Spécifications formelles avec B : TP2

1 Un projet minimum

Nous vous demandons de spécifier et d'implémenter une machine qui calcule le minimum de deux entiers naturels. Commencez par créer un repertoire `Minimum` avec les sous repertoires `bdp`, `spec` et `trad`.

Lancez l'atelier B et créez un projet (Bouton `Attach`) en indiquant respectivement `Minimum/bdp` et `Minimum/trad` comme repertoires de Base de Donnée et de Traduction. Avant d'ouvrir le projet ainsi créé, attachez lui la librairie standard `LIBRARY` (`Libraries/Add`) pour qu'il puisse accéder à l'implémentation de la machine `BASIC_IO`.

Exercice 1 *Dans le repertoire* `Minimum/spec`*, lancez* `emacs` *et créez un fichier* `CalMin.mch`*, dans lequel vous spécifierez une machine B qui fournie une fonction* `Minimum`*. Ajoutez ce composant au projet* `Minimum` *(* `Components/Add` *).*

Vous pouvez compléter le code suivant :

```
MACHINE CalMin

OPERATIONS
bb <-- Minimum(mm,nn) = PRE
    ???
  THEN
    bb:(???)
  END
END
```

Exercice 2 *Testez (et corrigez si necessaire) la syntaxe de* `CalMin`*. (* `Type Check` *).*

Exercice 3 *Générez les obligations de preuves (* `PO Generate` *).*
Si nécessaire prouvez les (`Prove/Automatic (force 0)` *puis* `BO Check` *).*

Exercice 4 *Dans un fichier que vous appellerez* `CalMin_Imp.imp`*, implémentez la machine* `CalMin`*. Ajoutez ce composant au projet* `Minimum` *et prouvez le.*

Vous pouvez compléter le code ci-dessous :

```
IMPLEMENTATION CalMin_Imp
REFINES CalMin

OPERATIONS

bb <-- Minimum(mm,nn) =
  BEGIN
    ???
  END
END
```

Exercice 5 *Dans le repertoire* `/users/linfg/strecker/Pub/Algo/TP2` *vous trouverez les trois fichiers* `Minimum_Int.mch` `Minimum_Int_Imp.imp` *et* `BASIC_IO.mch`. *Copiez les dans votre propre repertoire* `Minimum/spec` *(*`man cp`*), ajoutez les au projet et prouvez les.*

Exercice 6 *Generez du code* `c` **natif** *à partir du composant* `Minimum_Int_Imp` *(* `Translator` *). Compilez (commande* `make`*) et executez ce code (repertoire* `Minimum/trad/c`*).*

2 Addition bornée

Créez un projet `Addition` et une machine `ZCent`. On consière ici des calculs limités à l'ensemble **0..100** des entiers entre zéro et cent.

Exercice 7 *Spécifiez et implementez une fonction Add qui effectue l'addition de deux entiers entre zéro et cent et qui retourne un entier entre zéro et cent.*

Exercice 8 *Spécifiez et implementez une fonction Mul2 qui effectue la multiplication par deux d'un entier entre zéro et 100 et qui retourne un entier entre zéro et cent.*

3 Un exemple de projet B

Exercice 9 *Ouvrez le projet* `DAB`*, affichez graphe de dépendance du projet et son status. Combien reste t-il d'obligations de preuve non démontrées ? Executez le code* `c` *généré.*