Théorie des Langages 1

Recueil d'exercices

1 Induction structurelle

Exercice 1 Soit V un vocabulaire et soit un sous-ensemble $A \subseteq V$. Dans cet exercice on considérera la fonction $|\,.\,|_A : V^* \to \mathbb{N}$ qui à tout mot $w \in V^*$ associe le nombre d'occurrences d'éléments de A présents dans w. Ainsi, si $V = \{a, b, c, d\}$ et $A = \{a, b\}$, alors :

- $|cabdbacc|_A = 4$,
- $|bbaba|_A = 5$,
- $|ccdc|_A = 0$.

On pose $V = \{\wedge, \vee, \neg,), (, \bot, \top\}$, et on considère l'ensemble E des formules logiques défini par induction structurelle de la façon suivante :

Base. $\top \in E$ et $\bot \in E$.

Induction. Si w, w_1 et w_2 sont dans E, alors :

- $(\neg\, w) \in E$,
- $(w_1 \wedge w_2) \in E$,
- $(w_1 \vee w_2) \in E$.

▷ Question 1 Est-ce que les deux mots suivants sont dans E ? On justifiera en quelques mots les réponses.

1. $(\top \wedge \neg)$
2. $(\top \wedge \top \wedge \top)$

▷ Question 2 Montrer que $((\top \wedge \top) \vee (\bot \vee \top)) \in E$.

On définit les ensembles de symboles suivants :

$$U = \{\neg\}, \qquad B = \{\wedge, \vee\},$$
$$S = \{\top, \bot\}, \qquad N = U \cup B \cup S.$$

▷ Question 3 Soit $w_1 = (\top \wedge ((\neg\bot) \vee (\top \wedge (\bot \vee (\neg\top)))))$. Calculer $|w_1|_U$, $|w_1|_B$ et $|w_1|_S$.

▷ Question 4 Démontrer par induction structurelle que pour tout $w \in E$, on a $|w|_S = |w|_B + 1$.

▷ Question 5 **[Avancé]** Soit $w \in E$. Exprimer $|w|$ en fonction de $|w|_B$ et $|w|_U$.

▷ Question 6 **[Avancé]** Utiliser la question 5 pour justifier formellement les réponses à la question 1.

Exercice 2 Définir la fonction $|\,.\,|_A$ de l'exercice 1 par induction structurelle sur V^*.

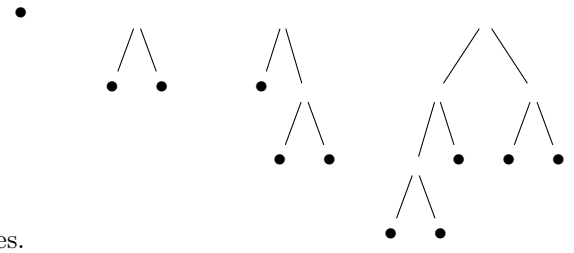
Exercice 3 Définitions inductives d'ensembles.

▷ Question 1 Donner des définitions inductives des langages suivants :

1. L'ensemble L_1 des mots sur $\{a, b\}$ de longueur paire.
2. L'ensemble L_2 des mots sur $\{a, b\}$ ne contenant pas deux a consécutifs.
3. L'ensemble L_3 des palindromes sur $\{a, b\}$.
4. **[Avancé]** L'ensemble L_4 des mots sur $\{a, b\}$ contenant un nombre pair de a.
5. **[Avancé]** L'ensemble L_5 des mots sur $\{a, b\}$ contenant autant de a que de b.

▷ Question 2 **[Avancé]** Prouver que ces définitions inductives sont correctes.

Exercice 4
On s'intéresse aux arbres binaires, mais pour simplifier, on considère qu'ils ne contiennent pas de données. On rappelle qu'un arbre binaire est un arbre dont tous les noeuds internes ont exactement deux fils.
En voici ci-contre quatre exemples :



▷ Question 1 Donner une définition inductive des arbres binaires.

▷ Question 2 Montrer par induction structurelle que dans un arbre binaire le nombre n_i de nœuds internes et le nombre n_f de feuilles (nœuds sans fils) satisfont la relation suivante : $n_f = n_i + 1$.

▷ Question 3 La formule d'énumération d'un ensemble inductif formé à partir d'un ensemble de cas de base B et d'un ensemble de constructeurs inductifs K est :

$$\begin{aligned} E_0 &\stackrel{\text{def}}{=} B, \\ E_{n+1} &\stackrel{\text{def}}{=} E_n \cup \{\kappa_i(e_1, \ldots, e_{k_i}) \mid \kappa_i \in K, e_1, \ldots, e_{k_i} \in E_n\} \end{aligned}$$

Donner les trois premiers ensembles E_n pour la définition des arbres binaires de la question précédente. Quelle est la hauteur des arbres contenus dans ces ensembles ? On rappelle que la hauteur d'un arbre est la longueur maximale d'un chemin de la racine (le nœud tout en haut de l'arbre) vers une feuille de l'arbre.

▷ Question 4 **[Avancé]** Soit H_n l'ensemble des arbres binaires de hauteur inférieure ou égale à n. Montrer l'égalité $E_n = H_n$ par récurrence sur n.

Exercice 5 [Avancé] On considère un langage E_p d'expressions préfixées *sur des chiffres*. Pour cela, on définit les ensembles suivants :

$$\begin{aligned} \text{Num} &= \{\texttt{Zr}, \texttt{Un}, \texttt{De}, \texttt{Tr}, \texttt{Qu}, \texttt{Ci}, \texttt{Si}, \texttt{Sp}, \texttt{Hu}, \texttt{Ne}\}, \\ \text{Op} &= \{+, -, \times\}. \end{aligned}$$

L'ensemble E_p est alors un langage sur le vocabulaire $V = \text{Num} \cup \text{Op}$, dont les éléments sont des expressions arithmétiques sur des chiffres, où les opérateurs sont placés avant les opérandes.

Par exemple, l'expression préfixée correspondant à $1 + 2$ est $+$ `Un De` ; l'expression préfixée correspondant à $(2 + 3) \times (3 - 1)$ est $\times$ $+$ `De Tr` $-$ `Tr Un`.

▷ Question 1 Donner une définition inductive du langage E_p.

On considère la fonction **eval** : $E_\text{p} \to \mathbb{N}$, qui calcule la valeur d'une expression préfixée. Par exemple, si $w =$ $\times$ $+$ `De Tr` $-$ `Tr Un` alors $\textbf{eval}(w) = 10$.

▷ Question 2 Donner une définition inductive de la fonction **eval**.

Pour un mot $w \in E_\text{p}$ donné, on note $|w|_\text{Num}$ le nombre d'éléments de Num présents dans w, et on note $|w|_\text{Op}$ le nombre d'éléments de Op présents dans w.

▷ Question 3 Prouver par induction structurelle que pour tout $w \in E_\text{p}$, on a $|w|_\text{Num} = |w|_\text{Op} + 1$.

2 Langages

Exercice 6 [A savoir faire] Soit L un langage. Montrer que les propositions suivantes sont équivalentes :

(i) $\varepsilon \in L$
(ii) $\forall i \geq 0$, $\varepsilon \in L^i$
(iii) $\forall i \geq 0$, $L^i \subseteq L^{i+1}$

Exercice 7 On considère dans cet exercice des langages définis sur $V = \{a, b\}$.

▷ QUESTION 1 Définir **par concaténation et itération** le langage L_1 des mots constitués d'une séquence de a suivie d'une séquence de b. Les séquences peuvent éventuellement être vides.

▷ QUESTION 2 Définir **par induction** le langage L_2 des mots de la forme $a^n b^n$ avec $n > 0$.

▷ QUESTION 3 Définir **à l'aide des opérations ensemblistes classiques** et des langages précédents le langage L_3 des mots de la forme $a^i b^j$ avec $i \neq j$.

Exercice 8 Soit V un vocabulaire. Etant donnés deux mots $x, z \in V^*$, on dit que x est conjugué à z s'il existe un mot $y \in V^*$ tel que $xy = yz$. On souhaite prouver que x est conjugué à z si et seulement s'il existe deux mots $u, v \in V^*$ tels que $x = uv$ et $z = vu$.

▷ QUESTION 1 Montrer que s'il existe deux mots $u, v \in V^*$ tels que $x = uv$ et $z = vu$ alors x est conjugué à z.

▷ QUESTION 2 On suppose que x est conjugué à z et que $x = \varepsilon$. Déterminer u et v tels que $x = uv$ et $z = vu$.

On suppose maintenant que x est conjugué à z et que $x \neq \varepsilon$. Soit alors y **de longueur minimale** tel que $xy = yz$.

▷ QUESTION 3 Montrer qu'on a nécessairement $|x| \geq |y|$.

▷ QUESTION 4 En déduire l'existence de u et v tels que $x = uv$ et $z = vu$.

▷ QUESTION 5 **[Avancé]** Montrer que la relation de conjuguaison est une relation d'équivalence.

Exercice 9 [A savoir faire] Etant donné un ensemble E, on rappelle qu'une relation ρ sur E est un sous-ensemble de $E \times E$. La relation ρ est :

- Réflexive si pour tout $x \in E$, on a $(x, x) \in \rho$.
- Symétrique si pour tout $x, y \in E$, on a $(x, y) \in \rho$ si et seulement si $(y, x) \in \rho$.
- Antisymétrique si pour tout $x, y \in E$, si $(x, y) \in \rho$ et $(y, x) \in \rho$, alors $x = y$.
- Transitive si pour tout $x, y, z \in E$, si $(x, y) \in \rho$ et $(y, z) \in \rho$, alors $(x, z) \in \rho$.

Soit $E = \{a, b, c, d, e\}$, et considérons la relation ρ définie par :

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

Construire les relations suivantes :

1. La fermeture réflexive de ρ (i.e. la plus petite relation réflexive contenant ρ).
2. La fermeture symétrique de ρ (i.e. la plus petite relation symétrique contenant ρ).
3. La fermeture transitive de ρ (i.e. la plus petite relation transitive contenant ρ).

Exercice 10 Soient L et M des langages sur un vocabulaire V. Montrer que si $L \subseteq M$ alors $L^* \subseteq M^*$. La réciproque est-elle vraie ? Justifier.

Exercice 11 Pour chacune des égalités suivantes identifier celles qui sont vraies et celles qui sont fausses. Pour la première catégorie on donnera une intuition de la preuve. Pour la seconde catégorie on donnera des contre-exemples et on identifiera les inclusions uni-directionnelles.

1. $L^* = (L^*)^*$
2. $L^* \cup M^* = (L \cup M)^*$
3. $(L^* \cup M^*)^* = (L \cup M)^*$
4. $L^+ = L^* \setminus \{\varepsilon\}$
5. $(LM)^* = L^* M^*$
6. $(L \cup M)^* = (L^* M^*)^*$

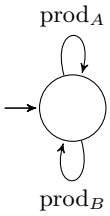
3 Automates finis et modélisation

Exercice 12 Construire des automates reconnaissant les langages suivants :

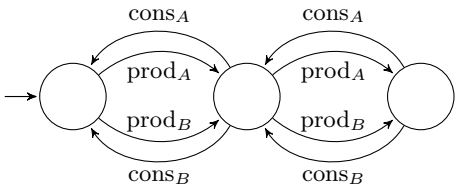
1. L'ensemble des nombres binaires sans zéro inutile en tête.
2. Les mots sur $\{a, b\}$ contenant deux a et/ou deux b consécutifs.
3. Les mots sur $\{a, b\}$ contenant un nombre pair de a et un nombre impair de b.

Exercice 13 L'objectif de cet exercice est de modéliser *un buffer à deux places* (c.-à-d. pouvant stocker au plus deux données) par un système producteur/consommateur pour deux types de ressources A et B, le tout à l'aide d'automates. Le vocabulaire (également appelé *ensemble de synchronisation*) est : { prodA, prodB, consA, consB }.

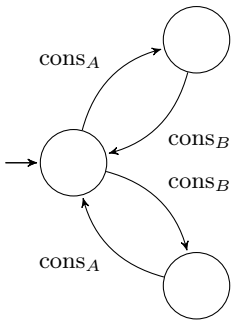
Supposons les trois automates suivants décrivant respectivement un producteur, un buffer et un consommateur. Notez qu'ils n'ont pas d'états acceptants car le système n'est pas sensé s'arrêter et évolue indéfiniment à chaque événement (chaque symbole prod_A, prod_B, cons_A, cons_B lu). Dans l'automate du consommateur, il y a implicitement des boucles prod_A et prod_B sur chaque état car le consommateur ignore ces événements. De même pour l'automate du producteur, il y a implicitement des boucles cons_A et cons_B.



Producteur



Buffer à deux places

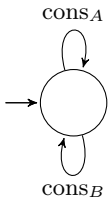


Consommateur

▷ Question 1 Calculer le produit des automates prod×buffer×cons.

▷ Question 2 Quel est le problème mis en avant dans ce produit ?

▷ Question 3 En supposant le consommateur plus classique ci-dessous, modéliser à nouveau l'automate attendu du buffer à deux places.



▷ Question 4 Quel automate faudrait-il donner pour le buffer afin de corriger le problème identifié ?

Exercice 14 L'ensemble des littéraux numériques en Python forme un langage, formellement défini dans `https://docs.python.org/3/reference/lexical_analysis.html#numeric-literals`. Dans cet exercice, on considère un sous-ensemble des littéraux entiers et flottants écrits en base 10. Ils sont composés d'une partie entière, d'une partie décimale optionnelle et d'un exposant optionnel ; ils sont définis sur le vocabulaire

$$V \stackrel{\text{def}}{=} \{\mathbf{0}, \ldots, \mathbf{9}, \mathbf{e}, \mathbf{E}, ., +, -\}.$$

On définit les huit ensembles suivants (suite sur la page suivante) :

$$\begin{aligned}
\texttt{nonzerodigit} &\stackrel{\text{def}}{=} \{\mathbf{1}, \ldots, \mathbf{9}\} \\
\texttt{digit} &\stackrel{\text{def}}{=} \{\mathbf{0}\} \cup \texttt{nonzerodigit} \\
\texttt{integer} &\stackrel{\text{def}}{=} \texttt{nonzerodigit}\,(\texttt{digit}^*) \cup \{\mathbf{0}\}^+ \\
\texttt{dot} &\stackrel{\text{def}}{=} \{.\} \\
\texttt{pointfloat} &\stackrel{\text{def}}{=} (\texttt{digit}^*)\,\texttt{dot}\,(\texttt{digit}^+) \cup (\texttt{digit}^+)\,\texttt{dot}
\end{aligned}$$

$$\begin{aligned} \texttt{exponent} &\stackrel{\text{def}}{=} \{\,\mathbf{e}, \mathbf{E}\}\,\{\,\varepsilon, +, -\}\,\texttt{digit}^+ \\ \texttt{exponentfloat} &\stackrel{\text{def}}{=} \big(\texttt{digit}^+ \cup \texttt{pointfloat}\big)\,\texttt{exponent} \\ \texttt{number} &\stackrel{\text{def}}{=} \texttt{integer} \cup \texttt{pointfloat} \cup \texttt{exponentfloat} \end{aligned}$$

▷ Question 1 Parmi les mots suivants, lesquels appartiennent à `number` ? Lesquelles n'y appartiennent pas ?
.314, **.3E+4**, **0.5E−2**, **0000**, **E67**, **1E7e3**, **6E+1234**, **2E++3.4**

▷ Question 2 Donner un automate qui reconnaît le langage `integer`.

▷ Question 3 Donner un automate qui reconnaît le langage `number`.

Exercice 15 Un fermier cherche à faire traverser une rivière à son chou, sa chèvre et son loup. Pour cela, il dispose d'une petite barque qui ne permet de transporter qu'un seul des trois à la fois (en plus de lui-même). Étant donné que le loup mange la chèvre et que la chèvre mange le chou, le fermier doit faire attention à qui ou quoi il laisse seuls sur chacune des rives. Le fermier peut-il faire traverser la rivière au chou, à la chèvre et au loup sans qu'aucun ne se fasse dévorer ?

▷ Question 1 Représenter le problème par un automate, en précisant le vocabulaire choisi.

▷ Question 2 Comment déterminer une stratégie à partir de l'automate ? Quelles sont les stratégies optimales ?

Exercice 16 [A savoir faire] Donner des automates reconnaissant les langages suivants :

1. $L_1 = \{\omega \in \{a,b\}^* \,|\, \omega$ contient au moins un a et un $b\}$.
2. $L_2 = \{\omega \in \{a,b\}^* \,|\, \omega$ ne contient pas deux a consécutifs$\}$.
3. $L_3 = \{\omega \in \{a,b\}^* \,|\, \omega$ ne contient pas *plus de* deux a consécutifs$\}$.
4. $L_4 = \{\omega \in \{a,b\}^* \,|\, \omega$ a bab pour suffixe$\}$.
5. $L_5 = \{\omega \in \{a,b\}^* \,|\,$ la cinquième lettre de ω est un $a\}$.

Exercice 17 [A savoir faire] Soit L_2 le langage défini inductivement de la façon suivante :
— $\varepsilon \in L_2$.
— Si $w \in L_2$, alors $bw \in L_2$.
— Si $w_1, w_2, w_3 \in L_2$, alors $w_1 a w_2 a w_3 \in L_2$.

▷ Question 1 Définir L_2 en compréhension.

▷ Question 2 Donner un automate qui reconnaît ce langage.

Exercice 18 [Avancé] On considère un vocabulaire V et une relation $R \subseteq V \times V$. On définit

$$H_R \stackrel{\text{def}}{=} \{w_1 \cdots w_k \,|\, k \geq 2,\, \forall 1 \leq i < k,\, (w_i, w_{i+1}) \in R\}.$$

Autrement dit, la relation R impose des contraintes sur les symboles qui peuvent se suivre au sein des mots de H_R. Le but de cet exercice est de montrer que H_R est régulier.

On pose $V_0 = \{a,b,c,d,e\}$ et $R_0 = \{(a,b),(b,e),(d,d),(a,c),(d,c),(e,d)\}$.

▷ Question 1 Enumérer les mots de H_{R_0} de longueur inférieure ou égale à 3.

▷ Question 2 Construire un automate qui reconnaît H_{R_0}.

▷ Question 3 En supposant V et R **quelconques**, démontrer que le langage H_R est reconnu par un automate fini.

Exercice 19 [Avancé] Donner en français un algorithme qui prend en entrée deux automates et qui teste si oui ou non ils reconnaissent le même langage.

Exercice 20 [Avancé] Soit L un langage régulier sur un vocabulaire V. On définit le langage $\sqrt{L}$ de la façon suivante :

$$\sqrt{L} \stackrel{\text{def}}{=} \{x \in V^* \mid xx \in L\}.$$

Démontrer que $\sqrt{L}$ est régulier.

Exercice 21 Le produit de deux automates est défini de la façon suivante : soient $A_1 = (Q_1, V, \delta_1, \{i_1\}, F_1)$ et $A_2 = (Q_2, V, \delta_2, \{i_2\}, F_2)$ deux automates qu'on suppose complets et sans ε-transition. On considère l'automate

$$A_1 \times A_2 \;=\; (Q_1 \times Q_2,\, V,\, \delta,\, \{\langle i_1, i_2\rangle\},\, F_1 \times F_2),$$

où δ est défini par : pour tous $\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle \in Q_1 \times Q_2$ et pour tout $a \in V$,

$$(\langle q_1, q_2\rangle, a, \langle q_1', q_2'\rangle) \in \delta \text{ si et seulement si } (q_1, a, q_1') \in \delta_1 \wedge (q_2, a, q_2') \in \delta_2.$$

On pose $A = A_1 \times A_2$; on souhaite prouver que $\mathcal{L}(A) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

▷ Question 1 Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0,1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

On considère maintenant deux automates A_1 et A_2 quelconques, complets et sans ε-transition. Soient, pour $n \geq 1$, les ensembles d'états $\{p_0, \ldots, p_n\} \subseteq Q_1$ et $\{q_0, \ldots, q_n\} \subseteq Q_2$. Soit $w \in V^*$.

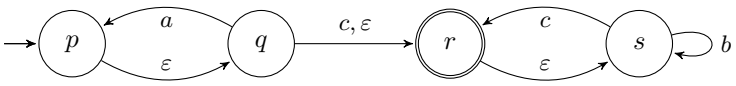
▷ Question 2 Démontrer par induction que les assertions suivantes sont équivalentes :
- $(p_0, a_1, p_1) \cdots (p_{n-1}, a_n, p_n)$ est un chemin de trace $w = a_1 \cdots a_n$ dans A_1 et $(q_0, a_1, q_1) \cdots (q_{n-1}, a_n, q_n)$ est un chemin de trace w dans A_2 ;
- $(\langle p_0, q_0\rangle, a_1, \langle p_1, q_1\rangle) \cdots (\langle p_{n-1}, q_{n-1}\rangle, a_n, \langle p_n, q_n\rangle)$ est un chemin de trace w dans A.

▷ Question 3 En déduire que $\mathcal{L}(A) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

▷ Question 4 Quel langage aurait été reconnu si l'ensemble des états finaux de A avait été $(F_1 \times Q_2) \cup (Q_1 \times F_2)$ et non pas $F_1 \times F_2$? Justifier.

4 Élimination des ε-transitions

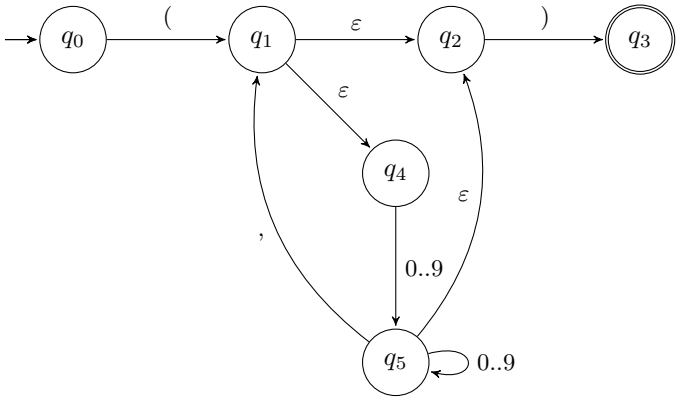
Exercice 22 [A savoir faire] Construire un automate sans ε-transition équivalent à celui ci-dessous :



Exercice 23 [A savoir faire] On s'intéresse à l'ensemble des tuples d'entiers (simples) en Python. Dans ce langage, il est possible d'écrire :
- Le tuple vide : ()
- Un tuple à un élément : (42,)
- Un couple : (42, 29) ou bien (42, 29,)
- Un triplet : (12, 25, 37) ou bien (12, 25, 37,)
- etc...

En particulier, (123) ne représente **pas** un tuple : il s'agit de la valeur 123 entourée de parenthèses superflues. Les autres expressions telles que (,) ou encore (, 42, 29) sont interdites.

On propose l'automate suivant pour reconnaître les tuples d'entiers.



Construire un automate sans ε-transition équivalent à celui proposé. L'automate proposé répond-il bien aux spécifications ? Justifier.

Exercice 24 Soit $A = (Q, V, \delta, I, F)$ un automate. On rappelle qu'un état $q \in Q$ est *accessible dans* A s'il existe un chemin dans A dont l'origine est dans I et l'extrémité est q. L'état q est *productif dans* A s'il existe un chemin dans A dont l'origine est q et l'extrémité est dans F.

On note $\mathrm{Acc}(A)$ l'ensemble des états accessibles dans A, et $\mathrm{Prod}(A)$ l'ensemble des états productifs dans A.

▷ Question 1 Soit $Q = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$. On considère l'automate $A = (Q, \{a, b\}, \delta, \{p_0, p_1\}, \{p_3, p_4\})$, où la relation de transition δ est définie par :

δ	a	b
p_0	p_1	p_2
p_1	-	p_3
p_2	p_1	p_3, p_4
p_3	-	-
p_4	-	p_3
p_5	p_6	p_4
p_6	p_4	p_4

Déterminer les ensembles $\mathrm{Acc}(A)$ et $\mathrm{Prod}(A)$.

On considère maintenant un automate A **quelconque, sans ε-transition**.

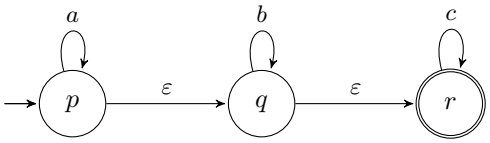
▷ Question 2 Définir un algorithme qui calcule l'ensemble des états accessibles dans A.

▷ Question 3 Même question pour l'ensemble des états productifs dans A.

▷ Question 4 Donner une condition nécessaire et suffisante sur $\mathrm{Acc}(A)$ et $\mathrm{Prod}(A)$ pour que $\mathcal{L}(A) \neq \emptyset$.

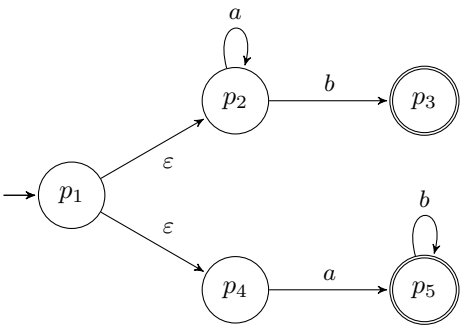
▷ Question 5 Donner une condition nécessaire et suffisante sur A permettant d'assurer $\mathcal{L}(A) = V^*$?

Exercice 25 [A savoir faire] Construire un automate sans ε-transition équivalent à celui ci-après. Quel est le langage reconnu par cet automate ?



5 Déterminisation

Exercice 26 [A savoir faire] Déterminiser l'automate suivant :



Exercice 27 [A savoir faire] On s'intéresse au langage L des mots binaires dont le dernier bit est un bit de parité. Plus précisément, un mot $wx \in \{0,1\}^+$ est dans L si le nombre de 1 dans w est impair et x vaut 1, ou si le nombre de 1 dans w est pair et x vaut 0. Autrement dit, on choisit x pour que le nombre de 1 dans wx soit pair.

Exemples : 0, 011011, 1010 et 001111 sont dans L.
Contre-exemples : 11010, 110001 et ε ne sont pas dans L.

Construire un automate déterministe complet reconnaissant L.

Exercice 28 [A savoir faire] Construire des automates déterministes complets reconnaissant :

1. les nombres entiers > 0, en base 10, qui sont des multiples de 100 ;
2. les nombres entiers > 0, en base 10, qui sont des multiples de 3 ;
3. **[Avancé]** pour n et k entiers, les nombres entiers > 0, en base k, qui sont multiples de n.

Exercice 29 [A savoir faire] Soit $Q = \{p_1, p_2, p_3, p_4\}$. On considère l'automate $A = (Q, \{a, b, c\}, \delta, \{p_1\}, \{p_3, p_4\})$, où la relation de transition δ est définie par :

δ	a	b	c	ε
p_1	p_1, p_2	-	p_3	-
p_2	-	p_1	-	p_3
p_3	-	p_3, p_4	-	-
p_4	-	p_4	-	-

Construire un automate déterministe complet équivalent à A.

Exercice 30 Un barman aveugle portant des gants de boxe joue au jeu suivant avec l'un de ses clients réguliers : quatre verres sont disposés en carré sur un plateau circulaire pouvant pivoter autour de son centre. L'objectif du barman est de mettre tous les verres dans le même sens : soit tous à l'endroit, soit tous à l'envers. Pour compliquer les choses, après chaque mouvement du barman, le client peut faire pivoter le plateau d'un ou plusieurs quarts de tour. Évidemment, comme le barman ne peut savoir s'il a gagné (il est aveugle et porte des gants de boxe), le client arrête le jeu lorsque le barman gagne.

La question est la suivante : le barman possède-t-il une stratégie gagnante ? Si oui, quelle est la stratégie optimale (i.e. celle avec un nombre minimum de coups) ?

▷ Question 1 En utilisant les symétries du problème, déterminer quels sont les états du plateau pertinents pour le barman. Faire de même pour les coups du barman.

▷ Question 2 Construire un automate non déterministe qui correspond à ce jeu du point de vue du barman. Que peut-on dire des états initiaux ?

▷ Question 3 Déterminiser cet automate, en déduire la réponse pour le barman.

Exercice 31 Pour $k > 0$, soit L_k le langage constitué des mots sur $\{0,1\}$ de longueur au moins k, et dont le $k^{\text{ième}}$ symbole **en partant de la fin** est un 1. Par exemple, 00101 et 100110111 sont dans L_3. Formellement,

$$L_k \stackrel{\text{def}}{=} \{a_1 \dots a_n \mid n \geq k \wedge a_{n-k+1} = 1\}.$$

▷ QUESTION 1 Construire un automate (non-déterministe) à $k+1$ états qui reconnaît L_k.

▷ QUESTION 2 Construire un automate déterministe complet minimal reconnaissant L_2.

On cherche à borner la taille minimale d'un automate déterministe complet reconnaissant L_k. Soit $A = (Q, \{0,1\}, \delta, \{q_0\}, F)$ un automate déterministe complet reconnaissant L_k. On définit $f : \{0,1\}^k \to Q$, qui à tout mot u de longueur k associe $\delta^*(q_0, u)$. Autrement dit, $f(u)$ est l'état atteint par le chemin de trace u dans A, partant de q_0.

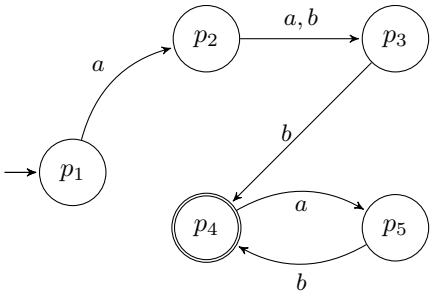
▷ QUESTION 3 **[Avancé]** Montrer que f est injective. En déduire une borne inférieure de la taille de A.

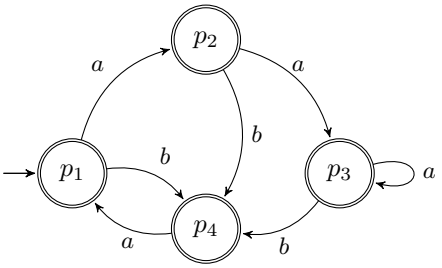
Exercice 32 [A savoir faire] Construire des automates déterministes reconnaissant les langages sur $\{a, b\}$ suivants :

1. L'ensemble des mots terminés par ab ou bien par ba.
2. L'ensemble des mots contenant au moins deux fois la séquence ab.
3. Le langage $\{aab\}^*\{b\}$.
4. Le langage $\{a\}^*\{aba\}^*$.

6 Minimisation

Exercice 33 Minimiser les automates suivants :





Exercice 34 Soit $Q = \{q_0, q_1, q_2, q_3, q_4\}$. Déterminiser et minimiser l'automate $A = (Q, \{a, b\}, \delta, \{q_0\}, \{q_4\})$, où la relation de transition δ est récapitulée ci-dessous :

δ	a	b	ε
q_0	q_1	$\times$	q_3, q_4
q_1	q_1	q_0	$\times$
q_2	$\times$	q_4	q_1
q_3	q_3	q_1	$\times$
q_4	$\times$	$\times$	q_3

Exercice 35 [A savoir faire] Construire les automates minimaux reconnaissant les langages suivants sur $\{a, b\}$:

1. L'ensemble des mots de longueur paire et terminés par ab.
2. L'ensemble des mots contenant la sous-chaîne aa.
3. L'ensemble $\{a\}\{aa, bb\}^*\{a, b\}^*\{b\}$.

7 Expressions régulières

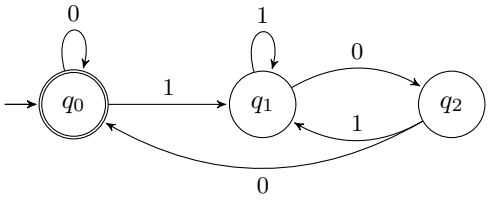
Exercice 36 [A savoir faire] Soit E une expression régulière. Simplifier les expressions suivantes :

1. $E.E^* + \epsilon$
2. $\epsilon.E$
3. $\emptyset^*$
4. ϵ^*
5. $\emptyset.E$
6. $\emptyset + E$

Exercice 37 Donner une expression régulière représentant chacun des langages suivants :

1. Les mots sur $\{0,1\}$ contenant deux 0 et/ou deux 1 consécutifs.
2. Les mots sur $\{0,1\}$ où chaque 0 est suivi d'un 1.
3. Les mots sur $\{0,1\}$ contenant au moins un 0.
4. Les mots sur $\{0,1\}$ composés de 0 et de 1 alternés.
5. Les mots sur $\{0,1\}$ de longueur paire.

Exercice 38 Calculer l'expression régulière correspondant à l'automate ci-dessous, en résolvant le système d'équations obtenu dans deux ordres différents, puis en utilisant la méthode par suppression d'états dans les mêmes ordres. Constater que les systèmes associés aux automates avec certains états supprimés correspondent aux différentes étapes de résolution du système initial.



Exercice 39 [A savoir faire] Caractériser (par une phrase en français) les langages représentés par les expressions régulières suivantes :

1. $0^*(10^*10^*10^*)^*$
2. $(1 + 01 + 001)^*(\epsilon + 0 + 00)$
3. $1^*(0 + \epsilon)1^*$

Exercice 40 On considère les *expressions régulières étendues*, qui sont obtenues en ajoutant aux expressions régulières les constructions suivantes :

— si E est une E.R., alors $\neg E$ est une E.R. étendue ;
— si E et E' sont des E.R., alors $E \cap E'$ est une E.R. étendue ;
— si E est une E.R., alors E^+ est une E.R. étendue.

La sémantique de ces opérateurs est la suivante :

— $\mathcal{L}(\neg E) = V^* \setminus \mathcal{L}(E)$;
— $\mathcal{L}(E \cap E') = \mathcal{L}(E) \cap \mathcal{L}(E')$;
— $\mathcal{L}(E^+) = \mathcal{L}(E).\mathcal{L}(E)^*$.

Démontrer qu'à toute E.R. étendue est associé un langage régulier.

Exercice 41 [A savoir faire] Nous considérons une représentation des messages codés en Morse à l'aide du formalisme suivant. Les signaux qui peuvent être émis sont :

— le signal de début de phrase : D ;
— les signaux pour constituer des mots : L (signal long) et C (signal court)
— le signal de fin de phrase : F.

Un mot en Morse est une succession de trois signaux, longs ou courts. Une phrase en Morse est une séquence non-vide de mots, précédée du signal de début de phrase, et terminée par le signal de fin de phrase. Un message en Morse est une séquence éventuellement vide de phrases.

Donner une expression régulière décrivant l'ensemble des messages valides en Morse.

Exercice 42 [A savoir faire] Donner une expression régulière représentant l'ensemble des mots avec un nombre pair de 0 et un nombre impair de 1, en définissant un automate reconnaissant ce langage et en résolvant les équations associées.

Exercice 43 [Avancé] Etant donnée une expression régulière E, on définit la *hauteur d'étoile de E* comme le nombre d'étoiles de Kleene imbriquées dans E. Par exemple, $\mathrm{H}_K(a) = \mathrm{H}_K((a+b).(c+d)) = 0$, $\mathrm{H}_K(a^*) = \mathrm{H}_K(ab^*(a+c)^*) = 1$ et $\mathrm{H}_K((ab^*c)^*) = 2$.

▷ Question 1 Définir formellement la fonction H_K sur l'ensemble des expressions régulières par induction structurelle.

La notion de hauteur d'étoile est étendue aux langages réguliers : si L est un langage régulier, alors $\mathrm{H}_K(L)$ est défini par :

$$\mathrm{H}_K(L) \;=\; \min\{\mathrm{H}_K(E) \,|\, \mathcal{L}(E) = L\}.$$

▷ Question 2 Soit $E = a(a^*b^*)^*bb$. Quelle est la valeur de $\mathrm{H}_K(E)$? Quelle est la valeur de $\mathrm{H}_K(\mathcal{L}(E))$?

▷ Question 3 Soit L un langage régulier. Démontrer que L est fini si et seulement si $\mathrm{H}_K(L) = 0$.

8 Propriétés de clôture

Exercice 44 On admet que $M = \{a^nb^n \,|\, n \geq 0\}$ n'est pas régulier. Montrer que les langages suivants ne sont pas réguliers non plus **sans se servir du lemme de l'étoile** :

1. $L_1 = \{w \in \{a,b\}^* \,|\, w$ a autant de a que de $b\}$
2. $L_2 = \{a^ib^jc^k \,|\, i + j = k \geq 0\}$
3. $L_3 = \{(ab)^{2n}(cd)^{2n} \,|\, n \geq 0\}$
4. **[Avancé]** $L_4 = \{uv \in \{a,b\}^* \,|\, vu \in \{a^nb^n \,|\, n \geq 0\}\}$

Exercice 45 En utilisant le lemme de l'étoile, montrer que les langages suivants ne sont pas réguliers :

1. $L_1 = \{wb^n \,|\, n \in \mathbb{N}, w \in \{a,b\}^n\}$
2. $L_2 = \{w \in \{a,b\}^* \,|\, w$ est un palindrome$\}$
3. **[Avancé]** $L_3 = \{1^{i^2} \,|\, i \geq 0\}$
4. **[Avancé]** $L_4 = \{1^p \,|\, p$ est premier$\}$

Exercice 46 Etant donné un vocabulaire V, on considère une famille $(L_i)_{i\in\mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

▷ Question 1 Soit $n \geq 0$. Montrer que le langage $M_n = \bigcup_{0\leq i\leq n} L_i$ est régulier.

▷ Question 2 Peut-on en déduire que $\bigcup_{0\leq i} L_i$ est régulier ? Justifier.

Exercice 47 Equivalence entre automates

▷ Question 1 Donner une méthode pour déterminer si deux automates sont équivalents.

▷ Question 2 Si deux automates ne sont pas équivalents, comment faire pour exhiber un contre-exemple, c.-à-d. un mot accepté par l'un mais pas par l'autre ?

▷ Question 3 En se basant sur la construction de l'automate produit (exercice 21), comment faire cette construction plus directement ?

Exercice 48 [A savoir faire] Les langages suivants sont-ils réguliers ?

1. $L_1 = \{a^nb^m \,|\, n \neq m\}$.
2. $L_2 = \{w \in \{a,b\}^* \,|\, |w|_a \neq |w|_b\}$.

9 Grammaires et Hiérarchie de Chomsky

Exercice 49 Langages hors-contextes et langages réguliers

▷ Question 1 Donner des grammaires hors-contextes engendrant les langages suivants :

1. $\{a^n b^p | n \geq p \geq 0\}$
2. $\{a^n b^p | n \neq p\}$
3. $\{a^n b^p | 2p \geq n \geq p\}$
4. $\{a^n b^p c^q | n + p = q\}$

▷ Question 2 Donner des grammaires régulières engendrant les langages suivants :

1. les mots sur $\{a, b\}$ ayant un nombre pair de a et impair de b
2. **[A savoir faire]** l'ensemble des constantes entières sans 0 inutiles en tête

Exercice 50 Opérations sur les langages

Soient $G_1 = (V_T, V_{N_1}, S_1, R_1)$ et $G_2 = (V_T, V_{N_2}, S_2, R_2)$ deux grammaires. On supposera sans perte de généralité que $V_{N_1} \cap V_{N_2} = \emptyset$.

▷ Question 1 Donner une grammaire G telle que $L(G) = L(G_1) \cup L(G_2)$. Comment prouver que cette grammaire est correcte ?

▷ Question 2 Même question pour les langages $L(G_1)$. $L(G_2)$ et $L(G_1)^*$.

▷ Question 3 En supposant que G_1 et G_2 soient de type T (régulière, hors-contexte, sous-contexte) que peut-on dire des grammaires proposées aux questions précédentes ?

Exercice 51 Langages sous-contextes

▷ Question 1 Soit la grammaire $G = (\{a, b, c\}, \{S, B, C\}, S, R)$ avec R l'ensemble des règles suivantes :

$$(1)\quad S \to abc \qquad (2)\quad S \to aSBc$$
$$(3)\quad cB \to Bc \qquad (4)\quad bB \to bb$$

a) Justifier le type de cette grammaire.

b) Construire une dérivation de la chaîne *aabbcc*.

c) Soit un mot quelconque de la forme $a^n b^n c^n$ avec $n > 0$. Donner une méthode générale permettant de produire ce mot à partir de la grammaire précédente.

▷ Question 2 **[Avancé]** Donner une grammaire sous-contexte engendrant les mots de la forme wcw avec $w \in \{a, b\}^*$. On pourra partir de la grammaire suivante, qui engendre les mots de la forme $wc\widetilde{w}$ avec $\widetilde{w}$ l'image miroir de w :

$$S \to aSa \mid bSb \mid c$$

Exercice 52 Langage des carrés

Soit $V_T = \{0, 1\}$. Le langage W des mots de la forme ww n'est pas hors-contexte mais on peut montrer que son complémentaire C l'est.

▷ Question 1 Soit Y le langage des mots sur V_T de longueur impaire dont le milieu est 0. Soit Z le langage des mots sur V_T de longueur impaire dont le milieu est 1. Donner une grammaire hors-contexte pour chacun de ces languages.

▷ Question 2 Montrer que toute chaîne de $YZ \cup ZY$ n'est pas de la forme ww.

▷ Question 3 Montrer que toute chaîne de longueur paire, qui n'est pas de la forme ww appartient à $YZ \cup ZY$. En déduire une grammaire pour C.

Exercice 53 Forme réduite d'une grammaire

Soit $G = (V_T, V_N, S, R)$ une grammaire hors-contexte. On donne les définitions suivantes :

— un symbole A de V_N est dit *productif* si et seulement si il existe une dérivation $A \Longrightarrow^* w$ avec $w \in V_T^*$.

— un symbole A de V_N est dit *accessible* si et seulement si il existe une dérivation $S \Longrightarrow^* w_1 A w_2$ avec $w_1, w_2 \in (V_T \cup V_N)^*$.

▷ QUESTION 1 Soit la grammaire $G = (\{a, b, c, d\}, \{S, A, B, C, D\}, S, R)$ avec R défini par :

$$\begin{array}{lllll} S \to AB & S \to \epsilon & A \to aA & A \to D & \\ B \to bB & B \to aS & C \to c & C \to cC & D \to dA \end{array}$$

Donner l'ensemble Pr des symboles productifs de G ainsi que l'ensemble Ac des symboles accessibles.

▷ QUESTION 2 Soit $G = (V_T, V_N, S, R)$ une grammaire hors-contexte quelconque.

1. Donner une méthode permettant de calculer, à partir des règles de la grammaire, l'ensemble Pr des non-terminaux productifs.
2. Même question pour l'ensemble Ac des non-terminaux accessibles.
3. En utilisant les résultats précédents, comment peut-on décider simplement si $L(G) \neq \emptyset$?

▷ QUESTION 3 L'algorithme de « nettoyage » des grammaires est le suivant :

1. On calcule Pr, l'ensemble des symboles productifs de la grammaire G, puis on construit $G' = (V_T, Pr, S, R')$ avec R' le sous-ensemble des règles de R ne contenant aucune occurrence d'un symbole non productif, ni en partie gauche, ni en partie droite : $R' = R - \{A \to w_1 B w_2 \mid A \notin Pr \vee\ B \notin Pr\}$.
2. On calcule Ac, l'ensemble des symboles accessibles de G', puis on construit $G'' = (V_T, Ac, S, R'')$ avec R'' le sous-ensemble des règles de R' ne contenant aucune occurrence d'un symbole non accessible, ni en partie gauche, ni en partie droite : $R'' = R' - \{A \to w_1 B w_2 \mid A \notin Ac \vee\ B \notin Ac\}$.

Appliquer cet algorithme à la grammaire $G = (\{a, b\}, \{S, A, C, D, E\}, S, R)$ pour R défini par :

$$S \to A \quad S \to a \quad A \to CD \quad C \to b \quad D \to A \quad E \to C$$

▷ QUESTION 4 En utilisant l'exemple précédent, montrer que l'algorithme naïf qui consisterait à construire la grammaire G' ci-après est faux (i.e. ne produit pas une grammaire réduite).

$G' = (V_T, Ac \cap\ Pr, S, R')$ avec R' le sous-ensemble des règles de R ne contenant aucune occurrence d'un symbole non accessible ou productif, ni en partie gauche, ni en partie droite : $R' = R - \{A \to w_1 B w_2 \mid A \notin Ac \vee\ B \notin Ac \vee\ A \notin Pr \vee\ B \notin Pr\}$.

Exercice 54 Grammaires régulières

On donne ci-après plusieurs manières de décrire les grammaires régulières. Soit $G = (V_T, V_N, S, R)$ une grammaire avec les restrictions suivantes :

— **Def1** : règles de la forme $A \to w$ ou $A \to uB$ avec $w \in V_T^*$, $u \in V_T^+$, $A \in V_N$ et $B \in V_N$.
— **Def2** : règles de la forme $A \to xB$ ou $A \to \epsilon$ avec $x \in V_T$ et $A \in V_N$ et $B \in V_N$.

▷ QUESTION 1 Soit L le langage défini sur le vocabulaire $\{a, b\}$ par $(ab)^*$. Une grammaire possible pour ce langage conforme à **Def1** est $S \to abS \mid \epsilon$. Donner une grammaire conforme à **Def2** pour ce langage.

▷ QUESTION 2 Montrer que la classe des langages définie par ces deux formes de grammaire est la même.

10 Grammaires hors-contexte

Exercice 55 Grammaire ambiguë

▷ QUESTION 1 Montrer que les grammaires suivantes sont ambiguës et proposer des grammaires équivalentes non ambiguës :

1. $S \to aSaS \qquad S \to \epsilon$
2. $S \to aSb \qquad S \to aS \qquad S \to \epsilon$

▷ QUESTION 2 Donner une grammaire non ambiguë pour le langage $\{a^n b^p | 2p \geq n \geq p\}$.

▷ QUESTION 3 **[Avancé]** Une ϵ-règle est une règle de la forme $A \to \epsilon$.

Une 1-règle est une règle de la forme $A \to B$ avec A et B éléments du vocabulaire non-terminal.

Soit G une grammaire hors-contexte ne contenant ni ϵ-règle ni 1-règle.

Pour tout mot $\omega \in \mathcal{L}(A)$, avec $A \in V_N$, on peut borner la longueur de la dérivation $A \Rightarrow^* \omega$ en fonction de $|\omega|$. Soit $A \Rightarrow^d \omega$. Montrer que $d \leq 2 * |w| - 1$.

Exercice 56 Preuves sur les grammaires

On définit deux langages L_1 et L_2 sur le vocabulaire $V = \{a, b\}$.

Soit $P_1(w)$ la propriété $|w|_a = |w|_b$ et $P_2(w)$ la propriété $\forall u \in Prefixe(w) |u|_a \geq |u|_b$. Rappelons que la notation $|w|_x$ représente le nombre d'occurrences du symbole x dans w et u est un préfixe de w si et seulement il existe une chaîne v sur V^* tel que $uv = w$.

— **Définition de L_1 par compréhension.**
$L_1 = \{w : P_1(w) \wedge P_2(w)\}$
— **Définition de L_2 par grammaire.**
$S \rightarrow \epsilon$, $S \rightarrow SS$ et $S \rightarrow aSb$ avec $L_2 = \mathcal{L}(S)$.

▷ Question 1 Justifier en quoi la chaîne *aabbab* appartient à la fois à L_1 et L_2.

▷ Question 2 On veut montrer que $L_2 \subseteq L_1$. On rappelle que ceci revient à montrer que tout mot dérivable à partir S vérifie les propriétés $P1$ et $P2$.

1. ajouter une règle de grammaire qui violerait la propriété $P2$.
2. en considérant la grammaire initiale prouvez $L_2 \subseteq L_1$.

▷ Question 3 On veut maintenant montrer que $L_1 \subseteq L_2$. Pour cela on doit montrer que tout mot vérifiant les propriétés $P1$ et $P2$ peut être dérivé de l'axiome.

1. On vous propose de faire l'analyse par cas suivante : (1) $w = \epsilon$, (2) $w = abu$, (3) $w = uab$ (4) $w = aub$. Justifier en quoi cette décomposition n'est pas complète, i.e. qu'il existe des mots de L_1 qui ne peuvent être produits comme une combinaison de ces différents cas.
2. prouvez $L_1 \subseteq L_2$.

Exercice 57 Décrire les langages de programmation

▷ Question 1 Écrire une grammaire décrivant les identificateurs Python V2 (non-terminal *Idf*) sur le vocabulaire V_1 ={**a**, ..., **z**,**A**, ..., **Z**, **0**, ..., **9**,**_** } (à partir de la version 3 d'autres caractères sont permis). On rappelle qu'un identificateur ne peut jamais commencer par un chiffre.

▷ Question 2 En Python l'instruction d'affectation est de la forme : $Inst \rightarrow Cible = Exp$. En partie gauche d'une affectation, on peut trouver (entre autre) les éléments suivants :
— un identificateur (ex : `x = 1`),
— l'accès à l'attribut d'un objet (ex : `o.x = 0`),
— l'accès à un élément d'une liste (ex : `l[i+1] = 0`)
Exemples : $x[y.z]$ $x.y[2]$

Soit V_2 le vocabulaire obtenu à partir de V_1 en ajoutant les symboles point (**.**), crochet ouvrant (**[**), crochet fermant (**]**) et virgule (**,**) (pour la question suivante).

Écrire une grammaire décrivant la catégorie syntaxique *Cible*. On utilisera le non-terminal *Idf* défini à la question 1 ainsi que le non-terminal *Exp* (une expression quelconque), à ne pas définir.

Donner l'arbre de dérivation associé à la cible $x.y[2]$.

La grammaire proposée est-elle ambigüe ? En donner une non-ambigüe.

▷ Question 3 En fait une cible peut aussi être une liste de cibles. Cette liste doit être non vide, commence par un crochet ouvrant et finit par un crochet fermant. Les cibles sont séparées par une virgule et la dernière occurence de cible peut, ou non, être suivie d'une virgule. Compléter la grammaire pour prendre en compte cette définition de cible.

▷ Question 4 Toute cible est une expression. Il existe par contre des expressions qui ne sont pas des cibles (ne peuvent apparaître en partie gauche d'affectation). Donner des exemples d'expression qui ne sont pas des cibles.

Exercice 58 [A savoir faire] Conversion entre grammaire régulière et automate

Soit la grammaire $G = (\{a, b, c\}, \{S, A, B, C\}, S, R)$ où R contient les règles suivantes :

$$S \rightarrow aS \mid aA \mid cB \qquad B \rightarrow bB \mid bC \mid \varepsilon$$
$$A \rightarrow B \mid bS \qquad C \rightarrow bC \mid \varepsilon$$

Construire un automate déterministe reconnaissant $\mathcal{L}(G)$.