

MGL7160: Exercice #4

6 février 2001

Le Nim est un jeu au cours duquel deux joueurs s'affrontent de la façon suivante: un certain nombre de bâtons (généralement 13) sont placés sur une table et chaque joueur doit, à tour de rôle, prendre *au moins un* bâton mais sans en prendre plus qu'un certain nombre (généralement trois). Celui qui prend le dernier bâton *perd la partie*.

Vous devez spécifier une **MACHINE** permettant de superviser une partie de Nim entre deux joueurs. Votre module devra permettre de généraliser le jeu pour des parties avec un nombre quelconque de bâtons (paramètre **nb_batons_depart**, non-nul, spécifié lorsqu'une nouvelle partie est amorcée à l'aide du message **debuter_partie**) ainsi que pour un nombre maximum de bâtons pouvant être pris à chaque tour (argument *générique* **prise_max**, strictement supérieur à 1).

Votre module (générique) devra exporter les opérations suivantes:

- **debuter_partie**: ce message permet de débiter une nouvelle partie. Ce message peut être appelé peu importe que la partie précédente soit terminée ou non. C'est dans ce message que le nombre initial de bâtons pour la partie (évidemment non-nul) doit être indiqué.
- **nb_batons_restant**: ce message indique le nombre de bâtons qui restent pour la partie en cours. Retourne 0 si la partie est terminée ou si aucune partie n'a encore été commencée.
- **prochain_joueur**: ce message indique quel doit être le prochain joueur à jouer pour la partie en cours. On suppose que les deux joueurs sont identifiés à l'aide du type défini dans le module suivant:

```
DEFINITION joueurs
  CONCEPT joueur: type
  WHERE
    joueur = enumeration{joueur1, joueur2}
END
```

On suppose que c'est le joueur **#joueur1** qui joue le premier coup dans une nouvelle partie. Une exception est signalée si ce message est appelé alors que la partie précédente est terminée et qu'une nouvelle partie n'a pas encore été amorcée.

- **gagnant**: ce message indique, une fois la partie terminée, le joueur qui a gagné la partie. Si ce message est appelé alors que la partie n'est pas encore terminée, une exception est signalée.
- **jouer_un_coup**: ce message permet au joueur dont c'est le tour de jouer (tel qu'identifié par le message **prochain_joueur**) d'indiquer le nombre de bâtons qu'il désire retirer (nécessairement plus grand ou égal à 1). Lorsque le coup est valide, le message retourne aussi (avec un **REPLY**) le nombre de bâtons qui restent après le coup du joueur. Une exception est signalée si le nombre indiqué n'est pas valide. Une exception est aussi signalée lorsqu'on tente de jouer un coup alors que la partie est terminée et qu'une nouvelle partie n'a pas été amorcée.
- **partie_terminee**: ce message indique si la partie en cours est terminée ou non.