

Exercice (en classe)

20 mars 2001

Processus concurrents et communication par l'intermédiaire de canaux

Soit le processus **Trier**, dont l'interface et le comportement sont présentés plus bas. Ce processus reçoit en entrée (port **in**) une suite de clés (chaînes de caractères) et de nombres point-flottants (chaque clé est associée à un unique nombre) et produit en sortie (port **out**) la version triée de cette suite (triée en fonction de la clé). Les ports **fin_in** et **fin_out** servent à indiquer la fin de chacune de ces suites:

```
PROCESS Trier[in, fin_in, out, fin_out](items: sequence{item}): NOEXIT :=
  in ?n: string ?v: float;
  Trier[in, fin_in, out, fin_out](add([nom:: n, val:: v], items))
[]
fin_in;
Envoyer[in, fin_in, out, fin_out](items)
WHERE
PROCESS Envoyer[in, fin_in, out, fin_out](items: sequence{item}): NOEXIT :=
  [items = []] ->
    fin_out;
    Trier[in, fin_in, out, fin_out]([])
  []
  [items ~= []] ->
    LET i = index_du_min(items) IN
      out !items[i].nom !items[i].val;
      Envoyer[in, fin_in, out, fin_out](items[1..i-1] || items[i+1..length(items)])
ENDPROC

CONCEPT index_du_min( s: sequence{item} SUCH THAT s ~= [] ) VALUE( i: nat )
WHERE
  i IN domain(s),
  items[i].nom = MINIMUM( it: item SUCH THAT it IN s :: it.nom )

CONCEPT item: type
WHERE item = tuple{nom:: string, val:: float}
ENDPROC
```

En utilisant ce processus, écrivez une spécification LOTOS (un module **SPECIFICATION**, appelé **TraiterFeuillesTemps**, définissant un groupe de processus) permettant de résoudre le problème suivant:

- On reçoit en entrée (sur le port **entree**) une série de valeurs de la forme suivante: une chaîne de caractères indique le nom d'un employé, laquelle chaîne est suivie du nombre d'heures travaillées pour chacune des journées de la semaine (un nombre peut être 0.0). Les noms ne sont pas nécessairement en ordre. De plus, il est possible que plusieurs blocs d'informations soient présents dans la suite pour un employé donné, et ce à des positions *non-consécutives* (voir l'exemple plus bas). La fin de la suite est indiquée par l'envoi d'un signal **fin_entree**.
- On produit sur le port **sortie** une liste *ordonnée* des employés avec le nombre *total* d'heures travaillées durant la période traitée. La fin de la liste est indiquée par un signal **fin_sortie**.

Entrée:

```
EMP2 8.0 8.0 8.0 8.0 8.0
EMP1 7.0 7.0 8.0 9.0 0.0
EMP3 7.0 7.0 8.0 9.0 0.0
EMP1 0.0 0.0 6.0 10.0 6.0
EMP2 8.0 8.0 8.0 8.0 8.0
```

Sortie:

```
EMP1 53.0
EMP2 80.0
EMP3 31.0
```

Exemple