Planche 4 type CCINP

juin 2024

1 Partie A :

On considère l'alphabet à deux lettres : $\Sigma = \{a, \bar{a}\}$.

Soit n un entier naturel non nul. Un buffer de taille n est modélisé par un automate $\mathcal{A}_n = (Q_n, \Delta_n, 0, F_n)$ sur l'alphabet Σ défini par :

- Q_n, l'ensemble des états de $\mathcal{A}_n$ est l'ensemble $\{0, 1, 2, \dots, n\}$. L'état i représente la présence de exactement i bits dans le buffer.
- La lettre a représente l'arrivée d'un bit dans le buffer. La lettre $\bar{a}$ représente la sortie d'un bit du buffer.
- Δ_n, est l'ensemble des transitions de $\mathcal{A}_n$:
  1. $\forall i \in \{0, \dots, n-1\}, (i, a, i+1)$
  2. $\forall i \in \{1, \dots, n\}, (i, \bar{a}, i-1)$
- 0 est l'état initial de $\mathcal{A}_n$
- $F_n = \{0, \dots, n\}$ est l'ensemble des états finaux.

On note $\mathcal{L}_n$ le langage reconnu par l'automate $\mathcal{A}_n$. Dans la suite, ϵ désigne le mot vide.

1. Soient m, n des entiers naturels non nuls. Justifier précisément l'équivalence :

$$\mathcal{L}_m \subset \mathcal{L}_n \Leftrightarrow m \leqslant n$$

   On pourra démontrer que $a^m \in \mathcal{L}_n$ équivaut à $m \leqslant n$.
2. Représenter l'automate $\mathcal{A}_1$; donner une expression rationnelle de $\mathcal{L}_1$.
3. Représenter l'automate $\mathcal{A}_2$; démontrer qu'une expression rationnelle de $\mathcal{L}_2$ est

$$\epsilon + a.(a\bar{a} + \bar{a}a)^*(\epsilon + a + \bar{a})$$

4. Soit n un entier naturel non nul. Le but de ces questions est de déterminer une relation de récurrence qui permettrait de calculer une expression rationnelle de $\mathcal{L}_n$. Pour j dans $\{0, \dots, n\}$, on note $\mathcal{L}_n^{0,j}$ le langage reconnu par l'automate $\mathcal{A}_n^{0,j} = (Q_n, \Delta_n, 0, \{j\})$, automate dont l'ensemble des états et l'ensemble des transitions sont identiques à ceux de $\mathcal{A}_n$ mais ayant le sommet j comme unique état final.
   (a) Démontrer la relation de récurrence : $\mathcal{L}_n^{0,0} = \{\epsilon\} \cup a \left\{\bar{a}a, \mathcal{L}_{n-1}^{0,0}\right\}^* \bar{a}$
   (b) Soit $j \in \{1, \dots, n\}$. Démontrer : $\mathcal{L}_n^{0,j} = \mathcal{L}_n^{0,0} a \mathcal{L}_{n-1}^{0,j-1}$.
   (c) Conclure.

2 Partie B : à traiter en Ocaml

Une liste d'entiers est dite convenable si elle se compose d'un nombre pair d'entiers $a_1, b_1, a_2, \dots, a_n.b_n$ tels que $a_1 \leq b_1 < a_2 \leq b_2 < \dots \leq b_{n-1} < a_n \leq b_n$.

On représente une réunion finie de k intervalles fermés disjoints dont les extrémités sont des entiers relatifs par la liste ordonnée de leurs extrémités. On admet que cette liste est convenable. L'ensemble vide est représenté par la liste vide.

1. Donner la liste représentant l'ensemble d'intervalles suivant : $[-1, 2], [4, 6], [-3, -2], [8, 8]$.
2. Ecrire une fonction `convenable : int list->bool` qui prend en entrée une liste et qui renvoie `true` si et seulement si elle est convenable.
3. On admet que l'intersection de deux réunions finies d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs est aussi une réunion d'intervalles fermés finis dont les extrémités sont des entiers relatifs. Donner la représentation de l'intersection de $[1, 2] \cup [4, 5]$ avec $[-1, 3] \cup [5, 7] \cup [8, 9]$ sous forme de liste convenable.
4. Ecrire une fonction `intersection : int list-> int list -> int list` qui prend en entrée deux listes convenables et qui renvoie une liste convenable qui est la représentation de l'intersection des deux réunions finies de segments disjoints représentées par chacune des listes passées en argument. On pourra utiliser l'exemple de la question précédente pour tester son résultat.

5. On admet que l'union de deux réunions finies d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs est aussi une réunion d'intervalles fermés finis dont les extrémités sont des entiers relatifs. Donner la représentation de la réunion de $[1,2] \cup [4,5]$ avec $[-1,3] \cup [5,7] \cup [8,9]$ sous forme de liste convenable.
6. Ecrire une fonction `union : int list-> int list -> int list` qui prend en entrée deux listes convenables et qui renvoie une liste convenable qui est la représentation de l'intersection des deux réunions finies de segments disjoints représentées par chacune des listes passées en argument. On pourra utiliser l'exemple de la question précédente pour tester son résultat.