IUT “A”
Département Informatique

Année Universitaire 2009/2010
Semestre 3

Modélisation[1]

Théorie des Graphes et Théorie des Langages

Max Chlebowski, Serge Iovleff, Bruno Jedynak

[1] Les TD sur les langages sont de Max Chlebowski. Les TD sur les graphes ont été été créés à l’origine par Bruno Jedynak puis repris par Serge Iovleff en 2003. Ils évoluent chaque année grâce aux différents intervenants dans cette matière. Je remercie en particulier Maxence Cuvilliez et Adel Guerziz pour leurs apports à ces TD.

Table des matières

Chapitre 1

Théorie des Graphes

1.1 Relations binaires

Exercice 1 : Sur l'ensemble $S = \{1, 2, 3, 5\}$, on considère les relations suivantes :
- xRy ssi $x + y$ est impair
- xSy ssi $x \leq y$

1. Représenter ces deux relations par une matrice et un graphe.
2. Quelles propriétés ont ces deux relations (parmi réflexive, symétrique, antisymétrique, transitive) ?
3. Donner la matrice et le graphe de $R \cap S$, $R \cup S$, $\overline{R}$, et S^t. Donner, si possible, une caractérisation de ces relations.

□

Exercice 2 : Les relations suivantes ont-elles réflexives, symétriques, antisymétriques ou transitives ? (Réponses à justifier).

1. R_1 est définie sur l'ensemble $E = \{a, b, c, d\}$ par la matrice :

	a	b	c	d
a	1	1		1
b	1	1		
c	1	1	1	
d				1

2. Sur l'ensemble $E = \{1, 2, 3, 4, 5, 6\}$

$$\begin{aligned} xR_2y &\Leftrightarrow x \text{ divise } y \\ xR_3y &\Leftrightarrow x \text{ est multiple de } y \end{aligned}$$

Existe t-il un lien entre R_2 et R_3 ?

3. Sur $\mathbb{N}$

$$xR_4y \quad \Leftrightarrow x - y \quad \text{est pair}$$

Peut-on exprimer autrement R_4 ?

□

Exercice 3 : (relations sur l'ensemble des parties d'un ensemble)

Soit $E = \mathcal{P}(X)$, l'ensemble des parties d'un ensemble X non vide. Définissons sur E, les relations suivantes :

$\forall A \in E, \forall B \in E$

$$\begin{array}{rll} A\mathcal{R}B & \text{ssi} & A \cap B = \emptyset \\ A\mathcal{S}B & \text{ssi} & A \subset B \\ A\mathcal{T}B & \text{ssi} & A \not\subset B \\ A\mathcal{U}B & \text{ssi} & A\mathcal{S}B \text{ ou } B\mathcal{S}A \\ A\mathcal{V}B & \text{ssi} & A\mathcal{S}B \text{ et } B\mathcal{T}A \end{array}$$

Pour chaque relation, étudiez les 4 propriétés suivantes : **réflexivité** (R), **symétrie** (S), **antisymétrie** (A) et **transitivité** (T).

□

Exercice 4 :

1. Que peut-on dire d'une relation à la fois symétrique et antisymétrique ?
2. On dit qu'une relation R sur un ensemble E possède la propriété (P) si l'on a jamais à la fois xRy et yRx :
   (a) Une telle relation est-elle antisymétrique ?
   (b) Une relation antisymétrique a-t-elle la propriété (P) ?

□

Exercice 5 : (un grand tableau)

Nous considérons les 4 propriétés suivantes : **réflexivité** (R), **symétrie** (S), **antisymétrie** (A) et **transitivité** (T).

Examiner les 16 combinaisons possibles des 4 propriétés ci-dessus (présence ou absence de la propriété) en donnant, si possible dans chaque cas un exemple de relation sur un ensemble contenant 3 éléments. Vous préciserez alors son graphe.

□

Exercice 6 : (composition de relations)

Soient $\mathcal{R}$ et $\mathcal{S}$, deux relations sur un ensemble E. Définissons la **composition des relations $\mathcal{R}$ et $\mathcal{S}$** de la manière suivante : $\forall x, y \in E$,

$$x(\mathcal{R} \circ \mathcal{S})y \text{ ssi } \exists z \in E, x\mathcal{R}z \text{ et } z\mathcal{S}y$$

1. Si E est un ensemble de personnes, $\mathcal{R}$ est la relation "être le fils de", et $\mathcal{S}$ est la relation "être le frêre de", que sont les relations $\mathcal{R} \circ \mathcal{S}$ et $\mathcal{S} \circ \mathcal{R}$?
2. Sur $E = \{a, b, c\}$, définissons les relations

$$\begin{array}{rcl} \mathcal{R} & = & \{(a,a),(a,b),(b,c),(c,b)\} \\ \mathcal{S} & = & \{(a,b),(b,c),(a,c)\} \end{array}$$

   Dessinez les matrices et les graphes des relations $\mathcal{R} \circ \mathcal{S}$ et $\mathcal{S} \circ \mathcal{R}$.
3. L'ensemble E est de cardinal fini. Donnez une méthode pour effectuer les calculs de composition de relations qui utilise le calcul **booléen** *et* le calcul **matriciel**.
4. Pour les relations de la question **2**, calculez $(\mathcal{R} \circ \mathcal{S})^t$ et $\mathcal{S}^t \circ \mathcal{R}^t$.
5. Montrez qu'en général, on a
$$(\mathcal{R} \circ \mathcal{S})^t = \mathcal{S}^t \circ \mathcal{R}^t$$

6. Pour la relations de la question **2**, calculez $\mathcal{R} \circ \mathcal{R}$ et $\mathcal{S} \circ \mathcal{S}$.
7. Montrez qu'une relation $\mathcal{R}$ quelconque est transitive si et seulement si

$$\mathcal{R} \circ \mathcal{R} \subset \mathcal{R}$$

□

1.2 Introduction aux Graphes

1.2.1 Exercices d'introduction

Exercice 7 : Définir le **sous-graphe** du graphe de la figure 1.1 engendré par les sommets

$$\{1, 2, 3, 4, 5, 6, 7, 8\}$$

ainsi que le **graphe partiel** engendré par les arêtes

$$\{(1, 9), (3, 9), (5, 9), (7, 9)\}.$$

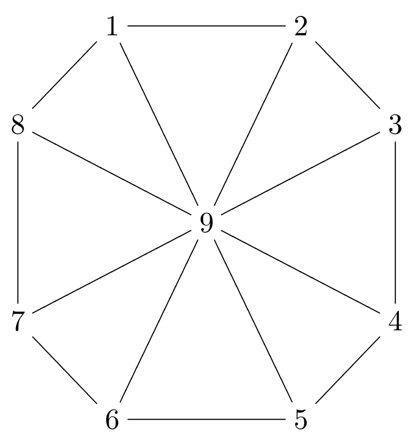


Fig. 1.1 – Sous-graphe et graphe partiel

□

Exercice 8 : Soit $G = \{S, A\}$ le graphe non orienté définie par $S = \{a, b, c, d, e\}$ et

$$A = \{(a, c), (a, e), (b, c), (c, d), (c, e), (d, e)\}$$

1. Représenter G
2. Écrire sa matrice et sa liste d'adjacence.
3. Dresser un tableau collectant les degrés des différents sommets. Quel est le degré du graphe ?

□

Exercice 9 : Pour un graphe non orienté, $G = (S, A)$, la **matrice d'incidence** $B = (b_{ij})$ est une matrice comportant $|S|$ lignes et $|A|$ colonnes. b_{ij} est le nombre de fois où le sommet i est incident à l'arête d'indice j (2 fois dans le cas d'une boucle).

Écrire la matrice d'incidence associée au graphe $G = (S, A)$, tel que $S = \{1, 2, 3, 4\}$ et $A = \{a_1, a_2, a_3, a_4, a_5\}$, avec $a_1 = (1, 2)$, $a_2 = (2, 3)$, $a_3 = (3, 1)$, $a_4 = (4, 1)$ et $a_5 = (4, 4)$. Que vaut la somme des éléments d'une colonne ? Que représente la somme des éléments d'une ligne ?

En déduire que dans un graphe non orienté $G = (S, A)$, on a

$$\sum_{x \in S} d(x) = 2|A|$$

et en conclure que pour tout graphe **non orienté**, le nombre de sommets de degré impair est pair.

□

Exercice 10 : Une ligue de Quidditch comporte 5 équipes. Le ministère de la magie prévoit le nombre de match par saison pour chaque équipe.

1. Chaque équipe joue 1 partie contre chacune des autre équipes. Dessiner le graphe associé. Que peut-on dire de ce graphe ?
2. Peut-on faire jouer exactement trois parties a chaque équipe ? Exactement 2 parties ?

□

Exercice 11 :[Généralisation] Le graphe **complet** $K_n^\star$ est le graphe orienté comportant n sommets tel que pour tout sommet x et tout sommet y <u>différent de x</u>, il existe un arc de x vers y. Quel est le nombre d'arcs de $K_n^\star$? Pour un sommet x, calculez $d(x)$, $d^+(x)$,$d^-(x)$.

Soit K_n le **graphe non orienté associé** à $K_n^\star$. Calculez son nombre d'arêtes. Pour chaque sommet x, calculez $d(x)$.

□

Exercice 12 : Dans un congrès, n personnes ($n \geq 2$) se retrouvent. Certaines d'entre elles s'échangent des poignées de main. Montrez qu'alors au moins deux convives ont serré le même nombre de main. Aide : Construisez la liste d'adjacence du graphe des "poignées de main" en supposant que l'énoncé est faux. Vous arriverez alors vite à une contradiction.

□

1.2.2 Fermeture Transitive

Exercice 13 : Dessinez le graphe qui admet pour **matrice (d'adjacence)**

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Calculez ensuite M^2, le produit de M par M, et $M^{[2]}$, le produit booléen de M par M. Interprétez.

□

Exercice 14 :

1. Dessiner le graphe non orienté qui admet pour **matrice (d'adjacence)**

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

2. Calculer M^2, puis M^3, $M + M^2$ (non booléen),
3. Combien y a t'il de chaînes de longueur 2 menant 1 à 2 ? 2 à 4 ? 3 à 4 ?
4. Combien y a t'il de chaînes de longueur ≤ 2 menant 1 à 2 ? 2 à 4 ? 3 à 4 ?
5. Combien y a t'il de chaînes de longueur 3 menant 2 à 3 ? 2 à 1 ?

6. Combien y a t'il de cycles de longueur 3 ayant 3 comme sommet de départ ? 4 comme sommet de départ ?
7. Donner un exemple de cycle de longueur 4. Est-il élémentaire ?
8. Donner tous les cycles de longueur 3. (Y en a t'il 1, 2, 3, ou 6 ?)

□

Exercice 15 : Donnez la **fermeture transitive** des graphes de la figure 1.2 en appliquant un algorithme que vous définirez au préalable.

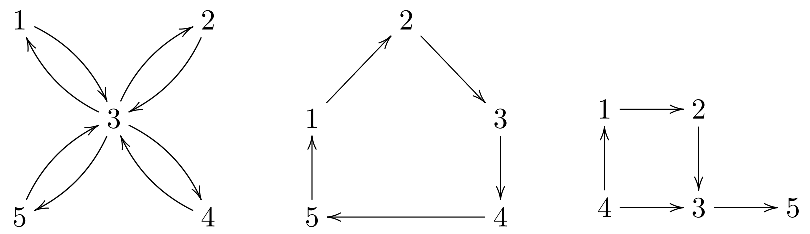


FIG. 1.2 – Fermeture transitive de graphes

□

Exercice 16 :[Projet Matière 2003] Considérons un graphe à n sommets représenté par sa matrice d'adjacence $R[i,j]$ et soit $\mathcal{R}$ la relation associée à cette matrice.

1. Donner un algorithme permettant de calculer le produit booléen de deux matrices
2. Que représente
$$\mathcal{F} = \mathcal{R} \cup \mathcal{R}^2 \cup \ldots \cup \mathcal{R}^n$$
3. En déduire un algorithme permettant de calculer la fermeture transitive de R
4. Soient $\mathcal{S}_1, \mathcal{S}_2, \ldots$ les graphes définis de la manière suivante :
$$\mathcal{S}_1 = \mathcal{R} \cup \mathcal{R}^2$$
$$\mathcal{S}_k = \mathcal{S}_{k-1} \cup \mathcal{S}_{k-1}^2, \text{ pour } k \geq 2$$
En déduire une méthode plus rapide que la précédente pour calculer la fermeture transitive de R.
5. Méthode de Roy-Warshall : la méthode de Roy-Warshall est la méthode la plus rapide connue pour déterminer la **fermeture transitive**, notée T(G) d'un graphe G. En voici la description :

> Soit $\theta_r(G)$ le graphe obtenu à partir du graphe G et du sommet r de la manière suivante :
>
> (a) $\theta_r(G)$ a les mêmes sommets que G. Il possède un arc du sommet i vers le sommet j dans l'un ou l'autre des deux cas suivants :
>   i. il y a un arc dans G de i vers j
>   ii. il y a un arc dans G de i vers r et un arc dans G de r vers j
>
> (b) pour un graphe contenant n sommets calculer $\theta_1(G)$, puis $\theta_2(\theta_1(G))$ puis $\theta_3(\theta_2(\theta_1(G)))$, jusqu'à $\Theta(G) = \theta_n(\ldots(\theta_2(\theta_1(G)))\ldots))$.
>
> (c) Le graphe ainsi obtenu est la fermeture transitive du graphe G.

Construire un algorithme qui calcule la fermeture transitive d'un graphe à l'aide de la méthode de Roy-Warshall.

6. Comparez les nombres d'opérations effectuées par chacun des 3 algorithmes.
7. Appliquez l'algorithme de Roy-Warshall aux deux derniers graphes de la figure 1.2

□

1.2.3 Cycles et Circuits

Exercice 17 : Au 18ème siècle, Léonard Euler aime se promener au printemps (D'où l'expression "prendre Euler") dans la charmante ville de Kőnigsberg (actuellement Kaliningrad, en Russie). Il souhaitait ne passer qu'une seule fois par chacun des sept ponts de cette ville. Qu'en pensez vous ?

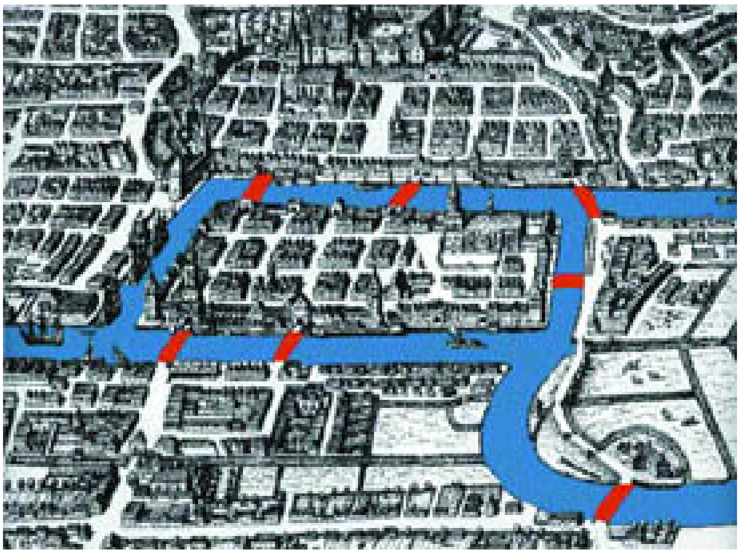


Fig. 1.3 – Ville de Kőnigsberg

□

Exercice 18 : Pour chaque sommet du graphe de la Figure 1.4, énumérez $\Gamma^+(x)$, $\Gamma^-(x)$, $d^+(x)$, $d^-(x)$. Ce graphe contient-il un *circuit hamiltonien* ? Est-il *connexe* ? *fortement connexe* ?

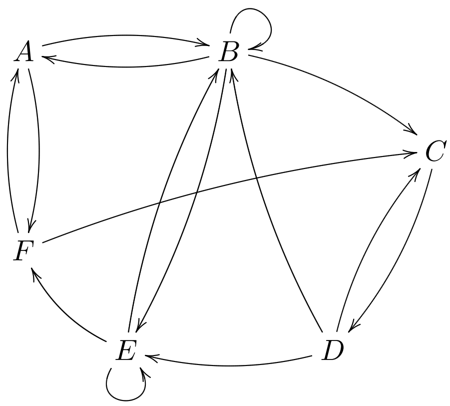


Fig. 1.4 – Circuit hamiltonien

□

Exercice 19 : Les graphes de la figure 1.5 admettent ils des *circuits hamiltoniens* ? des *circuits eulériens* ? des *cycles hamiltoniens* ? des *cycles eulériens* ?

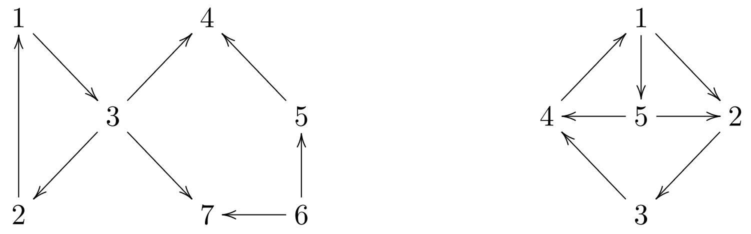


FIG. 1.5 – Circuits hamiltoniens et eulériens

□

Exercice 20 : Le graphe (a) admet-il une *chaîne eulérienne* ? Si oui, déterminez en une. Le *multi graphe* (b) admet-il un *cycle eulérien* ? Si oui, déterminez en un.

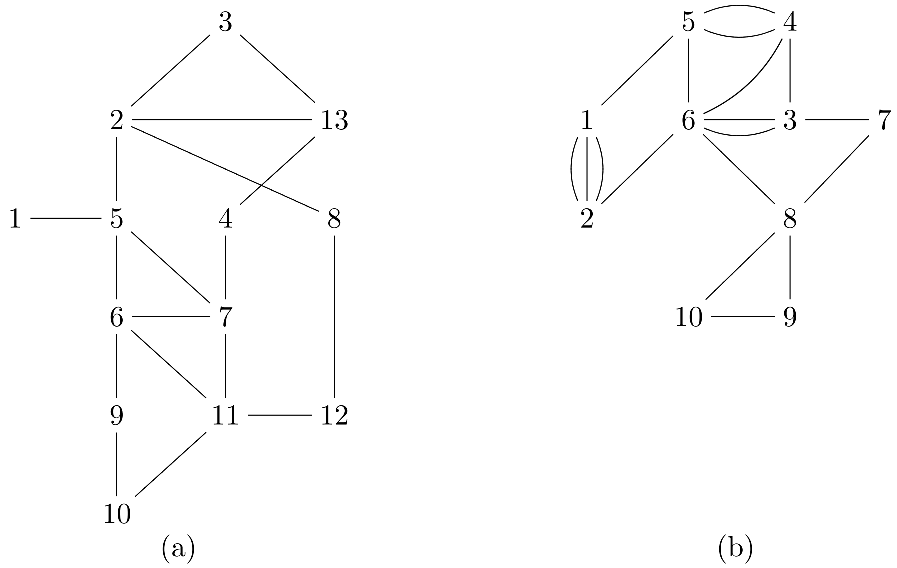


FIG. 1.6 – Chaîne et cycle eulériens

□

Exercice 21 : Peut-on sans relever le crayon tracer une ligne coupant une fois et une seule chaque segment des dessins suivants :

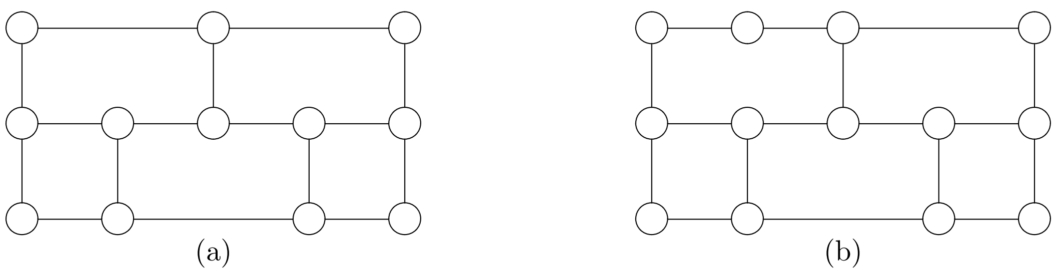


FIG. 1.7 – Tracé de chemin

□

1.2.4 Graphes Isomorphes

Exercice 22 : Vérifiez que les graphes de la figure 1.8 sont **isomorphes**.

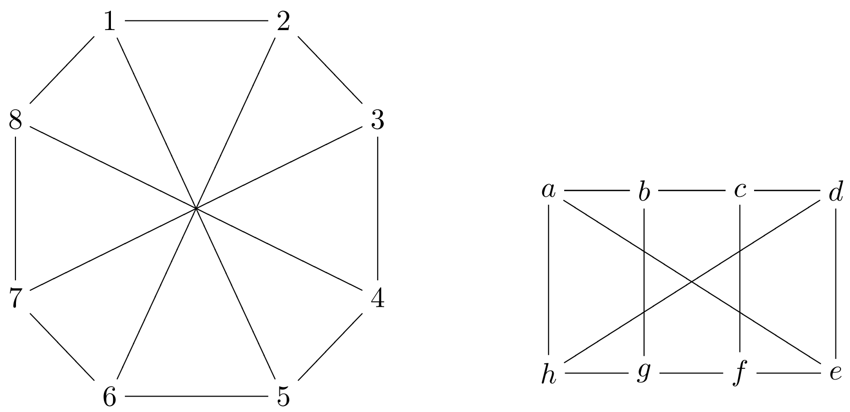
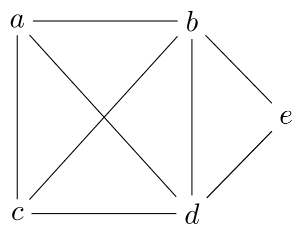


FIG. 1.8 – Graphes isomorphes

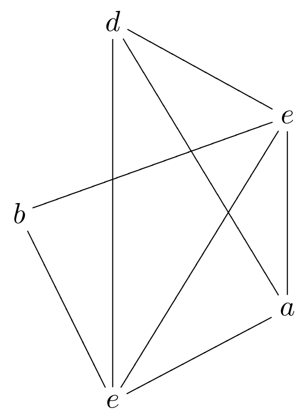
□

Exercice 23 :

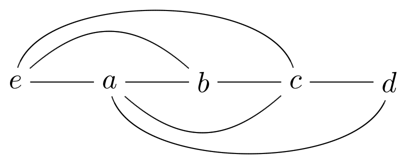
1. Parmi les 4 graphes G_1, G_2, G_3 et G_4 suivant, vous indiquerez lesquels sont isomorphes. Vous justifirez chacune de vos réponses (remarque : l'isomorphie est une relation d'équivalence).



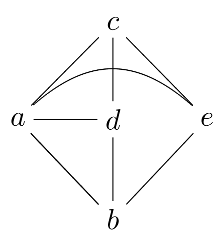
(G_1)



(G_2)



(G_3)



(G_4)

2. Le graphe G_1 admet-il un cycle hamiltonien ? Justifiez.

3. Les graphe G_1, G_2, G_3 et G_4 admettent-ils une chaîne eulérienne ou un chemin eulérien ? Justifiez. Si oui, construisez en un.

□

1.3 Exploration des Graphes

1.3.1 Composantes Connexes

Exercice 24 : Soit $G = (S, A)$ un graphe non orienté. Définissons la relation $\mathcal{R}$ par $x\mathcal{R}y$ ssi $x = y$ ou s'il existe une **chaîne** entre x et y. Montrez que cette relation est une **relation d'équivalence**.

Ses **classes d'équivalence** s'appellent les **Composantes Connexes**.

Étendre cette définition aux graphes orientés. On appelle les classes d'équivalences d'un graphe orienté les **Composantes Fortement Connexes**.

□

Exercice 25 : Combien de **composantes connexes** possède le graphe $G = (S, A)$ de la figure 1.9 ? Même question pour le **sous-graphe** engendré par $S - \{1\}$.

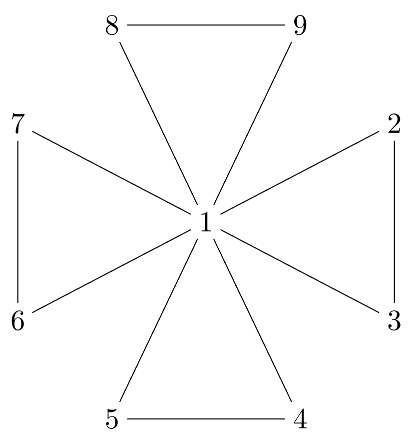


FIG. 1.9 – Sous-graphe et graphe partiel

□

Exercice 26 :

1. Dessinez le graphe G_0 qui admet pour **matrice (d'adjacence)**

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

L'algorithme suivant permet de trouver la **composante fortement connexe** d'un sommet noté x_0.

(a) marquer x_0 avec les signes (+) et (-) ;
(b) marquer du signe (+) tout successeur non encore marqué (+) d'un sommet marqué (+) ;
(c) marquer du signe (-) tout prédécesseur non encore marqué (-) d'un sommet marqué (-) ;
(d) Quand on ne peut plus marquer de sommet, les sommets marqués (+) et (-) constituent la composante fortement connexe de x_0.

2. Utilisez cet algorithme pour trouver toutes les **composantes fortement connexes** du graphe G_0.
3. Expliquez pourquoi un sommet x qui est marqué (+) et (-) appartient à la composante fortement connexe de x_0.
4. Expliquez pourquoi un sommet x qui n'est pas marqué, ou qui est marqué avec un seul signe, soit (+), soit (-) n'appartient pas à la composante fortement connexe de x_0.
5. On pourrait penser que les sommets marqués soit (+), soit (-), soit (+) et (-) constituent la **composant connexe** de x_0 pour le graphe . Donner un exemple de graphe à trois sommets pour lequel ce n'est pas le cas.
6. Le graphe réduit $G_R = (S_R, A_R)$ du graphe G=(S,A) est le graphe orienté défini comme suit : « *Les sommets de G_R sont les composantes fortement connexe de G. Pour C_i et C_j, deux sommets de G_R, (C_i, C_j) est un arc de G_R ssi il existe un arc dans G entre un sommet de C_i et un sommet de C_j* ».

   Tracer le graphe réduit de G_0.
7. Peut-il exister un circuit qui ne soit pas une boucle dans un graphe réduit ? Justifier votre réponse.

□

Exercice 27 : Un graphe $G = (S, A)$ est 2-connexe si pour tout sommet x de S, le **sous-graphe** engendré par $S - \{x\}$ est **connexe**.

Vérifiez que si toute paire de sommets de G appartient à un **cycle élémentaire**, alors G est 2-connexe. Est-ce encore vrai si on supprime “élémentaire” ?

□

Exercice 28 : Démontrez que les 2 propriétés suivantes sont équivalentes.

1. G=(S,A) est **fortement connexe**
2. $\forall X \subset S, X \neq S, X \neq \emptyset, \Gamma^+(X) \not\subset X$

où

$$\Gamma^+(X) = \bigcup_{x \in X} \Gamma^+(x)$$

□

1.3.2 Arbres Recouvrants

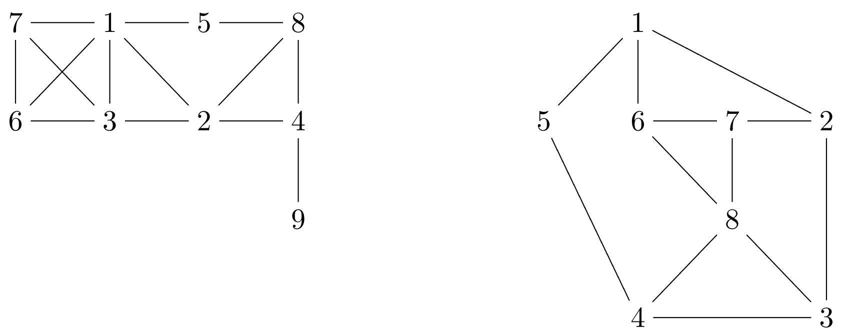
Exercice 29 : Effectuer un parcours en largeur des graphes (a) et (b) de la figure 1.10 en exécutant manuellement l'algorithme 1 en partant du sommet 1.

Vous donnerez à chaque fois l'**arborescence** ainsi obtenue.

Algorithm 1 Algorithme de parcours en largeur d'abord d'un graphe *connexe* G contenant n sommets en partant du sommet i. Cet algorithme utilise une file comme structure de donnée (FIFO). A la fin de l'algorithme $\pi(s)$ est le prédécesseur de s sur le parcours obtenu.

ParcoursLargeur(G,i)

```
F ← ∅
marquer i
enfiler i dans F
while F ≠ ∅ do
   s ← tête de F
   défiler F
   for v ∈ Γ+(s) do
      if v n'est pas marqué then
         π(v) ← s
         marquer v
         enfiler v dans F
      end if
   end for
end while
```



(a) (b)

Fig. 1.10 – Parcours de Graphes

Transformer l'algorithme de manière à utiliser une pile (**Attention** : ne pas empiler inutilement). L'algorithme ainsi obtenu permet d'effectuer un parcours en profondeur. Donner pour chacun des graphes les arborescences obtenues.

□

1.4 Algorithmes sur les Graphes Valués

1.4.1 Recherche du Plus Court Chemin

Algorithme de Bellman

Exercice 30 : La **fonction ordinale** est une numérotation des sommets d'un graphe orienté tel que :

- tous les sommets ont des numéros différents.
- s'il y a un arc d'un sommet de numéro i vers un sommet de numéro j alors $i < j$.

Algorithm 2 Algorithme de Bellman de recherche des plus courts chemins d'un sommet sans prédécesseur noté a vers tous les autres sommets dans un graphe G sans circuit contenant n sommets. A la fin de l'algorithme, $\lambda(i)$ est la valeur du plus court chemin de 1 à i et $\pi(i)$ est le prédécesseur de i sur un chemin minimal

Bellman(G, a)

numérotez les sommets selon une fonction ordinale en numérotant 1 le sommet a
$\lambda(1) \leftarrow 0$
$\pi(1) \leftarrow null$
for $i = 2$ à n **do**
  $\lambda(i) \leftarrow +\infty$
  $\pi(i) \leftarrow null$
end for
for $i = 2$ à n **do**
  for all $j \in \Gamma^-(i)$ **do**
    $v \leftarrow \lambda(j) + w(j, i)$
    if $v < \lambda(i)$ **then**
      $\lambda(i) \leftarrow v$
      $\pi(i) \leftarrow j$
    end if
  end for
end for

1. Une telle fonction n'existe que pour les graphes sans circuits. Dans ce cas, la méthode suivante permet de donner une telle numérotation.

   (a) numéroter 1 un sommet sans prédécesseur ; faire $k \leftarrow 2$
   (b) numéroter k un sommet sans prédécesseur ou un sommet dont tous les prédécesseurs sont déjà numérotés.
   (c) faire $k \leftarrow k + 1$ et recommencer l'étape précédente jusqu'à ce que tous les sommets soient numérotés.

   Appliquer cet algorithme au graphe de la figure suivante 1.11.
2. Existe-t'il plusieurs fonctions ordinales pour ce graphe ?
3. Appliquer cet algorithme aux graphes de la figure 1.13. Que constatez-vous ?
4. Montrez le résultat suivant : dans un graphe sans circuit, il existe au moins un sommet sans prédécesseur.
5. Utilisez le résultat précédent pour monter que si un graphe est sans circuit, alors il admet une numérotation topologique.

□

Exercice 31 : Exécutez l'algorithme 2 sur le graphe de la figure 1.11.

□

Exercice 32 : Modifier l'algorithme de Bellman pour trouver les plus longs chemins. Appliquer au graphe précédent.

□

Exercice 33 :[Chemins et chevaliers] En l'an de grâce 1479, le sire Gwendal, paludier à Guérande, désire aller vendre sa récolte de sel à l'une des grandes foires du Duché. Il connaît les gains qu'il

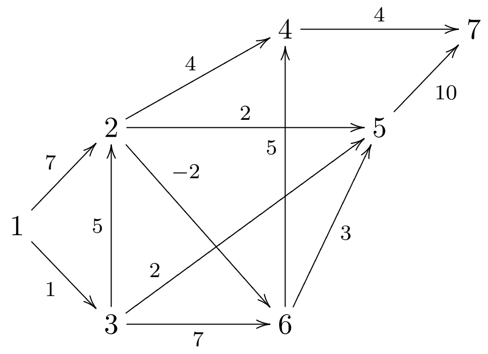


FIG. 1.11 – Graphe valué sans circuit

pourra réaliser dans chacune des foires (Table 1.1) , mais ceux-ci sont diminués des octrois qu'il devra acquitter le long du chemin emprunté pour s'y rendre (voir la table 1.2 et le graphe 1.12). Il s'agit de savoir à quelle foire, et par quel chemin le paludier, doit se rendre de façon à réaliser le plus grand bénéfice possible.

1. Montrer que ce problème est équivalent à un problème de recherche de chemin de valeur **minimale** dans un graphe valué. Vous prendrez soin de valuer chaque arc.

2. Parmi les deux algorithmes de recherche de chemins optimaux que vous avez vu en cours, l'un est applicable et l'autre non. Dites pourquoi pour chacun.

3. Utilisez l'algorithme approprié pour résoudre le problème. Vous présenterez le déroulement de l'algorithme comme il a été vu en cours et vous donnerez l'arborescence des chemins de valeur minimale.

4. L'année suivante les octrois sont augmentés d'une même valeur dans chacune des villes. Le trajet optimal du paludier est-il modifié ? Vous argumenterez précisément votre réponse. En particulier, vous donnerez un exemple de graphe à 3 sommets dans lequel l'ajout d'une unité sur chacun des arcs modifie le chemin de valeur minimal entre deux sommets.

foires	Rennes	Laudéac	Pontivy	Lorient
gains (en écus)	550	580	590	600

TAB. 1.1 – Tableaux des gains (en écus) dans les différentes foires.

villes	1	2	3	4	5	6	7	8	9	10	11	12	13	14
octrois (en écus)	10	12	15	5	15	10	3	10	5	20	4	5	20	7

TAB. 1.2 – Tableaux des octrois (en écus)

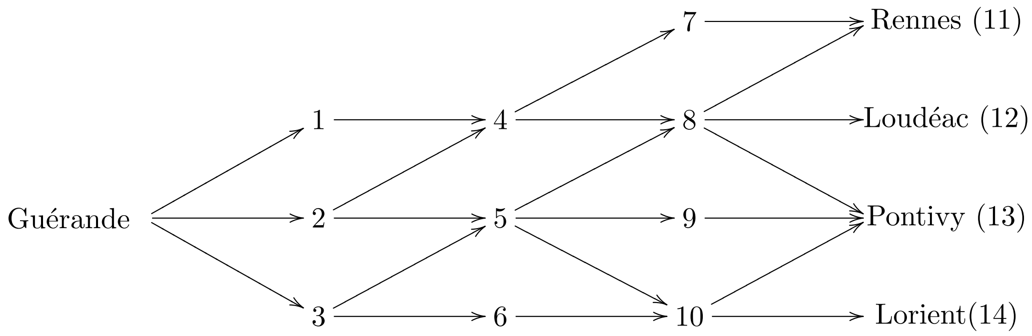


FIG. 1.12 – Graphe des chemins possibles du domicile du paludier aux différentes foires

□

Algorithme de Dijkstra

Algorithm 3 Algorithme de Dijkstra de recherche des plus courts chemins du sommet 1 vers tous les autres sommets dans un graphe orienté $G = (S, A)$. A la fin de l'algorithme, $\lambda(i)$ est la valeur du plus court chemin de 1 à i et $\pi(i)$ est le prédécesseur de i sur un chemin de poids minimal

Dijkstra(G) $[G = (S, A)]$

$\lambda(1) \leftarrow 0$
$\pi(1) \leftarrow null$
marquer le sommet 1
for $i = 2$ à n **do**
    $\lambda(i) \leftarrow +\infty$
    $\pi(i) \leftarrow null$
end for
while tous les sommets ne sont pas marqués **do**
    $m \leftarrow +\infty$
    for all $j \in S$ and j non marqué **do**
        if $\lambda(j) \leq m$ **then**
            $i \leftarrow j$
            $m \leftarrow \lambda(j)$
        end if
    end for
    marquer i
    for all $j \in \Gamma^+(i)$ and j non marqué **do**
        $v \leftarrow \lambda(i) + w(i, j)$
        if $\lambda(j) > v$ **then**
            $\lambda(j) \leftarrow v$
            $\pi(j) \leftarrow i$
        end if
    end for
end while

Exercice 34 : Appliquez l'algorithme 3 sur les graphes (a) puis (b) de la figure 1.13. Que remarquez vous sur le deuxième graphe ?

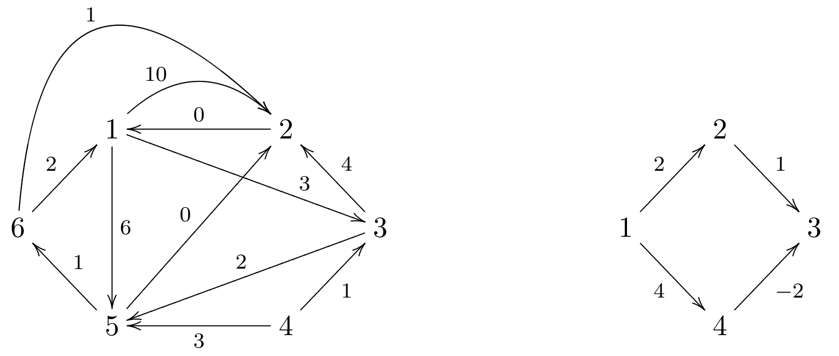


FIG. 1.13 – Deux graphes valués

□

Exercice 35 : En utilisant les données de la table 1.3 trouver à l'aide de l'algorithme de Dijkstra le coût minimum pour relier l'agence de Neuilly à une agence quelconque (ne pas faire de tableau).

□

1.4.2 Arbres Recouvrants Minimaux

Exercice 36 : Une banque Parisienne désire installer au moindre coût un réseau de transmission de données entre son siège situé à la bourse de Paris et 7 de ces succursales. Le coût d'une ligne entre deux agences est donnée par le tableau suivant :

	Bourse	Opéra	Etoile	République	St-Lazarre	Louvre	Neuilly
Opéra	5						
Etoile	18	17					
République	9	11	27				
St-Lazarre	13	7	23	20			
Louvre	7	12	15	15	15		
Neuilly	38	38	20	40	40	35	
Châtelet	22	15	25	25	30	10	45

TAB. 1.3 – Coût d'installation d'un réseau de transmission

Appliquer Prim pour minimiser le coût du projet. Même chose avec Kruskal.

□

1.4.3 Ordonnancement

Exercice 37 :

1. On désire réaliser un projet de mathématiques divisé en **tâches** soumises à des contraintes de succession (certaines tâches ne peuvent commencer que lorsque d'autres sont achevées), et ce dans les plus brefs délais possibles. Nous choisissons de modéliser ce problème par le graphe valué **potentiels-tâches** $G = (S, A)$. Celui-ci se construit comme suit :

(a) A chaque tâche, associer un sommet de G.
(b) Mettre un arc du sommet i vers le sommet j si la tâche i doit précéder la tâche j.
(c) Supprimer les raccourcis.
(d) Ajouter 2 sommets α et ω correspondant à 2 tâches fictives de durée nulle. α est la tâche **début** reliée aux sommets sans prédécesseurs et ω est la tâche **fin** reliée aux sommets sans successeurs du graphe.
(e) Valuer chaque arc (i, j) par la durée d_i de la tâche i jusqu'à ce que tous les sommets soient numérotés.

Donnez le graphe potentiels-tâches pour des tâches énumérées dans le tableau 1.4.

2. Supposons qu'un graphe potentiels-tâches contienne un circuit. Le projet est-il réalisable ?

   Dans la suite nous supposons que ce graphe ne contient pas de circuit.

3. Montrez qu'un graphe potentiels-tâches est connexe.

   Nous supposons que le projet débute à la date 0. Notons t_i la **date au plus tôt** à laquelle peut commencer la tâche i. $t_\alpha = 0$ et t_ω est la durée du projet.

4. Donnez les dates au plus tôt pour tous les sommets du graphe construit à la question 1.

5. Pour un sommet quelconque i, exprimez t_i en fonction des valeurs t_j et aux durées d_j des tâches j qui précèdent i.

6. Interprétez alors le problème du calcul des valeurs t_i en terme de problème de chemins optimal. Connaissez vous un algorithme applicable pour résoudre ce problème ?

   La **date au plus tard** T_i d'une tâche i est la date au plus tard à laquelle peut commencer une tâche i sans augmenter la durée t_ω du projet. En particulier $T_\omega = t_\omega$ et $T_\alpha = 0$.

7. Donnez les dates au plus tard pour tous les sommets du graphe construit à la question **1**.

8. Pour un sommet quelconque i, exprimez T_i en fonction de t_ω et de la valeur d'un plus long chemin de i à ω.

9. Quel algorithme pouvez vous utiliser pour calculer les T_i ?

   La **marge** m_i de la tâche i est la différence entre la date au plus tard et la date au plus tôt. $m_i = T_i - t_i$. Une tâche de marge nulle est une tâche **critique**. Un **chemin critique** est un chemin de α à ω n'empruntant que des tâches critiques.

10. Donnez un chemin critique pour le problême de la figure 1.4. De combien de jours au maximum peut-on retarder la tâche C sans retarder l'ensemble du projet ?

11. Montrez qu'un graphe potentiels-tâches possède toujours au moins un chemin critique.

□

tâches		durée en jours	tâches antérieures
A	comprendre le sujet	2	
B	choisir les structures de données	3	A
C	écrire et implémenter l'algorithme 1	1	A B
D	écrire et implémenter l'algorithme 2	2	A B
E	écrire et implémenter l'algorithme 3	3	A B C D
F	tester le programme	2	C D E
G	faire la partie théorique	3	A
H	rédiger le rapport	3	C D E G

TAB. 1.4 – Les tâches pour réaliser le projet de mathématiques sont énumérées dans le tableau ci-dessus où figurent également les contraintes d'antériorité et la durée des tâches

Exercice 38 : Donnez le graphe potentiels-tâches du problème suivant. Donnez les dates au plus tôt, au plus tard et le ou les chemins critiques.

tâches	durée en jours	tâches antérieures
A	2	
B	3	A
C	1	
D	2	
E	3	B C
F	3	A B C
G	3	A
H	3	D E G
I	2	B D F H

□

1.5 Flots

Exercice 39 :[Capacité d'un réseau routier] Avant d'établir un projet de construction d'autoroute, on désire étudier la capacité d'un réseau routier reliant la ville s à la ville p représenté par le graphe 1.14. Pour cela, on a évalué pour chaque route le nombre maximal de véhicules qu'elle peut écouler par heure. Ces évaluations sont indiquées en centaines de véhicules par heure sur les arcs du graphe.

On désire connaître le débit maximal de véhicules susceptibles de s'écouler de la ville s à la ville p.

1. Vérifier que ce problème est équivalent à trouver un flot de valeur maximale dans un réseau de transport $R = (S, A, s, p, c)$ que vous préciserez.
2. Soit $(X_1, \bar{X_1})$ la coupe définie par $X_1 = \{s\}$. Quelle est sa capacité ?
3. Soit $(X_2, \bar{X_2})$ la coupe définie par $X_2 = S - \{p\}$. Quelle est sa capacité ?
4. Que peut-on en déduire quant au débit maximal ?
5. Déterminez un flot complet pour ce réseau.
6. Déterminez un flot optimal pour ce réseau.
7. Quel est le débit maximal de véhicules susceptibles de s'écouler de la ville 1 à la ville 2 ?
8. Donnez une coupe de capacité minimale.

9. Un stagiaire remarqua que le débit maximal entre les villes s et p prédit pour le réseau routier, à savoir 2000 véhicules par heure, surestimait largement ses observations effectuées sur le terrain. Ses observations mesuraient un débit légèrement inférieur à 1700 véhicules par heure. Il se rendit compte que le débit horaire du réseau urbain des villes traversées n'avait pas été pris en compte.

   Le débit horaire durant la traversée des villes est donné par le tableau ci-dessous en centaines de véhicules par heure.

villes	a	b	c	d	e	f	g
débits	6	7	8	6	6	5	9

   Il eut alors l'idée "d'éclater" en 2 sommets chacun des sommets correspondant aux villes traversées, afin d'y faire figurer les contraintes de débit urbain.

   Donnez le réseau ainsi obtenu et vérifiez que le flot maximal dans ce réseau est de 1700 véhicules par heure.

□

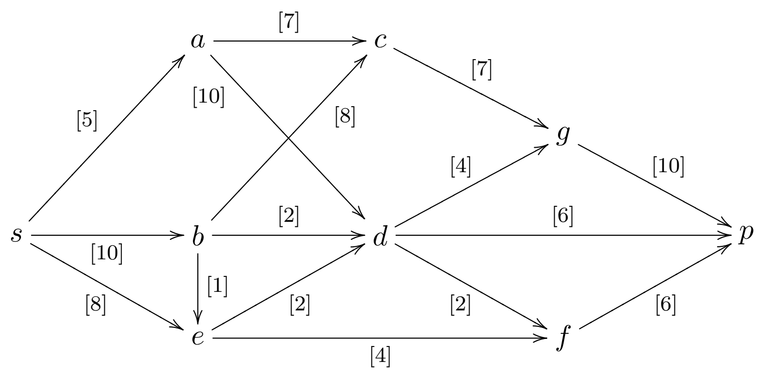


FIG. 1.14 – Réseau de transport routier

Exercice 40 :[amélioration d'un flot]

Trois villes J,K,L sont alimentées en eau grâce à quatre réserves A,B,C,D (nappes souterraines, châteaux d'eau, usines de traitement ...). Les réserves journalières disponibles sont de 15 milliers de mètres cubes pour A, de 10 pour B, de 15 pour C et de 15 pour D. Le réseau de distribution comprenant aussi bien des aqueducs romains que des canalisations récentes, peut être schématisé par le graphe de la figure 1.15. Les débits maximaux sont indiqués sur chaque arc en milliers de mètres cubes par jour.

Ces trois villes en pleine évolution désirent améliorer leur réseau d'alimentation afin de satisfaire des besoins futurs plus importants. Une étude a permis de déterminer les demandes journalières maximales à prévoir. Elles sont de 15 milliers de m^3 pour J, 20 milliers de m^3 pour K, et 15 milliers de m^3 pour L.

1. Déterminez la valeur du flot maximal pouvant passer dans le réseau actuel et donner la coupe minimale correspondante. Vous commencerez par déterminer un flot complet.
2. La valeur de ce flot est jugée nettement insuffisante, aussi le conseil intercommunal décide-t-il de refaire les canalisations (A,E) et (I,L).Déterminez les capacités à prévoir pour ces deux canalisations et la valeur du nouveau flot optimal.
3. Devant l'importance des travaux, le conseil intercommunal décide de ne pas refaire les deux canalisations en même temps. Dans quel ordre doit-on entreprendre leur réfection de façon à augmenter, après chaque tranche de travaux, la valeur du flot optimal passant dans le réseau ?

□

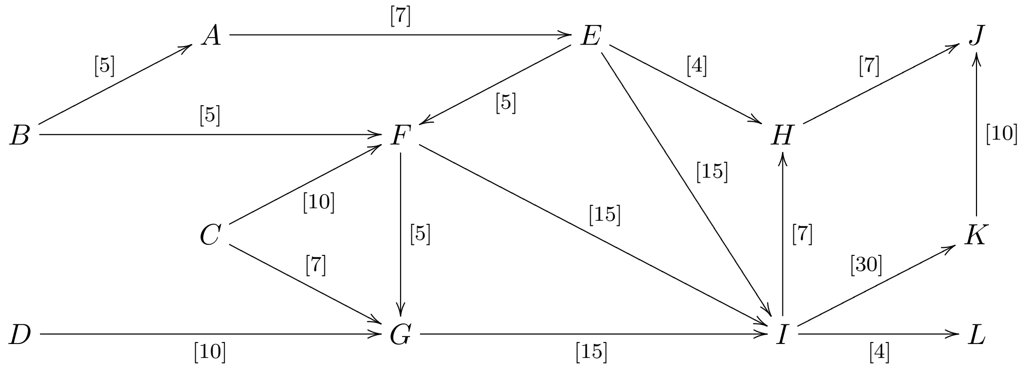
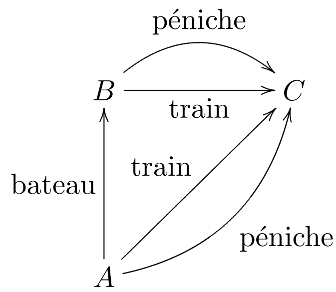


FIG. 1.15 – Réseau de distribution d'eau

Exercice 41 :[flot dynamique]

Du sable est disponible au port A, situé à l'estuaire d'un fleuve. Il doit être acheminé vers une usine de fabrication de verre située en C. Le transport peut s'effectuer directement de A à C, ou en passant par le port B. Il peut s'effectuer selon les 4 modalités suivantes



1. par train de A à C
2. par péniche de A à C
3. par bateau de petit tonnage de A à B puis par train de B à C
4. par bateau de petit tonnage de A à B puis par péniche de B à C

Au début de chaque heure, il est possible de faire partir un ou plusieurs convois. Les capacités de ces convois sont donnés dans le tableau ci-dessous (en tonnes de sable).

	train	péniche	bateau
A C	200	300	
B C	100	300	
A B			1200

Les capacités de stockage sur le quai de chargement est limité à 400 tonnes en B (initialement le stock en B est nul).

Les temps de transport, en heures, compte tenu des temps de chargement et de déchargement, sont les suivants :

	train	péniche	bateau
A C	6	7	
B C	3	5	
A B			4

Le tonnage de sable en A sera considéré comme illimité.

Notre objectif est de déterminer la quantité maximale de sable qu'il est possible d'acheminer de A vers C en 8 heures et quels moyens de transport doivent alors être utilisés.

Tracez le graphe représentant les liaisons de A vers C. Pour éviter d'obtenir un multigraphe, créez deux sommets fictifs T et P représentant les transports par train et par péniche respectivement.

Valuez les arcs de ce graphe par des couples de la forme (k,t) où k est la capacité (éventuellement infinie) et t le temps de parcours (éventuellement nul).

Créez un réseau de transport associé au graphe précédent de la manière suivante :

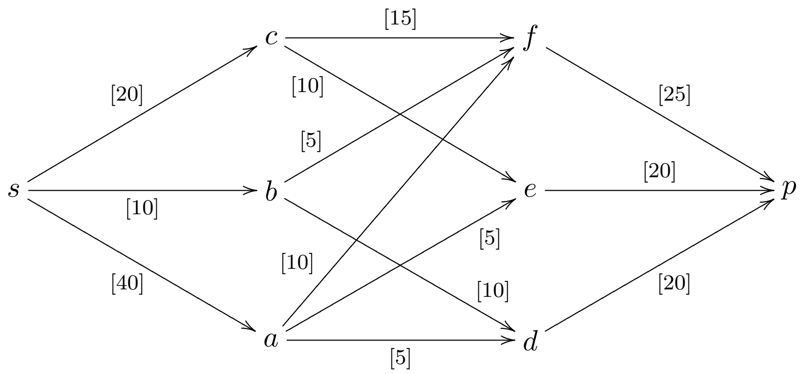
1. "éclatez" chaque sommet x du graphe précédent en plusieurs sommets $x_0, x_1, x_2, ...$ représentant le sommet x aux temps 0,1,2,... Le transport doit être étudié sur une durée de 8 heures donc la source du réseau est le sommet A_0 et le puits le sommet C_8. En tenant compte des temps de transport, vous constaterez qu'il n'est utile de conserver que 13 sommets.
2. créez un arc (x_i, y_j) chaque fois qu'un chargement parti de x au temps i peut arriver en y au temps j. La capacité de cet arc est la capacité indiquée sur le graphe précédent. Ajouter les arcs de la forme e (B_i, B_{i+1}) représentant les quantités stockées en B entre les temps i et i+1 et valuer ces arcs par la capacité de stockage.

Déterminez un flot maximal dans le réseau de transport ainsi obtenu. Vous déterminerez d'abord un flot complet.

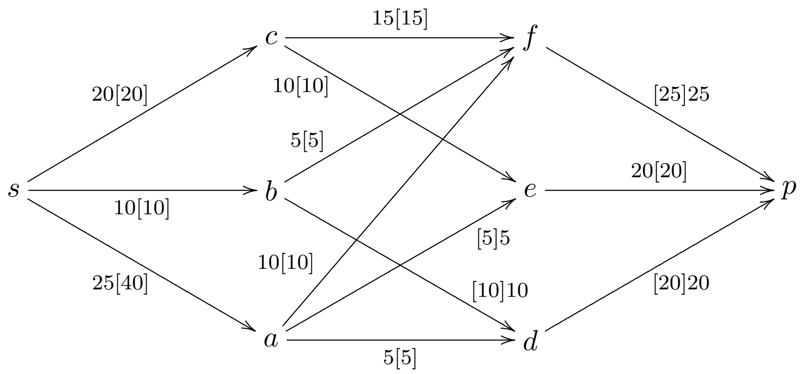
En déduire une solution du problème initial.

□

Exercice 42 : On considère le réseau de transport R ci-dessous. La source est le sommet **s**, le puits est le sommet **p** et les capacités sont indiquées entre crochets.



1. Donnez un flot **complet** pour ce réseau. Vous choisirez les sommets dans l'ordre alphabétique impérativement.
2. La figure ci-dessous présente un flot pour le réseau R.

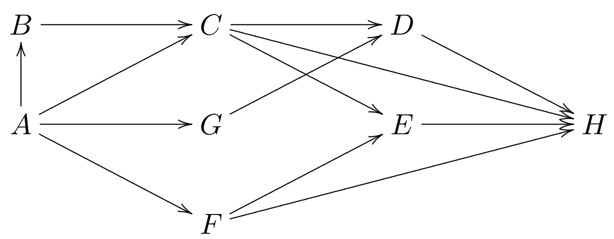


3. Ce flot est-il complet ? Justifiez votre réponse. Est-il possible que sa valeur soit différente de celle trouvée en **1** ?

4. En partant du flot présenté ci-dessus, donnez un flot de valeur maximale. Quelle est cette valeur.
5. Donnez une coupe de capacité minimale.
6. Donnez un flot de valeur maximal différent de celui trouvé en **3**.

□

Exercice 43 : On considère le graphe orienté ci-dessous



On s'intéresse aux chemins d'origine A et d'extrémité H. Deux tels chemins sont dits **disjoints au sens des arcs** s'il n'ont aucun arc commun.

1. Donnez le nombre maximal de chemins de A à H deux à deux disjoints au sens des arcs dans ce graphe.
2. Montrez que ce nombre est la valeur d'un flot maximal dans le réseau de transport déterminé par le graphe ci-dessus, de source A et de puits H, pour lequel vous indiquerez les valeurs à donner aux capacités.

   Deux chemins d'origine A et d'extrémité H sont **disjoints au sens des sommets** si leurs seuls sommets communs sont A et H.
3. Donnez le nombre maximal de chemins de A à H deux à deux disjoints au sens des sommets dans ce graphe.
4. Montrez que ce nombre est la valeur d'un flot maximal dans un réseau de transport à déterminer.

   Aide : pensez à éclater chaque sommet (sauf A et H) en deux sommets.
5. donnez alors un flot de valeur maximale et une coupe de capacité minimale.

□

Chapitre 2

Théorie des Langages

Exercice 1 - Quelques opérations simples entre langages.
Soient les langages $\Sigma = \{a, b, c\}$, $L1 = \{cab, ba\}$ et $L2 = \{aa, ba, \varepsilon\}$.
Décrire chacun des langages suivants :

$$L1 \cap L2, \quad L1 \cup L2, \quad L1 \setminus L2, \quad L2 \setminus L1, \quad L1 \cdot L2, \quad L2 \cdot L1,$$
$$L1^2, \quad L2^2, \quad L2^*, \quad L2^+, \quad \Sigma^*.$$

Exercice 2 - Exercice simple.

1 - Donner des exemples de langages finis.

2 - Soit L un langage... c'est-à-dire un ensemble de mots. On rappelle la définition inductive suivante :

$$L^0 = \{\varepsilon\} \quad \text{et pour } \ n \geq 0, L^{n+1} = LL^n.$$

Soit Σ un alphabet, autrement dit, Σ est un langage dont chaque mot est de longueur 1. Montrer par récurrence que Σ^n est l'ensemble des mots de longueur n construits sur Σ. Rappeler comment Σ^* et Σ^+ peuvent être écrits comme réunion de certains Σ^n.

3 - **3.1 -** Soit le langage $L = \{a\}$.
Quel est le nombre de mots du langage L^{10} ? du langage L^i ? Quel est le nombre de mots du langage $\bigcup_{1 \leq i \leq 10} L^i$ où $\bigcup$ désigne l'union des langages ? De même, déterminer le nombre de mots de $\bigcup_{0 \leq i \leq n} L^i$, puis le nombre de mots de $\bigcup_{1 \leq i \leq n} L^i$.

3.2 - Répondre aux mêmes questions pour $L = \{\varepsilon, a\}$.

3.3 - Idem pour $L = \{a, aa\}$.

3.4 - Idem pour $L = \{b, aa\}$.

Exercice 3 - Intersection, concaténation.
Soit l'alphabet $\Sigma = \{a, b\}$ et soient les langages :

$$L1 = \{a^n b^p : n, p \in \mathbb{N}\} \quad L2 = \{a^n b^n : n \in \mathbb{N}\}$$

et

$$L3 = \{a^n : n \in \mathbb{N}\} = a^* \quad L4 = \{b^n : n \in \mathbb{N}\} = b^*$$

On rappelle que, par convention, on pose $a^0 = b^0 = \varepsilon$.

1 - Donner des mots de chacun des langages.

2 - Déterminer l'intersection $L1 \cap L2$.

3 - Déterminer les mots de $L1$ qui ne sont pas dans $L3$ puis ceux de $L2$ qui ne sont pas dans $L4$ (en d'autres termes, déterminer $L1 \setminus L3$ et $L2 \setminus L4$).

4 - Pourquoi peut-on écrire que $L1 = L3 \cdot L4$, c'est-à-dire que $\{a^n b^p : n, p \in \mathbb{N}\} = a^* b^*$?
Peut-on aussi écrire que $L2$ – c'est-à-dire $\{a^n b^n : n \in \mathbb{N}\}$ égale aussi $a^* b^*$?

Exercice 4 - Exemples de langages.

1 - Décrire le langage dont les mots sont les éléments de $\mathbb{N}$ (plus précisément, on demande de donner une expression régulière du langage des écritures en base dix d'entiers naturels).

2 - Décrire le langage dont les mots sont les éléments de $\mathbb{Z}$.

3 - Soit $B = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ et $A = \{0\} \cup B$ deux alphabets.

3.1 - Quel est le langage $B \cdot A^* \cdot \{0\} \cup \{0\}$?

3.2 - Écrire le langage des multiples entiers naturels de 5, puis celui des entiers naturels strictement inférieurs à 1000.

Exercice 5 - Quelques expressions régulières.
Soit l'alphabet $\Sigma = \{a, b\}$. Montrer les résultats suivants :

1 - $\Sigma^* = \Sigma^*\Sigma^*$,

2 - $\Sigma^* = \Sigma^* \{ab\} \Sigma^* \cup b^*a^*$,

3 - $\Sigma^* = (a^* \cup \{b\})^*$,

4 - $\Sigma^* = (a^* \cup b^*)^*$,

5 - $\Sigma^* = (a^*b^*)^*$,

6 - $(ab)^*a = a(ba)^*$,

7 - $\varepsilon^* = \{\varepsilon\}$ (que vaut $\emptyset^*$?),

8 - $b^* \cup ab^* = \{\varepsilon, a\} b^*$,

9 - $(a^*)^* = a^*$,

10 - $(\{aa\} \cup (ab)^*)^* \neq \{a, ab\}^*$,

11 - $\Sigma^* \neq a^*b^*$.

Exercice 6 - Quelques expressions régulières.
Soit l'alphabet $\Sigma = \{a, b, c\}$. Pour chacun des langages suivants, proposer une expression régulière :

1 - Langage des mots sur Σ commençant par a,

2 - Langage des mots sur Σ commençant et se terminant par a,

3 - Langage des mots sur Σ commençant et se terminant par la même lettre,

4 - Langage des mots sur Σ commençant et se terminant par des lettres distinctes,

5 - Langage des mots sur Σ contenant au moins un a,

6 - Langage des mots sur Σ contenant exactement un a,

7 - Langage des mots sur Σ ne contenant aucun a,

8 - Langage des mots sur Σ contenant un nombre pair de a,

9 - Langage des mots sur Σ contenant un nombre impair de a,

Exercice 7 - Concaténation.

1 - Soit Σ un alphabet, « $\cdot$ » l'opération de concaténation définie sur Σ^*, et $(\Sigma^*, \cdot)$ le monoïde libre (Un *monoïde* est un ensemble muni d'une loi interne et associative, un *monoïde libre* est un monoïde admettant un élément neutre) des mots sur Σ.

1.1 - Rappeler pourquoi $(\Sigma^*, \cdot)$ est un monoïde libre.

1.2 - Montrer que $(\mathbb{N}, +)$ est un monoïde libre.

1.3 - Montrer que l'application « longueur » est un homomorphisme de monoïde libre. Cette application est définie par :

$$|\ldots| : \Sigma^* \to \mathbb{N}, m \mapsto |m|$$

où $|m|$ désigne la longueur du mot m, c'est-à-dire le nombre de lettres de m.

2 - À quelle condition sur Σ l'opération de concaténation est- elle commutative dans Σ^* ?

3 - Montrer que la proposition suivante est fausse : (quels que soient $L1$, $L2$, $L3$ trois langages construits sur Σ, on a : $L1 \cdot L2 = L1 \cdot L3 \Rightarrow L2 = L3$).

4 - Montrer en revanche que la proposition suivante est vraie : (quels que soient $m1$, $m2$, $m3$ trois mots de Σ^*, on a : $m1 \cdot m2 = m1 \cdot m3 \Rightarrow m2 = m3$).

Exercice 8 - Équations entre langages.
Soit Σ un alphabet et soient B, C, D et L des langages sur Σ.

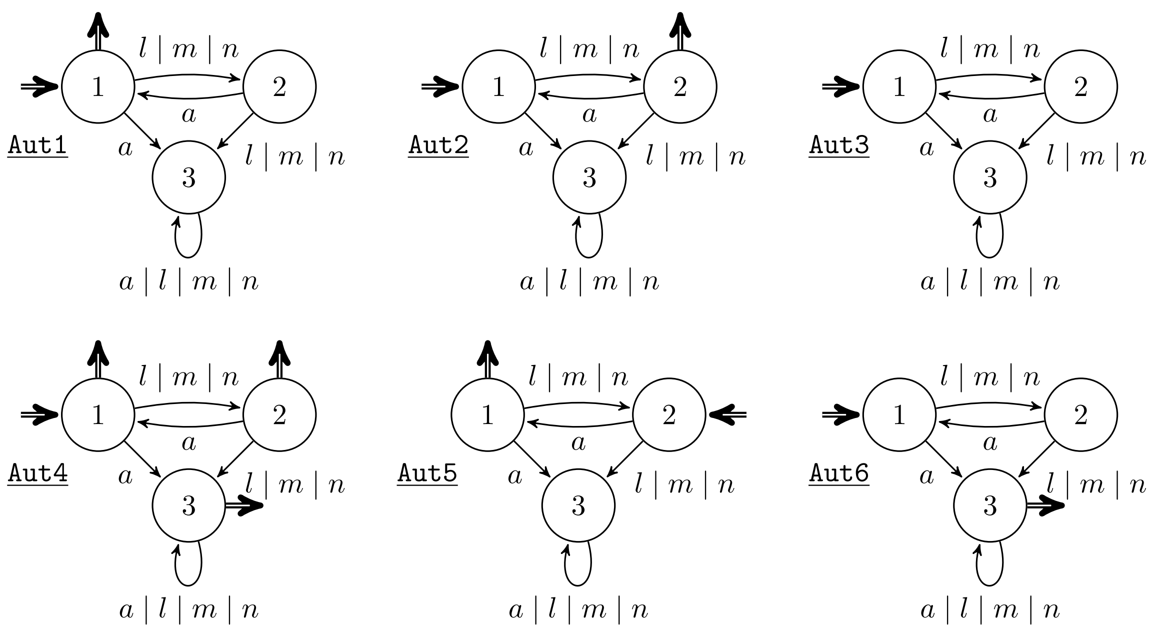
1 - Montrer que $B \cdot (C \cup D) = (B \cdot C) \cup (B \cdot D)$.

2 - Montrer que $B \cdot (C \cap D) \subseteq (B \cdot C) \cap (B \cdot D)$.

3 - Soit $B = \{a, b\}^*$. Si $C = \{\varepsilon, a\}$ et si $D = \{\varepsilon, b\}$, est-il exact que $BC = BD$? A-t-on $(BC = BD) \Rightarrow (C = D)$?

4 - Que peut-on dire de $\{\varepsilon\} \cdot L$? De $(\{\varepsilon\} \cup L)^*$? De $(L^*)^*$? Et de $\{\varepsilon\} \cup (L \cdot L^*$? Est-il vrai que $L^+ = L^* \setminus \{\varepsilon\}$?

Exercice 9 - Quelques AFdc sur l'alphabet $\Sigma = \{a, l, m, n\}$.
On considère les automates finis représentés par :

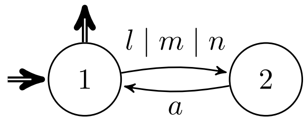


1 - On considère Aut1.

1.1 - Comment sont lus les mots : $lama, maman, laamll, \varepsilon$? Quels sont ceux qui sont acceptés ?

1.2 - Mêmes questions pour l'automate :

Aut1bis



(attention, il s'agit cette fois d'un AF déterministe mais non complet).

2 - Quel est le langage reconnu par Aut1 et par Aut1bis ? Essayez de justifier votre réponse...

3 - Pour chacun des 5 automates Aut2 à Aut6, donner une expression (si possible régulière) du langage qu'il reconnaît.

4 - Proposer un automate qui reconnaît le même langage que Aut3, mais qui soit plus simple. Même question pour Aut4.

Exercice 10 - Constructions d'automates sur $\Sigma = \{a, b, c\}$. Proposer des automates (AFdc, puis, si c'est plus simple, AFd) reconnaissant chacun des langages suivants :
Σ^*, $\{\varepsilon\}$, Σ, Σ^2, langage des mots de longueur au plus 2, langage des mots de longueur 2 ou plus, $\emptyset$, langage des mots commençant par a, langage des mots sans voyelle, langage des mots ne contenant pas deux lettres différentes, langage des mots ne contenant pas deux fois la même lettre.
Pour chacun de ces langages, proposer une expression régulière.

Exercice 11 - Un AFnd.
Soit $\Sigma = \{a, b, c\}$, on désire construire un AF, le plus simple possible, qui reconnaisse le langage des mots sur Σ qui contiennent le mot *bac* **et** se terminent par a.

1 - Donner une expression régulière de ce langage.

2 - Dessiner un AF, le plus simple possible, qui reconnaisse ce langage.

3 - Indiquer comment votre AF lit chacun des mots suivants (plus précisément, on demande d'indiquer **toutes** les lectures de chaque mot) :

$$abaca, \quad abacb, \quad abacbaca, \quad bcaa, \quad bcab, \quad \varepsilon.$$

4 - Donner la « définition mathématique » complète de votre AF.

5 - Par le procédé décrit en cours, transformer votre AF en un AFDC qui reconnaît le même langage.

6 - Indiquer comment votre AFDC lit chacun des mots proposés à la question **2**.

Exercice 12 - Un autre AFnd.
Reprendre les questions de l'exercice **2** pour le langage sur $\Sigma = \{a, b, c\}$ des mots qui contiennent le mot *bac* **ou** se terminent par a.

Exercice 13 - Des automates finis simples.
Soit $\Sigma = \{a, b\}$ un alphabet et soient les langages suivants : $L1 = \{a^n : n \geq 0\}$, $L2 = \{b^n : n \geq 0\}$, $L3 = L1L2$.

1 - Déterminer un automate fini dont le langage soit $L1$.

2 - Le langage $L3$ est-il le langage d'un automate fini ? Si le réponse est oui, construire un tel automate.

3 - Construire une grammaire dont le langage soit $L3$.

Exercice 14 - Des automates finis simples.
Soit $\Sigma = \{a, b, c\}$ un alphabet. Déterminer un automate fini associé aux langages suivants construits sur Σ :

1 - le langage $L1$ des mots contenant au moins la suite de 3 lettres *abc*,

2 - le langage $L2$ des mots contenant exactement 4 b,

3 - le langage $L3$ des mots commençant par (a ou c), et se terminant par b.

Exercice 15 - Transformations d'AFnd en AFdc.

1 - Soit l'automate $(\{a, b\}, \{0, 1, 2\}, \{0\}, \{2\}, \Theta)$ où Θ est donnée par sa table de transition :

	a	b
$-$ 0	0 | 1	1
1		2
2 $+$	0 | 1 | 2	1

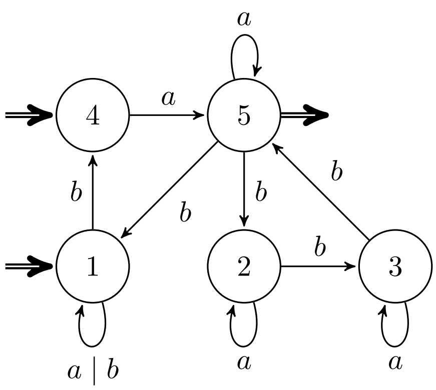
Pourquoi cet AF n'est-il pas déterministe complet ? Construire un AFdc qui reconnaisse le même langage.

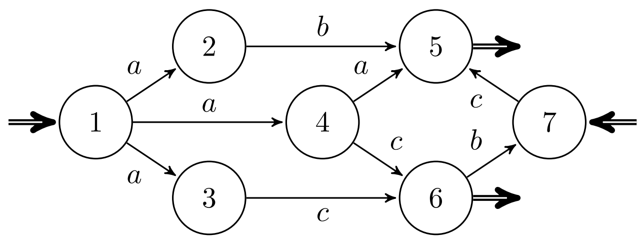
2 - Mêmes questions pour l'automate $(\{a, b\}, \{0, 1, 2\}, \{0, 1, 2\}, \{2\}, \Theta)$.

3 - Mêmes questions pour l'automate $(\{a, b\}, \{0, 1, 2\}, \emptyset, \{2\}, \Theta)$.

4 - Mêmes questions pour l'automate $(\{a, b\}, \{0, 1, 2\}, \{0\}, \{0, 1, 2\}, \Theta)$.

Exercice 16 - Transformations d'AFnd en AFdc.
Pour chacun des deux automates représentés ci-dessous, construire un AFdc qui reconnaisse la même langage :





Exercice 17 - Palindromes « Esope reste ici et se repose »
Soit Σ un alphabet et L un langage sur Σ. On définit sur Σ^* l'opération unaire d'« image miroir » notée $\ldots^{\mu}$:

$$\forall m \in \Sigma^* :$$
$$((m = a_1 a_2 \ldots a_{n-1} a_n \text{ et } \forall i \in \{1, 2 \ldots n\} : a_i \in \Sigma) \Rightarrow m^{\mu} = a_n a_{n-1} \ldots a_2 a_1).$$

On pose aussi $\varepsilon^{\mu} = \varepsilon$, le mot m^{μ} est donc appelé « image miroir » du mot m. Un mot de Σ^* tel que $m^{\mu} = m$ est appelé *palindrome* (Par exemple la suite de symboles « Esope reste ici et se repose » est

un palindrome, si du moins on ne tient pas compte des espaces).
On définit le langage miroir de L par

$$L^\mu = \{n \in \Sigma^* : \exists m \in L, n = m^\mu\} = \{m^\mu : m \in L\}.$$

Parmi les propriétés suivantes, indiquer celles qui sont vraies :

1 - Quel que soit le mot $m \in L$: $mm^\mu = (mm^\mu)^\mu$,

2 - quel que soit le langage L construit sur l'alphabet Σ : $L = (L^\mu)^\mu$,

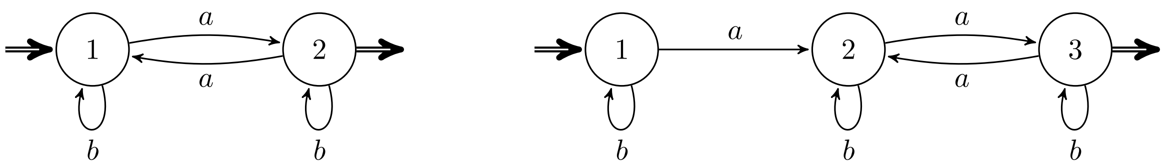
3 - quel que soit x appartenant au langage $L \cap L^\mu$, on a $x = x^\mu$,

4 - quels que soient les deux langages $L1$ et $L2$ construits sur Σ, $(L1L2)^\mu = (L2^\mu)(L1^\mu)$,

5 - quel que soit le langage L construit sur Σ, quel que soit n de $\mathbb{N}$: $(L^n)^\mu = (L^\mu)^n$,

6 - quels que soient $m \in L$ et $n \in L$, on a : $(nmn = (nmn)^\mu) \Rightarrow (m = m^\mu)$.

Exercice 18 - Langage IMPAIR**.**
On considère dans tout l'exercice l'alphabet $\Sigma = \{a, b\}$. On note IMPAIR l'ensemble des mots comportant un nombre impair d'occurrences de la lettre a. Soient $A1$, $A2$ les automates finis donnés par les schémas suivants :



1 - Soit W le langage b^*ab^*. Construire un automate fini $A3$ tel que $L(A3) = W$.

2 - Construire 5 mots de $L(A1)$. Combien ces mots comportent-ils d'occurrences de la lettre a ? Démontrer que $L(A1)$ est inclus dans IMPAIR.

3 - Démontrer, par récurrence sur le nombre d'occurrences de la lettre a, que IMPAIR est inclus dans $L(A1)$. Déduire des questions précédentes que : IMPAIR $= L(A1)$.

4 - Comparer les mots de $L(A2)$ avec les mots de $L(A1)$. En déduire un caractérisation de $L(A2)$.

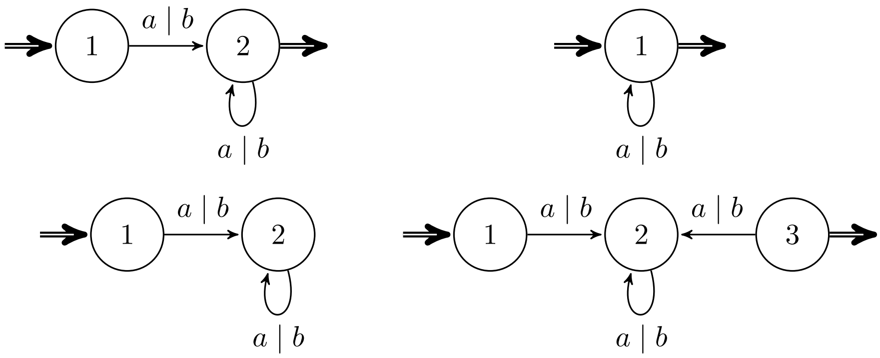
5 - Soit la grammaire $G = (\Sigma_T, \Sigma_{NT}, S, \Pi)$ suivante : $\Sigma_{NT} = \{S, F\}$, $\Sigma_T = \{a, b\}$, Pi est l'ensemble des règles sont :

$$\begin{aligned} S &\longrightarrow bS \mid aF \\ F &\longrightarrow bF \mid aS \mid \varepsilon. \end{aligned}$$

Construire 5 mots de $L(G)$. A-t-on $L(G) = L(A1)$ ou $L(G) = L(A2)$? Présenter la démontration de votre réponse.

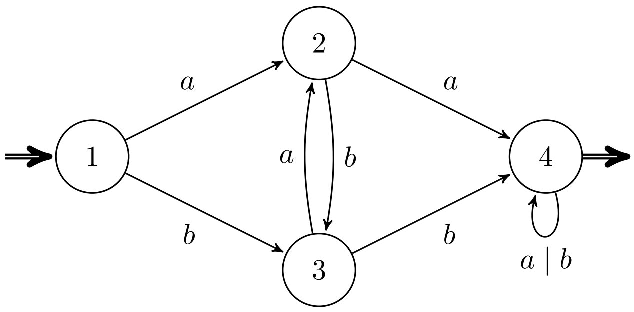
6 - Comparer $L(A3)$ avec $L(A1)$ et $L(A2)$.

Exercice 19 - Automates finis simples.
Déterminer les langages de :



Exercice 20 - Automates finis. *On traitera les deux questions indépendamment l'une de l'autre...*

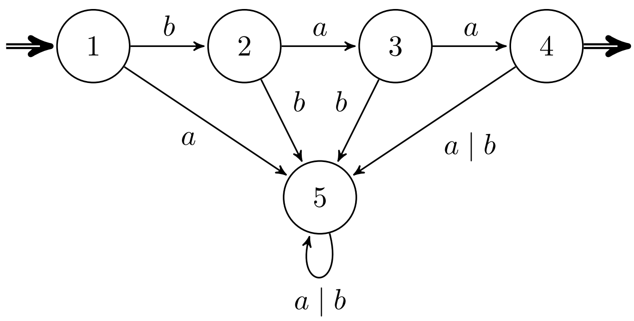
1 - Soit A l'automate fini ci-dessous :

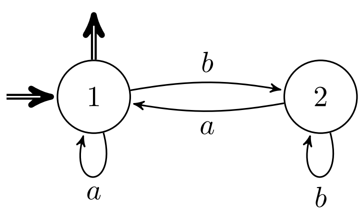


Démontrer que : $L(A) = \{a, b\}^* \{aa, bb\} \{a, b\}^*$.

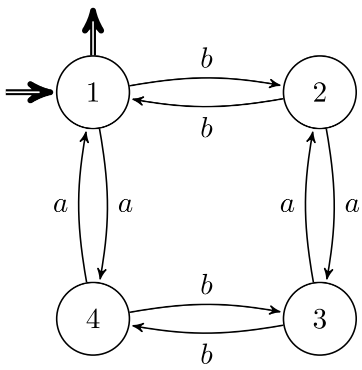
2 - Déterminer un AF dont le langage soit $\{a, b\}^* \{aa, bb\} \{a, b\}^*$. L'automate fini sera déterministe complet, bien sûr, mais rien n'interdit de passer d'abord par un automate fini non déterministe, de le rendre déterministe, et enfin de simplifier l'AFdc obtenu.

Exercice 21 - Automates finis simples.
Déterminer le langage de chacun des automates suivants :





Exercice 22 - Automate fini simple.
Soit l'automate :



Caractériser son langage.

Exercice 23 - Automate fini simple.
Construire un automate fini dont le langage soit $\Sigma^* aa \Sigma^*$, où $\Sigma = \{a, b\}$.

Exercice 24 - Des langages finis.
Soit l'alphabet $\Sigma = \{a, b, c\}$, on pose $\hat{\Sigma} = \{a, b\}$, puis $L1 = \hat{\Sigma}^3 c \hat{\Sigma} \Sigma$ et $L2 = \hat{\Sigma}^3 c \Sigma$.

1 - Montrer que $L1 \subseteq L2$.

2 - A-t-on $L2 \subseteq L1$?

3 - Construire un automate fini $A1$ dont le langage soit $L1$.

4 - Construire une grammaire $G1$ dont le langage soit $L1$.

5 - Construire une grammaire $G2$ et un un automate fini $A2$ dont le langage soit $L2$.

Exercice 25 - Deux grammaires simples.
Soient $G1$ et $G2$ les deux grammaires suivantes :

$$G1 = (\{a, b, c\}, \{S\}, S, \{S \longrightarrow aSb \mid c\})$$

et :

$$G2 = (\{a, b, c, d, e\}, \{S, X\}, S, \{S \longrightarrow aSb \mid X \mid c, X \longrightarrow eXd \mid \varepsilon\})$$

1 - Donner tous les mots de longueur inférieure ou égale à 5 dérivés par la grammaire $G1$.

2 - Donner tous les mots de longueur inférieure ou égale à 5 dérivés par la grammaire $G2$.

3 - Démontrer que $L(G1) = \{a^n c b^n : n \in \mathbb{N}\}$.

4 - Quel est le langage $L(G2)$?

5 - Montrer que $L(G1) \subseteq L(G2)$.

Exercice 26 - Types de grammaire, types de langage.
Soit la grammaire $G = (\Sigma_T, \Sigma_{NT}, S, \Pi)$ où :
$\Sigma_T = \{a, b, c\}$, $\Sigma_{NT} = \{S, X\}$, $\Pi = \{S \longrightarrow XabX, X \longrightarrow aX \mid bX \mid cX \mid \varepsilon\}$.

1 - Construire l'arbre de dérivation du mot $(ab)^2(cb)^2$.

2 - Le mot a^4cb est-il dérivable dans G ? Pourquoi ?

3 - **3.1 -** Déterminer le langage $L(G)$.

3.2 - Ce langage est-il produit par un automate fini ? Pourquoi ?

3.3 - Déterminer un grammaire dont le langage $L(G)$, et dont les règles sont du type suivant : $X \longrightarrow m$ ou $X \longrightarrow mY$ avec $m \in \Sigma_T^*$ et $X, Y \in \Sigma_{NT}$ (En d'autres termes, on cherche une grammaire de type 3 – ou *grammaire régulière* – équivalente à G...).

3.4 - Quels sont les types de grammaires envisagées dans cet exercice ? Quel est le type du langage considéré ?

4 - Déterminer un automate fini A tel que $L(A) = L(G)$.

Exercice 27 - Quatre grammaires.
Soient les quatre grammaires :

$$\begin{aligned}
&G1 = (\{a,b\}, \{S,X,Y\}, S, \{S \longrightarrow XabaY, X \longrightarrow bX \mid b, Y \longrightarrow aY \mid bY \mid \varepsilon\}),\\
&G2 = (\{a,b\}, \{S\}, S, \{S \longrightarrow aS \mid bS \mid a \mid b\}),\\
&G3 = (\{a,b\}, \{S,X,Y\}, S, \{S \longrightarrow X \mid Y, X \longrightarrow \varepsilon, Y \longrightarrow aY \mid bY \mid a \mid b\}),\\
&G4 = (\{a\}, \{S\}, S, \{S \longrightarrow aS \mid \varepsilon\})
\end{aligned}$$

et soient les quatre langages :

$$E1 = a^* \ , \ E2 = \{a,b\}^* \ , \ E3 = E2 \setminus \{\varepsilon\} \ , \ E4 = b^*(ba)^2 E2.$$

1 - Attribuer à chaque grammaire son langage.

2 - Les expressions proposées pour les langages sont-elles régulières ? Si non, proposer des expressions régulières pour ces langages.

3 - Les grammaires proposées sont-elles régulières ? Si non, proposer des grammaires régulières qui engendrent les mêmes langages.

Exercice 28 - Un langage simple.
Soit $\Sigma = \{a,b\}$. Soit L le langage des mots sur Σ dont la première et la dernière lettre sont distinctes.

1 - Construire un automate fini engendrant ce langage.

2 - Construire une grammaire engendrant ce langage.

3 - Donner une expresion régulière de ce langage.

Exercice 29 - Un langage non régulier.
Démontrer qu'il n'existe pas d'automate fini dont le langage est $L = \{a^n b^n : n \in N\}$.
En quoi cela justifie-t-il qu'il n'existe aucune expression régulière pour ce langage ? Justifier que, en revanche, L est algébrique.

Exercice 30 - Grammaires algébriques.
Donner le langage engendré par chacune des grammaires suivantes :

$$\begin{aligned}
&G1 = (\{a,b,c\}, \{A,X\}, X, \{X \longrightarrow Ac \mid c, A \longrightarrow aAb \mid ab\}),\\
&G2 = (\{a,b,c\}, \{A,B,C\}, A, \{A \longrightarrow aB \mid CB \longrightarrow aAb, C \longrightarrow cC \mid c\}),\\
&G3 = (\{a,b\}, \{S,X\}, S, \{S \longrightarrow XaaXX \longrightarrow aX \mid bX \mid \varepsilon\}),\\
&G4 = (\{a,b,c\}, \{S,B,C\}, S, \{S \longrightarrow aB \mid C \mid \varepsilon, B \longrightarrow aSb, C \longrightarrow cC \mid c\}).
\end{aligned}$$

Exercice 31 - Grammaires algébriques.
Étudier les langages dont les règles de réécriture sont les suivantes (comme habituellement, les terminaux sont en minuscule et les non terminaux sont en majuscule, le symbole initial est toujours S).

$$\begin{array}{rlcrl}
G1: & & \quad & G3: & \\
S & \longrightarrow SX & & S & \longrightarrow aS \mid bX \\
SX & \longrightarrow BS & & X & \longrightarrow bX \mid c \\
BS & \longrightarrow ab & & G4: & \\
B & \longrightarrow a & & S & \longrightarrow aaS \mid a \mid b \\
X & \longrightarrow c & & G5: & \\
G2: & & & S & \longrightarrow aX \\
S & \longrightarrow aaS \mid aa & & X & \longrightarrow bS \mid a
\end{array}$$

Exercice 32 - Exercice simple sur les grammaires.
Déterminer des grammaires $G1$ et $G2$ telles que $L(G1) = \{a^p b^q : p \in \mathbb{N} \setminus \{0\}, q \in \mathbb{N} \setminus \{0, 1, 2\}\}$ et $L(G2) = \{a^p b^q : p \in \mathbb{N} \setminus \{0\}, q \in \mathbb{N} \setminus \{0\}\}$.

Exercice 33 - Équivalence de grammaires, ambiguïté.
Soit la grammaire $G = (\Sigma_T, \Sigma_{NT}, S, \Pi)$, où $\Sigma_T = \{x, y\}$, $\Sigma_{NT} = \{S\}$ et où $\Pi = \{S \longrightarrow xSy \mid xS \mid \varepsilon\}$.

1 - Déterminer $L(G)$.

2 - Montrer que le mot x^2y peut être obtenu de deux façons distinctes. On dit que G est *ambiguë*.

3 - Soit alors la grammaire $G1 = (\Sigma_T, \Sigma'_{NT}, S, \Pi')$, où $\Sigma'_{NT} = \{S, T\}$ et où

$$\Pi' = \{S \longrightarrow xSy \mid xT \mid \varepsilon, T \longrightarrow xT \mid \varepsilon\}.$$

Montrer que $L(G1) = L(G)$.

4 - Montrer qu'à tout mot de $L(G1)$ correspond une dérivation unique dans la grammaire $G1$. On aura donc construit une grammaire non ambiguë $G1$ équivalente à la grammaire ambiguë G.

Exercice 34 - Deux grammaires pour un même langage.

1 - Soient les deux grammaires suivantes :

$$\begin{array}{l}
G1 = (\{a, b\}, \{S\}, S, \{S \longrightarrow ab \mid ba \mid SS \mid aSb \mid bSa\}) \\
G2 = (\{a, b\}, \{S, A, B\}, S, \\
\qquad\qquad \{S \longrightarrow aB \mid bA, A \longrightarrow a \mid aS \mid bAA, B \longrightarrow b \mid bS \mid aBB\}).
\end{array}$$

Quelle est la longueur des mots les plus courts de $L(G1)$? et de $L(G2)$?

2 - Déterminer $L(G1)$.

3 - Montrer que $L(G1) = L(G2)$.

Exercice 35 - Réduction de grammaires.
Soit la grammaire $G1 = (\Sigma_T, \Sigma_{NT}, S, \Pi)$, où $\Sigma_T = \{a, b, c\}$, $\Sigma_{NT} = \{S, A, B, C\}$ et Π est la liste de règles :

$$\begin{array}{lcl}
S \longrightarrow aB \mid bc & \quad & A \longrightarrow BAc \mid bSC \mid a \\
B \longrightarrow aSB \mid bBC & & C \longrightarrow SBc \mid aBC \mid ac.
\end{array}$$

1 - Simplifier $G1$ en supprimant le plus possible de règles inutiles, sans changer le langage. La nouvelle grammaire obtenue sera notée $G2$.

2 - Déterminer $L(G2)$.

3 - Déterminer une grammaire $G3$ très simple qui engendre $L(G2)$.

4 - Déterminer un automate fini produisant $L(G1)$.

Exercice 36 - Langages miroirs.
Soit la grammaire $G = (\Sigma_T, \Sigma_{NT}, S, \Pi)$, où : $\Sigma_T = \{a, b, c\}$, $\Sigma_{NT} = \{S, B, C\}$, les règles sont :

$$S \longrightarrow aB \mid C \ , \quad B \longrightarrow aAb \ , \quad C \longrightarrow cC \mid c \ .$$

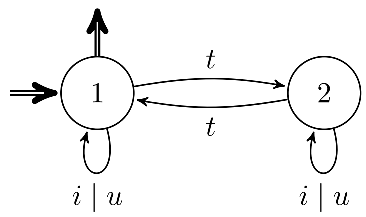
1 - Déterminer $L(G)$.

2 - De quel type est ce langage ?

3 - Déterminer une grammaire G' dont le langage $L(G')$ soit le langage miroir de $L(G)$.

4 - De façon plus générale, si l'on connait G et $L(G)$, comment construire une grammaire G' dont le langage $L(G')$ soit le langage miroir de $L(G)$?

Exercice 37 - Soit l'alphabet $\Sigma = \{i, t, u\}$.
1 - Soit A l'automate fini ci-dessous :



1.1 - Est-il déterministe ? Complet ? Justifier !

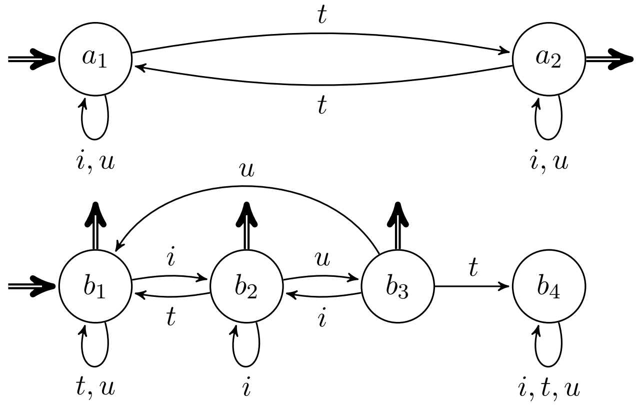
1.2 - Prouver que l'automate A reconnaît le langage (que nous noterons LA) des mots contenant un nombre pair de t.

2 - Soit LB le langage des mots sur Σ qui contiennent (au moins une fois) le mot *iut*.

2.1 - Dessiner un *automate fini non déterministe dont le fonctionnement soit le plus simple possible* et qui reconnaisse le langage LB.

2.2 - Par le procédé vu en cours, *en déduire* un automate fini déterministe et complet qui reconnaît le langage LB, puis justifier (en quelques mots) comment cet automate peut être simplifié.

3 - Soit C l'automate ci-dessous *(Attention, cet automate a DEUX états d'entrée !)*



Par le procédé vu en cours, construire un automate fini déterministe et complet C' qui reconnaît le même langage que C (on ne demande pas de dessiner C', on donnera par contre clairement les différents éléments qui parmettent de définir C', et en particulier on en donnera la table de transition).

4 - On modifie l'automate C' de la manière suivante :
–les états acceptants de viennent non-acceptants,
–inversement, les états non-acceptants de viennent acceptants.
On note D le nouvel automate fini déterministe et complet obtenu, et on note LD le langage reconnu par D.

4.1 - Dessiner l'automate D.

4.2 - Donner une « description en français » de LD.

Exercice 38 - Attention aux confusions entre expressions régulières...
Pour chacune des expressions régulières suivantes, proposer un AF qui la reconnaît.

Pour la dernière de ces expressions, il est fortement conseillé de s'appuyer sur une transition instantanée !
On pourra ensuite déterminer l'AFdc minimal pour chacune de ces expressions.

$$(b \mid ac)* \quad (b \mid a \mid c)* \quad (b(a \mid c))* \quad b(a \mid c)* \quad b* \mid (ac)* \quad b* \mid ac* \quad b*ac* \quad b*(ac)*$$

Exercice 39 - Des grammaires

1 - On envisage la grammaire (axiome S, variables S, X et Y) admettant les règles :

$$\begin{array}{ll} \texttt{R1}: & S \longrightarrow aX \\ \texttt{R2}: & X \longrightarrow Sbb \\ \texttt{R3}: & X \longrightarrow Y \\ \texttt{R4}: & Y \longrightarrow ccY \\ \texttt{R5}: & Y \longrightarrow c \end{array}$$

1.1 - Type de cette grammaire ?

1.2 - Construire 3 mots en montrant comment ils sont produits.

1.3 - Montrer quels sont les enchaînements possibles de ces règles, en déduire le langage produit.

1.4 - Ce langage est-il régulier ?

2 - Soit $L' = \left\{a^{m+1}c^{2n+1}b^{2p} : m, n, p \geq 0\right\}$. Ce langage est-il régulier ? Si oui, proposer un AF qui le reconnaît et une grammaire de type 3 qui l'engendre.

3 - Reprendre la première question en remplaçant $R1$ par $R1' : S \longrightarrow Xa$.

Exercice 40 - Des constructions d'AFdc.
Dans cet exercice, on s'intéresse au langage L des mots sur l'alphabet $x \mid y \mid z$ qui commencent par xy **ET** qui finissent par yyz ; plus précisément, on cherche à construire des AF (automates finis) qui reconnaissent ce langage.

1 - Donner la liste des mots du langage L de longueur inférieure ou égale à 6. *Indication : cette liste devrait comporter cinq mots ; si vous n'en trouvez que quatre, alors vous avez sans doute oublié le plus court d'entre eux !*

2 - On propose l'AF suivant :

	x	y	z
$-$ 1	2		
2		3	
3	3	3 | 4	3
4		5	
5			6
6 $+$			

Dessiner cet AF. Cet AF convient-il ? Pourquoi ?

3 - En tentant de corriger le défaut de l'AF précédent en lui ajoutant une transition de l'état 4 vers l'état 6, d'étiquette z :

	x	y	z
$-$ 1	2		
2		3	
3	3	3 | 4	3
4		5	6
5			6
6 $+$			

Le défaut de l'AF précédent a-t-il été corrigé ? Ce nouvel AF convient-il pour autant ? Pourquoi ?

4 - Proposer une expression régulière pour le langage L.

5 - Dessiner l'AF suivant puis expliquer pourquoi il reconnaît le langage L :

	x	y	z
$-$ 1	2		
2		3 | 4	
3	3	3 | 4	3
4		5	
5			6
6 $+$			

6 - Construire, par l'algorithme de *subset construction*, un AFdc (Automate Fini déterministe complet) équivalent à cet AF : on dessinera l'automate obtenu.
Indication : on devrait obtenir un AFdc à 7 états — on n'oubliera pas l'état de blocage...

7 - En déduire l'Afdc minimal équivalent.

Exercice 41 - Des grammaires
On considère la grammaire **G** dont l'ensemble des terminaux est $\{a, b, c\}$, dont l'ensemble des variables est $\{S, X, Y, Z\}$, dont l'axiome est S et qui a pour ensemble de règles de production :

R1 : $S \longrightarrow aX$
R2 : $Y \longrightarrow cbY$
R3 : $X \longrightarrow abcZ$
R4 : $X \longrightarrow aS$
R5 : $X \longrightarrow abYc$
R6 : $Z \longrightarrow abcZ$
R7 : $Y \longrightarrow \varepsilon$

1 - Cette grammaire est-elle régulière ? Linéaire ? Algébrique ? Pourquoi ?

2 - Proposez 3 mots produits par la grammaire **G** en montrant pour chacun d'eux comment (production ou arbre de dérivation) il est produit.

3 - Dessiner un « automate fini » qui montre quels sont les enchaînements possibles entre règles lorsque l'on cherche à produire des mots terminanux en partant du symbôle initial S ; en particulier, justifier que certaines règles sont « inaccessibles » : on précisera clairement quelles sont ces règles inutiles.

4 - À l'aide de l'« automate » de la question précédente, déterminer le langage produit par la grammaire **G**. *Indication : on devrait trouver* $\left\{a^{2m+2}(bc)^{n+1} \ : \ m,n \in \mathbb{N}\right\}$.

5 - Montrer que le langage précédent peut être écrit sous forme d'expression régulière (on donnera une telle expression régulière) et proposer une grammaire régulière qui engendre ce langage.

6 - On envisage maintenant le langage $\left\{a^{2n+2}(bc)^{n+1} \ : \ n \in \mathbb{N}\right\}$.

6.1 - Proposer une grammaire linéaire qui engendre ce langage.

6.2 - Est-il possible de proposer une grammaire régulière qui engendre ce langage ? Justifier la réponse !

Exercice 42 - Automates finis & grammaires.

1 - Soit un AFdc $\mathbb{A}$ sur l'alphabet $a \mid b \mid c$ qui reconnaît le seul mot a^{2007}.

1.1 - Justifier que alors cet automate a au moins 2009 états...

1.2 - En quoi la réponse précédente est-elle modifiée si l'alphabet est réduit à la seule lettre a ?

2 - On a étudié en cours la grammaire (linéaire)[1] $S \longrightarrow aSb \mid \varepsilon$ qui produit le langage $\{a^n b^n : n \geq 0\}$. En vous inspirant de cet exemple, proposer des grammaires pour chacun des langages suivants (on se contentera d'en énoncer les règles...) :

2.1 - $\left\{a^{2\,n}(bc)^{3\,n} : n \geq 0\right\}$,

2.2 - $\left\{a^{2\,n}b^3c^{2007\,n} : n \geq 0\right\}$,

2.3 - $\left\{a^{2\,n}b^{3\,n}c^{2007} : n \geq 0\right\}$,

2.4 - $\{a^m b^n : m \geq n \geq 0\}$,

2.5 - langage des mots sur $\{a; b\}$ qui comportent autant de a que de b.

2.6 - langage des mots sur $\{a; b; c\}$ qui comportent autant de a que de b.

Exercice 43 - Des transitions « instantanées »

1 - On désire construire l'AFdc minimal sur l'alphabet $x \mid y \mid z$ du langage $x{*}y{*}zx{*}y{*}$,

On propose pour cela l'AF (évidemment non déterministe) suivant :

			ε	x	y	z
$-$	1		2	1		
	2				2	3
	3		4	3		
	4	$+$			4	

En enchaînant les procédés vus en cours (suppression des transitions « instantanées », *subset construction*, algorithme de Nérode), construire l'AFdc minimal équivalent.

[1] Comme à l'habitude, les variables sont notées par des lettres majuscules et les terminaux par des lettres minuscules, la variable de départ est S comme *start* : on se contente donc pour définir cette grammaire d'en énoncer les régles de production qui sont donc $S \longrightarrow aSb$ et $S \longrightarrow \varepsilon$.

2 - On envisage maintenant l'AF :

	ε	x	y	z
$-$ 1	2	1		
2	3		2	
3	5			4
4		3		
5 $+$			5	

2.1 - Proposer une expression régulière de son langage reconnu.

2.2 - Construire l'AFdc minimal équivalent.

Exercice 44 - On envisage les deux grammaires :

$$\begin{aligned} \mathbf{G} &= (\ \{a,b,c,d,e,+,\times\}\,,\{S\}\,,S,\{S \longrightarrow S+S \mid S\times S \mid a \mid b \mid c \mid d \mid e\}\ ) \\ \mathbf{G'} &= (\ \{a,b,c,d,e,+,\times\}\,,\{S,X\}\,,S,\{S \longrightarrow S+X \mid S\times X \mid X\ ,\ X \longrightarrow a \mid b \mid c \mid d \mid e\}\ ) \end{aligned}$$

On remarquera que ces grammaires étant algébriques (c'est-à-dire de type 2), on peut montrer la production d'un mot par un arbre de dérivation.

1 - Montrer que le mot $a+b\times c\times d+e$ appartient au langage produit par **G** en montrant son arbre de dérivation. Que remarque-t-on ?

2 - Idem pour **G'**. Le défaut précédent subsiste-t-il ? L'arbre trouvé correspond-il à celui que votre habitude mathématique attendait ?

3 - Proposer une grammaire qui impose la priorité « à droite ».

4 - On envisage maintenant la grammaire **G"** définie par :

$$\mathbf{G''} = (\ \{a,b,c,d,+,\times\}\,,\{S,T,F\}\,,S,\{S \longrightarrow S+T \mid T\ ,\ T \longrightarrow T\times F \mid F\ ,\ F \longrightarrow a \mid b \mid c \mid d \mid e\}\ )$$

Montrer comment est lu le mot $a+b\times c\times d+e$: y a-t-il ambiguïté ?

5 - Comment modifier la grammaire **G"** de façon à accepter (en leur donnant bien sûr le sens habituel en mathématique) les expressions parenthésées telles que $(a+b)\times c\times d+e$?

Exercice 45 - On envisage l'« automate à pile » suivant destiné à lire les (c'est-à-dire à construire les arbres syntaxiques associés à des) expressions arithmétiques parenthésées, comme $((a+(b\times c))+d)-e$.

– **début :** au départ, la pile contient le symbole de fond de pile $\boxed{\bullet}$; l'expression[2] à analyser *expression* est remplacée par (*expression*).

– **empilement 1 :** à la lecture d'un symbole représentant une variable a, b..., on empile un pointeur vers ce symbole.

– **empilement 2 :** à la lecture de l'un des symboles (, $+$ $-$, / ou $\times$, on empile ce symbole.

– **réduction :** à la lecture du symbole) :

  – **cas** (gBd) **:** si le dessus de la pile est du type $\boxed{\bullet}\boxed{\ \ldots\ }\boxed{(}\boxed{g}\boxed{B}\boxed{d}$ où g et d sont des pointeurs et où B est un opérateur binaire $+$, $-$, / ou $\times$, alors dépiler les quatre cases du dessus de la pile puis empiler un pointeur sur B, avec g et d pour fils gauche et droit,

  – **cas** (p) **:** si le dessus de la pile est du type $\boxed{\bullet}\boxed{\ \ldots\ }\boxed{(}\boxed{p}$ où p est un pointeur, alors dépiler les deux cases du dessus de la pile puis empiler le pointeur p,

[2] Autrement dit, avant de procéder à l'analyse de l'expression, on la fait précéder d'une parenthèse ouvrante et suivre d'une parenthèse fermante.

- **autres cas :** envoyer un message d'erreur et tout arrêter !

– **fin :** en fin de lecture, la pile doit se réduire au symbole de fond de pile et à un pointeur $\boxed{\bullet} \boxed{p}$ où p est un pointeur vers l'arbre de l'expression analysée ; si c'est le cas, renvoyer cet arbre, si non, envoyer un message d'erreur et tout arrêter !

1 - Montrer la lecture de $((a + (b \times c)) + d) - e$ en détaillant les étapes de réduction.
Faire de même pour $((a + (b \times c) + d) - e$.

2 - Quelle difficulté serait apparue lors de la lecture de l'expression précédente si l'on n'avait pas pris la précaution, avant tout calcul, d'encadrer l'expression par des parenthèses comme expliqué en **début** ?

3 - On désire autoriser des écritures plus compactes en sous-entendant des parenthèses grâce à des priorités — par exemple, $a+b/c$ pour $(a+(b/c))$ — et à des associativités gauches entre opérateurs de même priorité — par exemple, $a - b + c$ pour $((a - b) + c)$.

Pour cela, on modifie les règles précédentes en ajoutant un nouveau cas de réduction :

lorsqu'on s'*apprête* à lire un symbole binaire B :

–on s'assure que le sommet le pile est du type $\boxed{\bullet} \boxed{\;\ldots\;} \boxed{B'} \boxed{p}$ où p est un pointeur ;

–si le symbole B' est de priorité inférieure strictement à celle de B — et pour cette circonstance le symbole de fond de pile $\bullet$ et la parenthèse ouvrante (sont considérés comme de priorité minimale — alors on lit et empile B,

–si par contre B' est de priorité supérieure ou égale à B, alors, sans lire B, on réduit les trois sommets du haut de la pile.

Montrer comment est lue l'expression $a + b \times c + d - e$.

4 - Si maintenant nous désirons que notre langage puisse accepter des expressions telles

$$([(a \times b) - (c + a)] + (d - e))$$

où l'on peut utiliser deux types de “parenthèses” : (et) d'une part, [et] d'autre part.

Comment “corriger” notre automate à pile pour qu'il accepte de telles expressions ?