

Opérateurs de comparaison ou d'unification en Swi-Prolog

1. On n'a pas de calculs à effectuer

On utilise l'opérateur d'unification `=` (p. ex., `R=4` ou `R=c` ou `R=S` ou `R=[a,b,c]`).

Négation : `\=`

2. On a besoin de faire des calculs

2.1 On veut affecter un calcul à une variable non instanciée

On utilise l'opérateur **is** avec la variable non instanciée à gauche ; tout doit être instancié à droite (p. ex., `R is T+2` mais il faut que `T` ait une valeur au moment où on fait l'opération).

Négation : `not(... is ...)`

2.2 On veut comparer deux calculs

On utilise l'opérateur `==` mais les deux côtés doivent être complètement instanciés. Par exemple `R+4 == S*2` mais il faut que `R` et `S` aient une valeur au moment de l'opération.

On ne peut donc pas utiliser cet opérateur pour instancier une variable.

Négation : `=\=`

3. Exemples

?- `R=4.`

`R = 4.`

?- `R=S.`

`R = S.`

?- `R=S+1.`

`R = S+1.` *réussit mais le calcul n'est pas fait !*

?- `R is S+1.`

ERROR: Arguments are not sufficiently instantiated

?- `S=4, R is S+1.`

`S = 4,`

`R = 5.`

?- `S=4, R+2 == S+1.`

ERROR: Arguments are not sufficiently instantiated

?- `S=4, R=3, R+2 == S+1.`

`S = 4,`

`R = 3.`