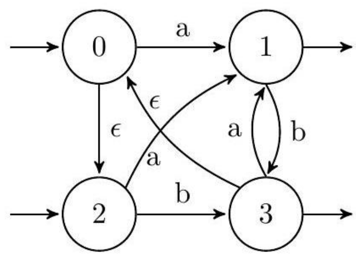
I Langage local, linéaire et automate de Glushkov

1. Construire l'automate de Glushkov reconnaissant $(ab+b)^*b$.
2. Montrer que si e est une expression linéaire alors $L(e)$ est un langage local.
3. Montrer qu'il existe un nombre fini de langages locaux sur un alphabet fixé.
4. Soient $\mathcal{L}_{\text{rat}}$, $\mathcal{L}_{\text{loc}}$, $\mathcal{L}_{\text{lin}}$ l'ensemble des langages rationnels, locaux et linéaires (c'est à dire décrits par une expression rationnelle linéaire). Quelles sont les inclusions entre ces 3 ensembles? Ces inclusions sont-elles strictes?

II Exercice CCP (élimination des états)



1. Déterminiser cet automate.
2. Construire une expression régulière dénotant le langage reconnu par cet automate, à l'aide de la méthode d'élimination des états.
3. Décrire simplement avec des mots le langage reconnu par cet automate.

III Calcul des ensembles $P(L)$, $S(L)$, $F(L)$

Soit L un langage. On rappelle les définitions du cours :

- $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
- $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
- $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)

Dans la suite, on utilise le type suivant d'expression rationnelle :

```
type 'a regexp =
    | Vide | Epsilon | L of 'a
    | Union of 'a regexp * 'a regexp
    | Concat of 'a regexp * 'a regexp
    | Etoile of 'a regexp
```

1. Écrire une fonction `has_eps : 'a regexp -> bool` déterminant si le langage d'une expression rationnelle contient ε.
2. Écrire une fonction `union : 'a list -> 'a list -> 'a list` telle que, si `l1` et `l2` sont des listes sans doublon (ce qu'on suppose être le cas...), `union l1 l2` renvoie une liste sans doublon contenant les éléments des deux listes. Par exemple, `union [1; 2] [3; 1]` peut renvoyer `[1; 2; 3]` (l'ordre des éléments de la liste de retour n'importe pas).
3. Écrire une fonction `prefixe` de type `'a regexp -> 'a list` telle que `prefixe e` renvoie $P(L(\texttt{e}))$.
4. Que faudrait-il modifier à `prefixe` pour obtenir une fonction `suffixe` renvoyant $S(L)$?
5. Écrire une fonction `produit : 'a list -> 'b list -> ('a * 'b) list` effectuant le produit cartésien de 2 listes : si `l1` et `l2` sont des listes, `produit l1 l2` renvoie une liste de tous les couples distincts composés d'un élément de `l1` et un élément de `l2`. Par exemple, `produit [1; 2] [3; 1]` peut renvoyer `[(1, 3); (1, 1); (2, 3); (2; 1)]` (l'ordre des éléments de la liste de retour n'importe pas).
6. En déduire une fonction `facteur` de type `'a regexp -> ('a * 'a) list` telle que `facteur e` renvoie $F(L(\texttt{e}))$.

IV Stabilité des langages rationnels

Pour chacune des propositions suivantes, expliquer pourquoi elle est vraie ou donner un contre-exemple :

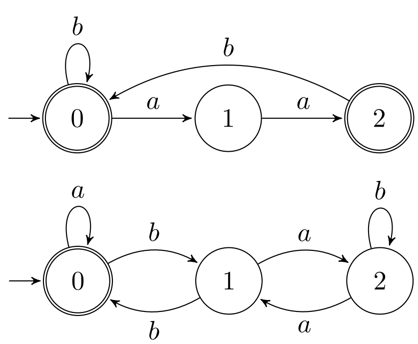
1. Si L est rationnel et $L \subseteq L'$ alors L' est rationnel.
2. Si L' est rationnel et $L \subseteq L'$ alors L est rationnel.
3. Si L est rationnel alors L^* est rationnel.
4. Si L^* est rationnel alors L est rationnel.
5. Si L est rationnel sur un alphabet Σ alors $\Sigma^* \setminus L$ (complémentaire de L) est rationnel.
6. Une union finie de langages rationnels est rationnel (L_1, ..., L_n rationnels $\implies \bigcup_{k=1}^{n} L_k$ rationnel).
7. Une union dénombrable de langages rationnels est rationnel (L_1, ..., L_n, ... rationnels $\implies \bigcup_{k=1}^{\infty} L_k$ rationnel).
8. Une intersection finie de langages rationnels est rationnel (L_1, ..., L_n rationnels $\implies \bigcap_{k=1}^{n} L_k$ rationnel).
9. Une intersection dénombrable de langages rationnels est rationnel (L_1, ..., L_n, ... rationnels $\implies \bigcap_{k=1}^{\infty} L_k$ rationnel).

V Reconnaissable $\implies$ rationnel avec le lemme d'Arden

1. (Lemme d'Arden) Soient A et B deux langages tels que $\varepsilon \notin A$. Montrer que l'équation $L = AL \cup B$ (d'inconnue le langage L) admet pour unique solution A^*B.

Soit $(\Sigma, Q, q_0, F, \delta)$ un automate déterministe. Si $q_i \in Q$ est un état, on pose $L_i = \{m \in \Sigma^* \mid \delta^*(q_i, m) \in F\}$ (étiquettes des chemins de q_i vers un état final). En écrivant des relations entre les L_i puis en résolvant de proche en proche avec le lemme d'Arden, on obtient une expression rationnelle pour le langage L_0 de l'automate.

2. Appliquer cette méthode pour trouver une expression rationnelle pour chacun des automates ci-dessous. En remplaçant a par 0 et b par 1, que reconnaît le 2ème automate?



VI Reconnaissable $\implies$ rationnel par prog. dyn. ($\approx$ Floyd-Warshall)

Soit $(\Sigma, Q, 0, F, \delta)$ un automate déterministe dont les états sont des entiers de 0 à n (et 0 est l'état initial).
Soit $L(i, j, k)$ le langage des étiquettes des chemins de i à j n'utilisant que des états intermédiaires strictement inférieurs à k.

1. Montrer que $L(i, j, 0)$ est rationnel, pour tous les états i, j.
2. Prouver une équation de récurrence sur $L(i, j, k)$.
3. En déduire, par récurrence, que tout langage reconnaissable est rationnel.

VII Racine d'un langage

Soit L un langage reconnaissable sur Σ. Montrer que $\sqrt{L} = \{m \in \Sigma^* \mid m^2 \in L\}$ est un langage reconnaissable.