L'objectif de ce TD est de montrer l'implication "automate → expression rationnelle" du théorème de Kleene :

Théorème 1. *(Kleene, sens "automate → expression rationnelle")*
Soit $\mathcal{A}$ un automate fini reconnaissant le langage $\mathcal{L} = \mathcal{L}(\mathcal{A})$.
Il existe une expression rationnelle E telle que $\mathcal{L}(E) = \mathcal{L}$. De plus, le calcul de E à partir de $\mathcal{A}$ est effectif – c'est-à-dire qu'il existe un algorithme renvoyant un tel E sur l'entrée $\mathcal{A}$.

Nous allons décrire dans les grandes lignes la preuve du Théorème 1 et nous convaincre de son exactitude.
Fixons un alphabet Σ, et considérons un automate fini $\mathcal{A}$ (possiblement avec epsilon-transitions) reconnaissant le langage $\mathcal{L}$ sur Σ.

Question 1. *Expliquez, quitte à modifier légèrement l'automate $\mathcal{A}$, pourquoi on peut supposer que $\mathcal{A}$ possède exactement un état initial q_i et un état final q_f avec $q_i \neq q_f$.*

L'idée générale consiste à éliminer un par un les états de $\mathcal{A}$, jusqu'à ce qu'il ne reste plus que q_i et q_f dans $\mathcal{A}$. Pour ce faire, on va considérer un type plus général d'automate, où les transitions seront étiquetées non plus par des lettres, mais par des expressions rationnelles. Dans ce contexte, pour passer d'un état à un autre, on lit non plus une lettre, mais un mot qui appartient au langage de l'expression rationnelle qui étiquette la transition entre les deux états.
Plus formellement, une exécution de $\mathcal{A}$ sur le mot u est une séquence

$$q_i = q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} q_2 \cdots q_{n-1} \xrightarrow{u_n} q_n$$

où

- $u = u_1 \cdot u_2 \cdots u_n$,
- pour chaque i, $u_i \in \mathcal{L}(E_i)$ où E_i étiquette la transition de q_{i-1} vers q_i.

Question 2. *On souhaite avoir au plus une transition entre deux états. À supposer qu'il existe plusieurs transitions entre deux états p et p' de $\mathcal{A}$, expliquez comment on peut regrouper ces transitions en une seule (étiquetée par une expression rationnelle) sans changer le langage reconnu par l'automate.*
Appliquez cette technique sur l'automate de la Figure 1.

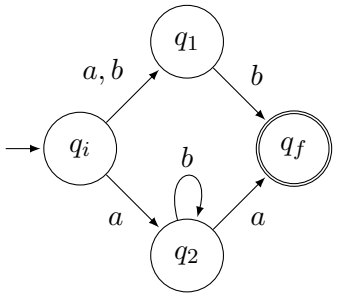


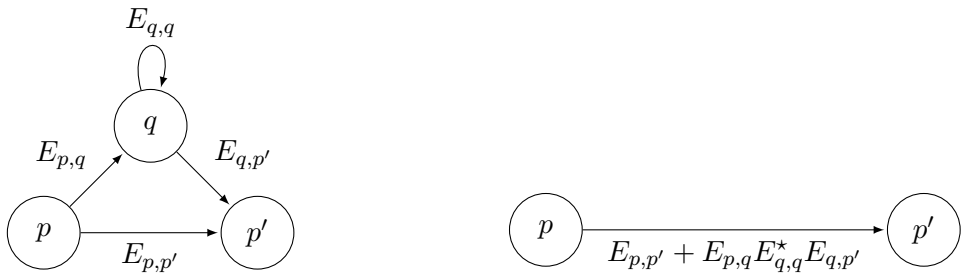
FIGURE 1 – Premier exemple.

Question 3. *On décide de supprimer l'état q_1 de l'automate de la Figure 1. Comment modifier les transitions (étiquetées par des expressions rationnelles) de cet automate pour que le langage reconnu reste le même.*

Question 4. *Après avoir supprimé l'état q_1, on décide de supprimer l'état q_2. Comment doit-on maintenant modifier les transitions de cet automate de sorte qu'il conserve le même langage ?*

Décrivons maintenant la procédure d'élimination des états dans un cadre général :

```
Tant qu'il reste un état q différent de q_i et q_f
  Pour chaque états p et p' (y compris p = p') comme dans la Figure 2a
    // avec possiblement E_{p,p'} = ∅ et E_{q,q} = ε
    Modifier l'expression étiquetant la transition de p à p' comme en
Figure 2b
  Supprimer l'état q
L'expression E souhaitée est celle qui étiquette la transition de q_i à q_f
```



(a) Transitions entre p, p' et q dans $\mathcal{A}$ (b) Transition entre p et p' après suppression de q

FIGURE 2 – Conséquence sur la transition de p à p' de l'élimination de l'état q dans $\mathcal{A}$.

Question 5. *Que peut-on dire du langage reconnu par l'automate obtenu après application de l'algorithme précédent sur $\mathcal{A}$?*

Question 6. *En appliquant cette procédure, donnez une expression rationnelle reconnaissant le même langage que l'automate de la Figure 3.*

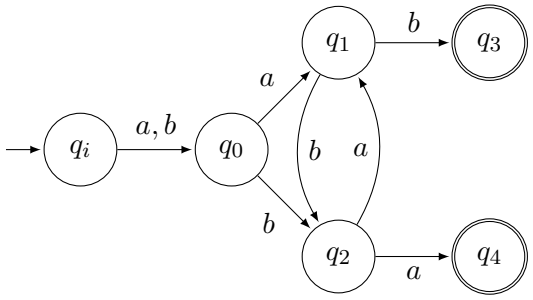


FIGURE 3 – Deuxième exemple.