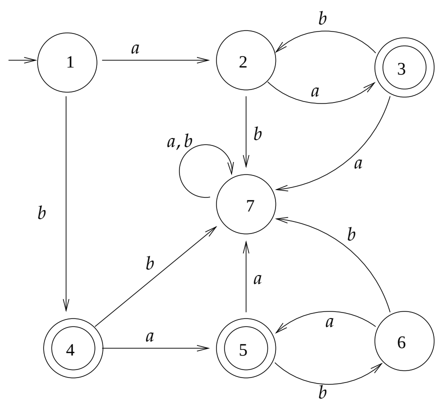
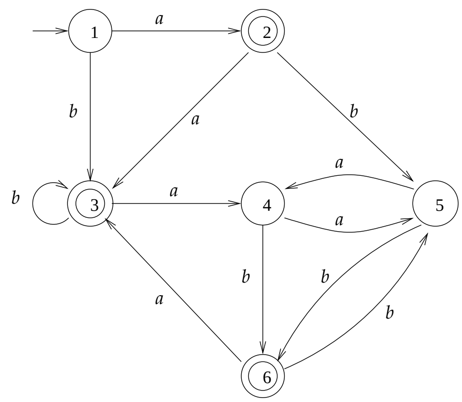
Module Langages Formels
TD 4 : Minimisation et Langages Rationnels

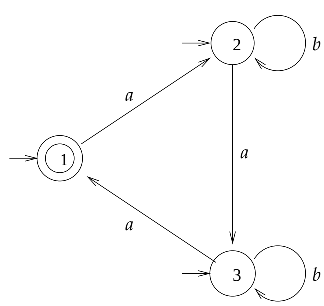
Exercice 1 `Minimisation`

1.1. Minimiser les automates suivants en utilisant l'algorithme vu en cours :





1.2. Déterminiser et minimiser l'automate suivant. A votre avis si on généralise à n états, combien d'états aura le déterminisé ? Le minimal ?



Exercice 2 `Caractérisation de Nérode`

Définition:
Une relation d'équivalence $\equiv$ est une **congruence à droite** ssi $\forall t \in \Sigma^*,\ u \equiv v \Rightarrow ut \equiv vt$.

C'est une **congruence à gauche** ssi $\forall t \in \Sigma^*,\ u \equiv v \Rightarrow tu \equiv tv$. On parle de **congruence** si elle est compatible à droite et à gauche.

Une congruence est **d'index fini** ssi l'ensemble de ses classes d'équivalence est fini.

Nous allons prouver le thérorème suivant :

Théorème (Myhill-Nérode) :
Il y a équivalence entre les assertions suivantes :

1. L est rationnel.
2. L est une union de classes d'une congruence d'index fini.
3. L est une union de classes d'une congruence à droite d'index fini.
4. La relation $\sim_L$ définie par $u \sim_L v \Leftrightarrow \forall t \in \Sigma^*(ut \in L \Leftrightarrow vt \in L)$ est une congruence à droite d'index fini.

2.1. $1 \Rightarrow 2$

Soit un langage L rationnel, reconnu par un automate fini déterministe $A = (Q, \Sigma, \delta, q_0, F)$. On définit $u \sim_A v$ par $\forall q \in Q,\ \delta(q, u) = \delta(q, v)$.

2.1. 1. Montrer que $\sim_A$ est une congruence.

On considère l'application

$$\varphi : \begin{array}{ccc} \Sigma^*/_{\sim_A} & \rightarrow & Q^Q \\ \overline{u}^A & \mapsto & (q \mapsto \delta(q, u)) \end{array}$$

2.1. 2. Montrer que φ est injective. En déduire que $\sim_A$ est d'index fini, et que L est l'union de certaines de ses classes d'équivalence.

2.2. $3 \Rightarrow 4$

Soit $\equiv$ la congruence à droite d'index fini donnée par hypothèse. On sait que $L = \bigcup_{x \in X} \overline{x}$.

2.2. 1. Montrer que $\sim_L$ est une congruence à droite.

2.2. 2. Montrer que $\forall u, v \in \Sigma^*,\ u \equiv v \Rightarrow u \sim_L v$. En déduire que $\sim_L$ est d'index fini.

2.3. $4 \Rightarrow 1$

On suppose que $\sim_L$ est une congruence à droite d'index fini, montrer que $u \sim_L v \Leftrightarrow u^{-1}L = v^{-1}L$. En conclure que L est rationnel.

2.4. Utiliser le théorème de Myhill-Nérode pour montrer que $\{a^n b^n \mid n \in \mathbb{N}\}$ n'est pas rationnel.

Exercice 3 `Caractérisation logique de` Rat(Σ)

Le but de cet exercice est d'illustrer le théorème de Büchi, qui donne une caractérisation logique de l'ensemble des langages reconnaissables :

> **Théorème (Büchi 1960, Elgot 1961)** :
> Un langage de mots finis est reconnaissable par automate fini si et seulement si il est définissable en logique MSO[S] (Monadique du second ordre avec successeur).

Une logique du premier ordre se construit avec
- Un ensemble de variables : $x, y \ldots$
- Des symboles de relations (unaires ou binaires dans notre cas) : $S, <, = \ldots$
- Des connecteurs logiques : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- Des quantificateurs : $\forall, \exists$

Pour une logique du second ordre, on ajoute
- Des variables du second ordre : $X, Y \ldots$
- L'application d'une variable du premier ordre à une variable du second ordre (en tant que formule atomique) : $X(x)$

On parlera de logique monadique si on ne quantifie que des symboles de relation unaire.

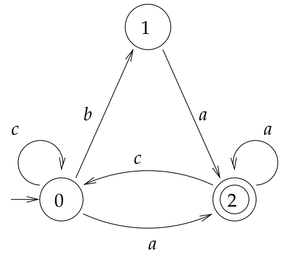
On utilisera par la suite les variables x et y pour dénoter la position d'une lettre dans un mot. S correspondra à la relation successeur sur les entiers ($S : \begin{array}{l}[0,n] \to [0,n] \\ 0 \leq k < n \mapsto k+1\end{array}$, et $Q_a(x)$ signifiera que la lettre a est à la position x.

$\underline{w}$ est un modèle d'une formule si on connaît : $\mathsf{dom}(w)$, S_w, $(Q_a)_{a\in\Sigma}$. (le mot w est considéré comme une fonction de $[1, n]$ dans Σ). Le langage décrit par une formule est **l'ensemble des modèles $\underline{w}$ vérifiant cette formule**.

Par exemple, la formule logique suivante signifie que "toute occurence de b est suivie d'une occurence de a" :

$$\forall x (Q_b(x) \Rightarrow (\exists y,\ Sxy \wedge Q_a(y)))$$

On considère l'automate suivant, dont le langage reconnu est tel que
- toute occurence de b est suivie d'une occurence de a ;
- aucune occurence de b ne suit immédiatement une occurence de a ;
- tout mot reconnu se termine par une occurence de a.



3.1. Construire la formule logique correspondant au langage reconnu par cet automate.

3.2. En supposant qu'il existe un prédicat X exprimant "*x est en position impaire*", donner la formule qui dénote l'ensemble des mots de longueur paire.

$$\exists X \begin{pmatrix} & (\dots) \\ \wedge & (\dots) \\ \wedge & (\dots) \end{pmatrix}$$

Soit L un langage reconnu par un automate fini $A = (Q, \Sigma, \delta, q_0, F)$, avec $n = |Q|$. On veut construire la formule logique dénotant L. Pour chaque état q_i, on supposera l'existence d'un prédicat X_i, $X_i(x)$ signifiant "*avant de lire la x^e lettre, l'automate est dans l'état q_i*".

3.3. Proposer une formule (1) signifiant que, lors de la lecture d'un mot, l'automate est toujours dans un et un seul état.

3.4. Proposer une formule (2) signifiant que, lors de la lecture d'un mot, l'automate commence toujours en q_0 (on notera $\mathsf{min}(x)$ la formule $\neg\exists z,\ Szx$).

3.5. Proposer une formule (3) signifiant que, lors de la lecture d'un mot chaque transition est une transition valide de l'automate.

3.6. Proposer une formule (4) signifiant que la lecture d'un mot de L termine toujours par un état final (on notera $\mathsf{max}(x)$ la formule $\neg\exists z,\ Sxz$).

La formule suivante caractérise donc exactement les mots du langage L :

$$\exists X_1 \exists X_2 \dots \exists X_n \begin{pmatrix} & (1) \\ \wedge & (2) \\ \wedge & (3) \\ \wedge & (4) \end{pmatrix}$$