L246

SESSION 2002

ÉPREUVE D'INFORMATIQUE

Filière : 2ème concours

DURÉE : 3 heures

L'usage des calculatrices électroniques de poche à alimentation autonome sans imprimante et sans document d'accompagnement est autorisé. Cependant, une seule calculatrice à la fois est admise sur la table. Aucun échange n'est autorisé entre les candidats.

Tournez la page S.V.P.

1 Première partie : Réseaux de tri

Cette partie est consacrée à l'étude de circuits constitués uniquement de *comparateurs-échangeurs*, qui sont des composants prenant deux nombres en entrée et renvoyant le plus petit sur la première sortie et le plus grand sur la seconde sortie (voir Figure 1). Par convention, le "min" sortira tou-

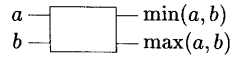

FIG. 1 – Un comparateur-échangeur

jours par la borne du haut et le "max" par la borne du bas.

Les circuits que nous allons construire sont constitués de comparateurs-échangeurs (que nous appellerons simplement *comparateurs* dans ce qui suit) reliés entre eux par des fils qui ne font que transmettre l'information. On adopte la règle selon laquelle les fils peuvent se croiser, mais aucun fil ne peut "revenir en arrière", autrement dit on doit dessiner les circuits de telle manière que l'information aille toujours de la gauche vers la droite dans un circuit (les entrées–en nombre pair–étant situées à gauche). Ainsi, le circuit de la Figure 2 est correct, et est équivalent du point de vue du calcul qui est effectué à celui de la Figure 1.

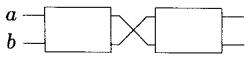

FIG. 2 – Un circuit constitué de deux comparateurs

La construction de tels circuits permet d'introduire du parallélisme dans les calculs (dès lors qu'il y a plus de deux entrées), ce qui peut accélérer ceux-ci. Avec les conventions de dessin que l'on a fixées, on peut définir une notion de *nombre de couches* d'un circuit : en s'efforçant de représenter au maximum les comparateurs les uns au-dessus des autres, le nombre de couches d'un circuit est le nombre minimal de "colonnes" de comparateurs intervenant dans sa représentation graphique. Ainsi, le circuit de la Figure 2 a deux couches, car il n'est pas possible de superposer ses deux comparateurs.

Question 1 Quelle quantité est mesurée par le nombre de couches d'un circuit ?

Étant donné un circuit, que peut-on dire sur les sorties, pour un ensemble d'entrées donné ?

On souhaite trier des listes de nombres dont la longueur est égale à une puissance de 2. On peut remarquer pour commencer que l'on a déjà ce qu'il faut pour trier des listes de deux nombres. Pour des listes plus grandes, on effectue des tris par *fusion* (la fusion de deux listes triées renvoie une liste triée constituée de tous les éléments fournis en entrée). On commence donc par concevoir des circuits calculant la fusion de deux listes de nombres triées et de longueur 2^k, pour $k \geq 1$ (le tri en lui-même sera traité à la question **5**).

Question 2 On commence par un circuit avec quatre entrées a_1, b_1, a_2, b_2 et quatre sorties c_1, c_2, c_3, c_4 tel que si l'on a $a_1 \leq a_2$ et $b_1 \leq b_2$, alors $c_1 \leq c_2 \leq c_3 \leq c_4$. En d'autres termes, ce circuit devra effectuer la fusion de deux listes triées de taille 2.

Dessinez un tel circuit, et justifiez-en soigneusement la correction.

Question 3 Passons maintenant à la fusion de deux listes triées de taille 4 chacune. On suppose que les listes $a_1, \ldots, a_4$ et $b_1, \ldots, b_4$ sont triées, et l'on veut en réaliser la fusion (i.e. récupérer une liste $c_1, \ldots, c_8$ triée en sortie).

Proposez un circuit en trois couches permettant d'effectuer ce calcul. Il est recommandé de respecter l'indication suivante : la fusion se fera en commençant par fusionner les listes constituées des éléments d'indice impair (a_1, a_3, b_1 et b_3) d'une part, et d'indice pair (a_2, a_4, b_2 et b_4) d'autre part.

Question 4.1 Définissez maintenant un circuit calculant la fusion de deux listes triées de taille 2^n en une liste de taille 2^{n+1}.

Question 4.2 (plus difficile) Justifiez rigoureusement la construction proposée en **4.1**.

Question 5 (cette question peut être traitée indépendamment des questions précédentes) Une fois que l'on a pour tout $k > 1$ un circuit (que l'on nommera F_k) calculant la fusion de deux listes de taille 2^k, indiquer comment définir un circuit qui effectue le tri d'une liste de taille 2^n, pour $n > 0$ donné.

Question 6 On note n_k le nombre de comparateurs d'un circuit effectuant le tri de 2^k nombres et c_k le nombre de couches de ce circuit. On peut montrer qu'il existe des circuits vérifiant

$$n_k = \mathcal{O}(2^k k^2) \qquad \text{et} \qquad c_k = \mathcal{O}(k^2)\,.$$

Commenter, sachant que le temps nécessaire au tri de n nombres sur une machine "classique" (sans utiliser le parallélisme) croît en $\mathcal{O}(n \log(n))$.

La notation $\mathcal{O}$ est définie comme suit : étant donnée une fonction f de $\mathbb{N}$ dans $\mathbb{R}^+$, on note :

$$\mathcal{O}(f) \;=\; \{g : \mathbb{N} \to \mathbb{R}^+ \mid \exists c \in \mathbb{R}^{+\star}.\, \exists n_0 \in \mathbb{N} \text{ tels que } \forall n > n_0.\, g(n) \leq c f(n)\}\,.$$

Tournez la page S.V.P.

2 Seconde partie : Calcul de sites

Avertissement. Cette partie comprend une longue section introductive qui présente les notions manipulées. Celle-ci est à lire avec soin avant de commencer à traiter les questions, de manière à se familiariser avec le formalisme.

On s'intéresse ici à un modèle formel basé sur la notion de *site*. On suppose pour cela que l'on dispose d'un ensemble de *noms*, que l'on notera avec des lettres minuscules $a, b, c, \ldots$. Nous commençons par introduire un certain nombre de définitions.

2.1 Définitions

Sites et sous-sites. Un site a deux composantes : son *nom* et son *contenu* ; on note

$$a[\,E\,]$$

le site de nom a et de contenu E. Un site peut comporter un ou plusieurs *sous-sites*, qui sont des sites situés dans son contenu. Ainsi,

$$a[\,b[\,E\,],\ e[\,f[\,E'\,],\ g[\,E''\,]\,]\,]$$

décrit un site a contenant deux sous-sites b et e, les sites f et g étant eux-mêmes sous-sites de e (le contenu des sites b, f et g n'est pas spécifié).

Particules. Outre des sous-sites, un site peut contenir des *particules*, de la forme in a, out a ou open a, a étant un nom quelconque (nous reviendrons plus loin sur le rôle des particules).

Multi-ensembles. Le contenu d'un site est un *multi-ensemble* de sous-sites et de particules. Un multi-ensemble est une collection d'entités dans laquelle chaque entité peut apparaître plusieurs fois (mais où l'ordre dans lequel on écrit les entités n'a pas d'importance). On notera les éléments d'un multi-ensemble en les séparant par des virgules, et on écrira e, E pour désigner le multi-ensemble obtenu en ajoutant l'élément e au multi-ensemble E. Ainsi, les multi-ensembles

$$\text{in } a,\ \text{out } c,\ \text{in } a \qquad \text{et} \qquad \text{out } c,\ \text{in } a,\ \text{in } a$$

sont identifiés, et peuvent s'écrire par exemple in a, E avec $E = \text{in } a,\ \text{out } c$, mais sont en revanche distincts du multi-ensemble in a, out c. Le multi-ensemble vide sera noté $\emptyset$, ou ne sera pas noté du tout s'il s'agit du contenu d'un site (autrement dit on écrira $a[\,]$ pour le site a dont le contenu est vide).

Termes. On appellera *terme* un multi-ensemble de sites et de particules. Par définition, le contenu d'un site est un terme, et un site (vu comme le multi-ensemble singleton correspondant) est aussi un terme. Deux termes seront identifiés s'ils ne diffèrent que par utilisation des propriétés de multi-ensemble (réarrangement des termes au sein d'un multi-ensemble : $a[\,\text{in } c,\ b[\,d[\,],\ e[\,]\,]\,]$ et $a[\,b[\,e[\,],\ d[\,]\,],\ \text{in } c\,]$ désignent ainsi le même terme). Dans le cas contraire, on dira que les termes sont *distincts*.

Voici deux exemples de termes, S_1 et S_2 :

$$S_1 = a[\,\text{in } b,\ \text{out } c,\ b[\,\text{out } b,\ \text{in } c\,],\ c[\,]\,],\ a[\,]$$
$$S_2 = b[\,d[\,],\ e[\,],\ f[\,e[\,],\ h[\,]\,],\ h[\,\text{in } a,\ \text{open } t\,]\,]$$

Remarque : Nous utilisons E, E' et E'' pour désigner des multi-ensembles de manière générique (ce sont en quelque sorte des "variables de multi-ensembles"). Les termes qui seront manipulés pour répondre aux questions **1** à **3.3** ci-dessous seront en revanche complètement spécifiés, et ne comporteront pas de telles notations, qui seront réservées à l'énoncé de propriétés générales des termes.

Évolution des termes. Ayant défini la notion de terme, nous décrivons à présent comment les termes peuvent évoluer. Les évolutions d'un terme correspondent à des mouvements (particules d'entrée et de sortie) ou à des ouvertures. Elles sont notées à l'aide du symbole $\longrightarrow$, et sont définies par les cinq règles suivantes :

· **règle R1 :** pour tous noms a et b et pour tous termes E et E' (éventuellement vides), un terme de la forme $a[\,\text{in } b,\ E\,],\ b[\,E'\,]$ peut évoluer vers le terme $b[\,a[\,E\,],\ E'\,]$ (exécution d'une instruction

in) ; on note ceci

$$a[\,\mathsf{in}\ b,\ E\,],\ b[\,E'\,] \longrightarrow b[\,a[\,E\,],\ E'\,]$$

(on remarque en passant que la particule in b est effacée au cours de l'évolution)

· **règle R2 :** pour tous noms a et b et pour tous termes E et E', un terme de la forme $a[\,b[\,\mathsf{out}\ a,\ E'\,],\ E\,]$ peut évoluer vers le terme $a[\,E\,],\ b[\,E'\,]$ (exécution d'une instruction out) ; on note ceci

$$a[\,b[\,\mathsf{out}\ a,\ E'\,],\ E\,] \longrightarrow a[\,E\,],\ b[\,E'\,]$$

· **règle R3 :** pour tout nom a et pour tout terme E, un terme de la forme open a, $a[\,E\,]$ peut évoluer vers le terme E (exécution d'une instruction open) ; on note ceci

$$\mathsf{open}\ a,\ a[\,E\,] \longrightarrow E$$

· **règle R4 :** pour tous termes E et E', si $E \longrightarrow E'$, alors pour tout terme F, on a $E,\ F \longrightarrow E',\ F$;

· **règle R5 :** pour tous termes E et E', si $E \longrightarrow E'$, alors pour tout nom a, on a $a[\,E\,] \longrightarrow a[\,E'\,]$

2.2 Questions

Question 1.1 Quelle est la règle parmi les règles **R1** à **R3** intervenant dans l'évolution suivante :

$$a[\,\mathsf{open}\ b,\ c[\,\mathsf{out}\ a\,],\ b[\,\mathsf{open}\ c\,],\ b[\,\mathsf{in}\ c\,]\,] \longrightarrow a[\,c[\,\mathsf{out}\ a\,],\ \mathsf{open}\ c,\ b[\,\mathsf{in}\ c\,]\,]\ ?$$

Justifier rigoureusement cette évolution, en indiquant comment les règles ci-dessus sont appliquées.

On notera E_1 le terme à gauche de la flèche dans le cas traité ci-dessus. Similairement, montrer que $E_2 = a[\,h[\,\mathsf{in}\ e,\ \mathsf{in}\ c\,],\ e[\,c[\,\mathsf{out}\ e\,]\,]\,]$ peut évoluer vers le terme $a[\,e[\,h[\,\mathsf{in}\ c\,],\ c[\,\mathsf{out}\ e\,]\,]\,]$.

Remarque : Dorénavant, il ne sera pas nécessaire de justifier les évolutions aussi soigneusement qu'à la question précédente : plus besoin de faire référence à toutes les utilisations des règles **R1** à **R5**.

Question 1.2 Indiquer quelles autres évolutions immédiates peuvent avoir les termes E_1 et E_2 de la question ci-dessus.

Question 2 On dit qu'un terme E exhibe un choix s'il existe deux termes distincts E_1 et E_2 tels que $E \longrightarrow E_1$ et $E \longrightarrow E_2$. Le choix est dit *confluent* s'il existe un terme E' tel que $E_1 \longrightarrow^* E'$ et $E_2 \longrightarrow^* E'$, où $\longrightarrow^*$ représente un nombre fini quelconque (éventuellement nul) d'évolutions. Le choix est non confluent dans le cas contraire.

Donnez deux exemples de termes comportant au plus deux sites (pouvant éventuellement être sous-site l'un de l'autre), et exhibant l'un un choix confluent, l'autre un choix non confluent.

Question 3 On enrichit à présent le modèle en ajoutant un opérateur, noté !, qui peut être utilisé pour préfixer les particules, de manière à former des *particules éternelles* : alors que dans les règles **R1** à **R3** ci-dessus, chaque évolution a pour effet de faire disparaître une particule, une particule éternelle n'est pas effacée lors d'une évolution : ainsi l'évolution de la règle **R1** devient

$$a[\,!\mathsf{in}\ b,\ E\,],\ b[\,E'\,] \longrightarrow b[\,a[\,!\mathsf{in}\ b,\ E\,],\ E'\,]$$

... et de même pour les règles **R2** et **R3**. On travaille donc à présent avec des sites pouvant contenir des particules qui sont soit ephémères (sans !), soit éternelles.

Pour $k \geq 1$, on dit qu'un terme E *exhibe une boucle d'ordre* k s'il existe des termes $E_1, \ldots, E_k$ distincts de E et distincts deux à deux tels que l'on ait : (1) $E \longrightarrow E_1$; (2) pour tout $i \in [1, \ldots, k-1]$, $E_i \longrightarrow E_{i+1}$; (3) $E_k \longrightarrow E$.

Question 3.1 Donner un terme exhibant une boucle d'ordre 1, en justifiant votre réponse.

Question 3.2 (plus difficile) Même question avec une boucle d'ordre 2.

Question 3.3 Donner un terme exhibant une boucle d'ordre $k \geq 1$ et pouvant faire une évolution vers un terme bloqué (i.e. ne pouvant plus faire d'évolution).

Question 3.4 Soit un terme E d'ordre k avec $k \geq 1$. Montrer que dans aucune des évolutions $E \longrightarrow E_1 \longrightarrow \ldots \longrightarrow E_k \longrightarrow E$ il n'y a application de la règle **R3**.

Question 4 (plus difficile) À supposer que l'on dispose de structures de données adaptées à la manipulation des termes, décrire (aussi rigoureusement que possible) un algorithme récursif prenant en entrée un terme et renvoyant l'ensemble des termes vers lequel le terme donné en entrée peut faire une évolution.