

i. Le contexte vars pour les variations

La logique d'utilisation et d'organisation des informations pour le contexte `vars` est similaire à celle du contexte `signs`; nous indiquons donc juste les différences.

1. Voici les informations possibles pour le comportement d'une expression f en supposant que n valeurs pivots x_1, x_2, \dots, x_n ont été données via `bounds = x_1 , x_2 , ... , x_n`. De nouveau, nous posons $p = n - 1$.

- A minima, il faut indiquer `f : v_1 v_2 ... v_p` où v_1, v_2, \dots, v_p donnent des informations sur les p intervalles ouverts $]x_1 ; x_2[,]x_2 ; x_3[, \dots ,]x_p ; x_n[$ respectivement. Les valeurs possibles pour les v_k sont les suivantes.
 - (a) `<` indique une expression strictement croissante sur l'intervalle concerné.
 - (b) `>` indique une expression strictement décroissante sur l'intervalle concerné.
 - (c) `=` indique une expression constante sur l'intervalle concerné.
 - (d) `u` indique une expression non définie sur l'intervalle concerné (comme pour les signes).
- Pour des valeurs pivots précises, on peut indiquer des images ou des limites à droite et/ou à gauche. Voici ce qui est disponible.
 - (a) L'absence d'expression est possible pour ne rien indiquer du tout.
 - (b) Toute expression sans lettre `u`, ni ponctuation `!` est interprétée comme une valeur image au format mathématique `LATEX`.
 - (c) Une valeur image contenant `u` et/ou la ponctuation `!` en tant que « *token L^AT_EX* » devra être protégée par des accolades. Ceci vient du fait que dans le langage codant les variations ces deux caractères ont une signification spéciale (voir ci-dessus et l'item suivant).
 - (d) `!` indique que l'expression n'est pas définie au pivot concerné. On peut aussi, si besoin, indiquer des limites à gauche et/ou à droite. Voici les cas possibles.
 - `!` utilisé seul n'indique aucune limite.
 - `! d` indique juste d comme limite à droite.
 - `g !` indique juste g comme limite à gauche.
 - `g ! d` indique g et d comme limites à gauche et à droite respectivement.

Cette syntaxe est interdite pour le tout dernier pivot.

Cette syntaxe est interdite pour le tout premier pivot.

Cette syntaxe est interdite pour les premier et dernier pivots.

Note. Les espaces autour des doubles points, du point d'exclamation 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 croît de 5 à 7 sur ]x_1 ; x_2[
% f est constante sur [x_2 ; x_3]
% f décroît de 7 à 0 sur ]x_3 ; x_4[

% 5 = valeur en x_1
% 7 = valeur en x_2 et x_3
% 0 = valeur en x_4

bounds = x_1 , x_2 , x_3 , x_4 ;
vars = f : 5 < 7 = 7 > 0
```

```
% f croît de 5 à 7 sur ]x_1 ; x_2[
% f non définie sur [x_2 : x_3]
% f constante sur ]x_3 ; x_4[

% 5 = valeur en x_1
% 7 = lim à gauche en x_2
% 9 = lim à droite en x_3

bounds = x_1 , x_2 , x_3 , x_4 ;
vars = f : 5 < 7! u !9 =
```

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;vars=f:5<7=7>0
```

```
bounds=x_1,x_2,x_3,x_4;vars=f:5<7!u!9=
```