

1 Construction d'une calculatrice rudimentaire à 8 bits

Spécification en B d'une machine à calculer rudimentaire qui manipule des valeurs entières stockables sur 8 bits. La machine à calculer dispose de deux registres : un registre principal nommé RP et un registre secondaire nommé RS. La machine propose les opérations suivantes.

- `inc1` : incrémentation de son registre principal de 1.
- `dec1` : décrémentation de son registre principal de 1.
- `cmp` : comparaison de deux valeurs, l'une dans RP l'autre dans RS ; l'opération `cmp` renvoie un résultat qui est 1 ou 0 (valeurs égales, ou valeurs différentes).
- `storeRP(val)` : stockage d'une valeur `val` donnée dans le registre RP.
- `getRP` : récupération de la valeur stockée dans RP (prévoir une variable de sortie).

Après avoir analysé l'énoncé, écrivez la spécification en B de la machine à calculer ainsi décrite.

2 Gestion de position dans le plan (robot)

Soit à contrôler l'évolution des valeurs de deux variables `xx`, `yy`. Les valeurs sont entières. Elles peuvent représenter les coordonnées (x,y) d'une pointe traçante.

Chacune des deux variables évolue sur une plage δ à partir de sa valeur initiale ; par exemple les valeurs possibles de `xx` sont entre x_0 et $x_0 + \delta_x$; les valeurs possibles de `yy` sont entre y_0 et $y_0 + \delta_y$.



On veut écrire en B une machine abstraite `GestPosition` qui va proposer ses opérations à un programme de contrôle de la pointe traçante.

La machine abstraite `GestPosition` fournira les opérations suivantes :

- Changement de la valeur de `xx` ; on donne une nouvelle valeur `nx` en paramètre
- Changement de la valeur de `yy` ; on donne une nouvelle valeur `ny` en paramètre
- Incrémentation de la valeur de `xx` de `px` (un paramètre) ;
- Incrémentation de la valeur de `yy` de `py` ;
- Décrémentation de la valeur de `xx` de `px` ;
- Décrémentation de la valeur de `yy` de `py` ;
- Récupération de la valeur de `xx` ;
- Récupération de la valeur de `yy` ;
- Test si une valeur donnée `ux` est entre les bornes de `xx` ;
- Test si une valeur donnée `uy` est entre les bornes de `yy` ;
- Test si les valeurs actuelles de `xx` et `yy` sont dans les bornes.

Spécifiez une machine abstraite B (sans les opérations) pour décrire l'espace d'états correspondant à l'énoncé.

Complétez la machine abstraite par la spécification des opérations qui sont décrites.

3 Contrôle d'un aéroplane (*Aircraft*) - propriétés de cohérence

On considère un système d'aide au pilotage d'un aéroplane. On traitera une version simplifiée de contrôle des déplacements dans un plan borné. Pour la version simplifiée, on considère trois modes de déplacement. Le mode décollage (takeof), le mode déplacement de croisière et le mode atterrissage (landing). Pour chacun des trois modes, on définit une plage de coordonnées (x, y) .

L'aéroplane se déplace dans le plan et selon les modes ainsi définis. Des déplacements avec des coordonnées (x, y) sont ordonnés par un opérateur.

Le but du système d'aide au pilotage (ou système de contrôle) est de maintenir l'aéroplane dans des coordonnées prévues selon les modes ; ainsi seuls les déplacements avec des coordonnées cohérentes sont effectués.

Modéliser ce système avec les propriétés requises en Event-B.

4 Lecteurs et rédacteurs - garantie d'exclusion mutuelle

Modélisation d'un système réparti comprenant une ressource et des processus rédacteurs et lecteurs qui utilisent la ressource. Les rédacteurs mettent à jour la ressource. Les lecteurs consultent mais ne modifient pas la ressource.

Les exigences du système sont comme suit :

- pour la cohérence de la ressource, seul un rédacteur peut y accéder à la fois. Les rédacteurs s'excluent entre eux.
- on ne peut pas avoir en même temps des lecteurs et des rédacteurs qui accèdent à la ressource
- on autorise plusieurs lecteurs à accéder à la ressource simultanément.
- chaque processus effectue une demande de la ressource ; soit il est autorisé et à utiliser la ressource et il finit par la libérer, soit il est mis en attente jusqu'à ce que la ressource soit disponible.

Modéliser en Event-B ce système, en respectant les exigences indiquées.

```
MACHINE
  CtrlProcessus
SETS
  ...
CONSTANTS
  ...
PROPERTIES
  ...
VARIABLES
  ...
INVARIANT
  ...
INITIALISATION
  ...
END
```

petit mémo du langage de B

#	et !	il existe et quel que soit
$x : T$		x appartient à l'ensemble T
$E <: G$		E inclus dans G
\neq		négation de $=$
\setminus		union
\cap		intersection
$a \mapsto b$		couple (a,b)
$r : E \leftrightarrow G$		relation de E vers G
$f : E \mapsto G$		fonction partielle
$f : E \twoheadrightarrow G$		fonction totale
$f : E \hookrightarrow G$		injection
$f : E \twoheadrightarrow G$		surjection totale
$S \restriction r$		restriction du domaine de r par S
$S \text{ antirestriction } r$		antirestriction de domaine