

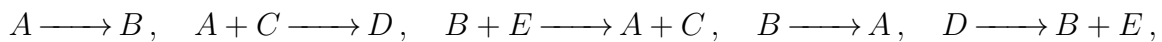
On cherche à concevoir des structures de données dédiées à l'analyse de systèmes de réactions chimiques. On suppose l'existence du fichier donné Figure 1.

Les systèmes de réactions chimiques généralisées fournissent un formalisme simple de modélisation des systèmes dynamiques. Un livre de référence est [1]. Le système suivant est une variante d'un modèle inventé par Alfred Lotka dans les années 1920 et Vito Volterra.

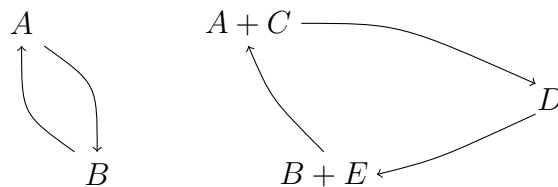


Les *espèces* chimiques sont désignées par les lettres H , L et Y . Elles représentent des quantités d'herbe, de lapins et de lynx¹. Le système est composé de trois *réactions* : (1), (2) et (3). Les expressions $H + L$, $H + 2L$, $L + Y$, $2Y$, Y et \emptyset sont appelées des *complexes*. Elles ont l'apparence de combinaisons linéaires d'espèces chimiques. Les coefficients sont des entiers. Ce sont les *coefficients stœchiométriques*. Par exemple, le coefficient stœchiométrique de l'espèce L dans le complexe $H + 2L$ est égal à 2 ; le coefficient stœchiométrique de l'espèce H est égal à 1. Le complexe qui apparaît en partie gauche d'une réaction est le *réactant* de la réaction. Le complexe qui apparaît en partie droite est le *produit* de la réaction.

Il est utile d'étudier un système du point de vue de la théorie des graphes. Un graphe souvent associé à un système de réactions chimiques est le *graphe FHJ* (du nom de Feinberg, Horn et Jackson) du système. Il s'agit d'un graphe orienté dont les sommets sont les complexes du système. Les arcs sont donnés par les réactions. Par exemple, le système suivant :



a pour graphe FHJ :



Question 1. On veut implanter les réactions ainsi que les listes chaînées de réactions. Pour chacun de ces types, définir un fichier d'entête, la déclaration C ainsi que les directives d'inclusion à placer dans les fichiers d'entête.

1. On utilise le vocabulaire de la chimie, bien qu'on n'ait pas affaire du tout à des substances réagissant dans des éprouvettes.

```

/*
 * N = le nombre maximal d'espèces chimiques autorisé
 *
 * Un complexe est défini par le tableau de ses coefficients stoechiométriques
 * coeff[0] = le coefficient stoechiométrique de A
 * coeff[1] = celui de B
 *      ...
 * coeff[N-1] = celui de la dernière espèce
 */

#define N 10

struct complexe {
    int coeff [N];
};

```

FIGURE 1 – Le fichier `complexe.h`.

On cherche à construire la matrice d'adjacence M du graphe FHJ d'un système \mathcal{S} de réactions chimiques. Cette matrice est indicée en lignes et en colonnes, par les complexes du système \mathcal{S} . Notons n le nombre de complexes de \mathcal{S} . Il s'agit donc d'attribuer un numéro appartenant à l'intervalle $[0, n - 1]$, à chaque complexe de \mathcal{S} . Ce numéro servira d'indice dans M .

On cherche donc une structure de données, une sorte de dictionnaire de complexes, qui permette d'attribuer un numéro à tous les complexes qu'elle contient. Une fois cette structure remplie, on veut pouvoir retrouver efficacement le numéro à partir du complexe.

Deux approches possibles sont listées ci-dessous.

1. Le programme qui utilise la structure de données procède en deux temps : il commence par enregistrer les complexes dans la structure sans se soucier de leur numéro. Quand tous les ajouts sont finis, il appelle une fonction/action chargée de numéroter tous les complexes enregistrés.
2. Un numéro est attribué à chaque complexe, dès son ajout dans la structure. La gestion du numéro est interne à la structure de données (la fonction ou l'action qui utilise la structure n'a pas besoin de gérer une variable locale `numero`, distincte de la structure).

Question 2. Décrire votre solution dans ses grandes lignes : quelle approche choisissez-vous ? quelle implantation de dictionnaire choisissez-vous, parmi celles qui ont été étudiées en cours ?

Question 3. Préciser votre solution : donner les déclarations de type et leurs spécifications.

Question 4. Si vous avez choisi la première approche, donner la fonction/action qui numérote les complexes quand les ajouts sont finis. Si vous avez choisi la seconde approche, donner la fonction/action qui effectue l'ajout. Dans les deux cas, vous pouvez définir des fonctions auxiliaires.

Question 5. Traduire en C, en utilisant votre solution, le pseudo-code de la Figure 2. Vous pouvez supposer que M est de dimension $NMAX \times NMAX$ où $NMAX$ est une constante supposée définie.

```
procedure matrice_adjacence_FHJ (M, L)
  M est une matrice initialement remplie de zéros (donnée/résultat)
  L est une liste de réactions (donnée)
begin
  Soit D une variable locale ayant le type que vous avez défini en question 9
  Initialiser D à vide
  for chaque réaction r de L do
    Ajouter à D le réactant et le produit de r (deux complexes)
  end do
  Si nécessaire, numéroter les complexes présents dans D
  for chaque réaction r de L do
    i := le numéro, dans D, du réactant de r
    j := le numéro, dans D, du produit de r
    M[i][j] := 1
  end do
end
```

FIGURE 2 – Pseudo-code construisant la matrice d'adjacence M du graphe FHJ d'un système de réactions chimiques.

Question 6. Pour tester si le théorème de déficience zéro s'applique à un système de réactions chimiques \mathcal{S} , il faut calculer le nombre de composantes connexes du graphe FHJ et tester si ces composantes connexes sont fortement connexes. Ces deux tests peuvent être effectués par des parcours en profondeur d'abord sur le graphe FHJ. Ces parcours peuvent se réaliser en utilisant des piles. Donner la déclaration et les spécifications d'un type pile adapté à ces parcours.

Références

- [1] Péter Érdi and János Tóth. *Mathematical models of chemical reactions : theory and applications of deterministic and stochastic models*. Princeton University Press, 1989.