3[
% f = 0 en x_2

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

```
% Pas d'infos sur ]x_1 ; x_2[
% f = 0 sur {x_1} U ]x_2 ; x_3[
% f non définie sur ]x_1 : x_2] et en x_3

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

1. Notons que ceci est totalement étrange.

2. Le panneau français indiquant un danger est un triangle rouge contenant un point d'exclamation.

3. Les lettres `u` et `z` sont pour `u·ndefined` et `z·ero` où « *undefined* » signifie « *non défini* » en anglais.