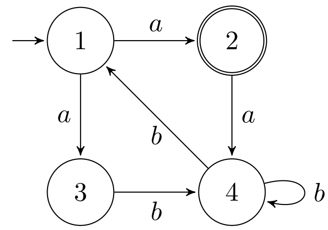
I Algorithme de déterminisation

1. Déterminiser l'automate suivant en utilisant l'algorithme du cours:



2. Donner une expression rationnelle pour le langage reconnu par cet automate.

II Clôture des langages reconnaissables

Si $m = m_1...m_n$ est un mot, on définit son miroir $\widetilde{m} = m_n...m_1$. Si L est un langage, on définit son miroir $\widetilde{L} = \{\widetilde{m} \mid m \in L\}$.

1. Montrer que le miroir d'un langage reconnaissable est reconnaissable.

Si L est un langage sur Σ, on définit:

- $Pref(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*,\ uv \in L\}$: ensemble des préfixes des mots de L.
- $Suff(L) = \{u \in \Sigma^* \mid \exists v \in \Sigma^*,\ vu \in L\}$: ensemble des suffixes des mots de L.
- $Fact(L) = \{u \in \Sigma^* \mid \exists v, w \in \Sigma^*,\ vuw \in L\}$: ensemble des facteurs des mots de L.

2. Montrer que si L est reconnaissable alors $Pref(L)$, $Suff(L)$, $Fact(L)$ le sont aussi.
3. Montrer que si L est rationnel alors $Pref(L)$, $Suff(L)$, $Fact(L)$ le sont aussi (puisqu'on va montrer que rationnel = reconnaissable, c'est une preuve alternative à la précédente).

III Reconnaissable ou non ?

Pour chacun de ces langages, dire s'il est reconnaissable ou non. Justifier.

1. L_1 = mots sur $\{a, b\}$ sans lettres consécutives égales.
2. L_2 = mots sur $\{a, b\}$ ayant un nombre pair de a et dont le nombre de b est multiple de 3.
3. $L_3 = \{m \in \{a, b\}^* \mid |m|_a = |m|_b\}$ (où $|m|_a$ est le nombre de a du mot m).
4. L_4 = écritures en base 2 des multiples de 5.
5. $L_5 = \{a^p \mid p \text{ est un nombre premier}\}$.

IV Algorithmes sur les automates

1. À quelle condition nécessaire et suffisante simple le langage reconnu par un automate est vide? Décrire un algorithme pour le savoir.
2. À quelle condition nécessaire et suffisante simple le langage reconnu par un automate est fini? Décrire un algorithme pour le savoir.
3. Décrire un algorithme pour déterminer si deux automates admettent le même langage.
4. Soit A un automate à n états. Montrer que si $L(A)$ est non vide alors il contient un mot de longueur $\leq n - 1$.
5. On dit qu'un automate est **émondé** si, pour tout état q, il existe d'une part un chemin d'un état initial à q et d'autre part un chemin de q à un état final. Montrer que tout automate est équivalent à un automate émondé.

V Oral ENS info

On fixe un alphabet Σ avec $|\Sigma| > 1$. Un mot $w \in \Sigma^*$ est un palindrome s'il s'écrit $w = a_1 \cdots a_n$ et qu'on a $a_i = a_{n-i+1}$ pour tout $1 \leq i \leq n$. On note $\Pi \subseteq \Sigma^*$ le langage des palindromes. Pour un automate fini A sur Σ, on note $L(A)$ le langage reconnu par A.

1. Soit $\Pi_n := \Pi \cap \Sigma^n$. Montrer que pour tout automate fini déterministe complet A, pour tout $n \in \mathbb{N}$, si $L(A) \cap \Sigma^{2n} = \Pi_{2n}$, alors A a au moins $|\Sigma|^n$ états.
2. En déduire que le langage Π n'est pas régulier.
3. Étant donné un automate fini A sur Σ, peut-on calculer un automate A_Π qui reconnaisse $L(A) \cap \Pi$?
4. Pour tout mot $u = b_1 \cdots b_m$ de Σ^*, on note $\overline{u} := b_m \cdots b_1$ son miroir. Étant donné A, peut-on calculer un automate A'_Π qui reconnaisse $\{u \in \Sigma^* \mid u\overline{u} \in L(A)\}$?
5. On appelle Π_{pair} l'ensemble des palindromes de longueur paire, i.e., $\Pi_{\text{pair}} = \bigcup_{n\in\mathbb{N}} \Pi_{2n}$. Proposer un algorithme qui, étant donné un automate fini A sur Σ, détermine si $L(A) \cap \Pi_{\text{pair}}$ est vide, fini, ou infini. Discuter de sa complexité en temps et en espace.
6. Modifier l'algorithme de la question 4 pour calculer la cardinalité de $L(A) \cap \Pi_{\text{pair}}$ quand cet ensemble est fini, en faisant l'hypothèse que l'automate d'entrée A est déterministe. Comment la complexité est-elle affectée?
7. Modifier l'algorithme des questions 4 et 5 pour qu'il s'applique à $L(A) \cap \Pi$.

VI Algorithme KMP

On s'intéresse à la recherche d'un facteur $m = m_1 ... m_n$ dans un texte, sur un alphabet Σ.

1. Écrire une fonction naïve de type `'a list -> 'a list -> bool` pour résoudre ce problème.
   Quelle est sa complexité?
2. Expliquer comment construire un automate déterministe complet reconnaissant les mots ayant m comme facteur.
   Une fois cet automate construit, quelle est la complexité pour savoir si un mot contient m comme facteur?

On note $Pref(m)$ l'ensemble des préfixes de m.
L'algorithme KMP (Knuth-Morris-Pratt) consiste à considérer l'automate $K = (\Sigma, Pref(m), \varepsilon, \{m\}, \delta)$, où l'état initial est ε, l'état final est m et $\delta(p, a)$ est le plus long suffixe du mot pa qui est aussi préfixe de m.

3. Dessiner K si $m = 01000$ (avec $\Sigma = \{0, 1\}$). Quel est le langage reconnu par K? Comment modifier K pour qu'il reconnaisse $\Sigma^* m \Sigma^*$?
4. Écrire une fonction `delta : 'a list -> 'a list -> 'a list` telle que `delta u m` renvoie le plus long suffixe de `u` qui est préfixe de `m`.
5. Comparer la complexité de l'algorithme KMP avec les deux méthodes précédentes.

Remarque : Il est possible de calculer δ plus rapidement pour construire l'automate K, avec une complexité $O(|m|)$.

VII Résiduel

Soit L un langage sur un alphabet Σ. Soit $u \in \Sigma^*$. On définit le langage $u^{-1}L = \{v \in \Sigma^* \mid uv \in L\}$ (qu'on appelle résiduel de L).

1. Montrer que si L est reconnaissable alors $u^{-1}L$ est reconnaissable.
2. Quels sont tous les résiduels possibles de a^*b^*? De $\{a^n \mid n \text{ est pair}\}$?
3. Montrer que si L est reconnaissable alors $\{u^{-1}L \mid u \in \Sigma^*\}$ est fini (il n'y a qu'un nombre fini de valeurs possibles pour $u^{-1}L$ quand u varie dans Σ^*).
4. Montrer que $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas reconnaissable.

Un mot m est un palindrome s'il se lit de la même façon dans les deux sens (ou encore: $\widetilde{m} = m$).

5. Écrire une fonction Caml pour déterminer si une liste de lettres est un palindrome. Complexité?
6. Montrer que l'ensemble des palindromes (sur un alphabet à au moins 2 lettres) n'est pas reconnaissable.