*Ce problème comporte 6 parties indépendantes. La plupart des questions peuvent être traitées en admettant les résultats des questions précédentes. Les algorithmes demandés seront écrits dans un langage algorithmique proche de Pascal ou de LISP. Toute structure de données utilisée sera clairement définie.*

Un alphabet $A$ est un ensemble fini de symboles appelés lettres. Un mot $u$ sur $A$ est une suite finie $(a_1, \ldots, a_n)$. Ce mot sera noté simplement $u = a_1 \ldots a_n$ si aucune confusion n'est possible. L'entier $n$ est la longueur du mot $u$, notée $|u|$. Le mot vide est l'unique mot de longueur nulle. Pour toute lettre $a$ de $A$, le nombre d'occurrences de la lettre $a$ dans le mot $u$ est noté $|u|_a$. Le sous-ensemble de $A$ constitué des lettres ayant au moins une occurrence dans le mot $u$ est appelé alphabet du mot $u$ et est noté $\text{alph}(u)$. L'ensemble de tous les mots sur un alphabet $A$ fixé est noté $A^*$.
Cet ensemble est muni d'une opération interne appelée concaténation. La concaténation de deux mots $u = (a_1, \ldots, a_n)$ et $v = (b_1, \ldots, b_m)$ est simplement le mot $(a_1, \ldots, a_n, b_1, \ldots, b_m)$ noté $u \cdot v$ ou simplement $uv$. La concaténation est une opération qui est associative mais non commutative dès que l'alphabet $A$ a au moins deux lettres. Le mot vide est l'élément neutre pour la concaténation.
Pour $B \subseteq A$, la projection sur $B^*$ d'un mot $u \in A^*$ est le mot obtenu à partir de $u$ en effacant toutes les lettres n'appartenant pas à $B$. L'application qui à tout mot $u$ associe sa projection sur $B^*$ est notée $\Pi_B$.
Exemple 1: Soit $A = \{a, b, c\}$ et $u = abaaba$. Nous avons alors $|u| = 6$, $|u|_a = 4$,$|u|_b = 2$, $|u|_c = 0$ et $\text{alph}(u) = \{a, b\}$. Si $v = cba$ alors $uv = abaabacba$ et $vu = cbaabaaba$. Ainsi $\Pi_{\{a,c\}}(uv) = aaaaca$ et $\Pi_{\{b,c\}}(vu) = cbbb$.

On considère sur $A$ une relation irréflexive et symétrique $I \subseteq A \times A$. La relation $I$ est appelée relation d'indépendance et son complément $D = (A \times A) \setminus I$ est la relation de dépendance associée. Deux sous-ensembles $B$ et $C$ de $A$ sont dits indépendants si $(b, c) \in I$ pour tout $b \in B$ et $c \in C$. Deux mots $u, v \in A^*$ sont dits indépendants si leurs alphabets $\text{alph}(u)$ et $\text{alph}(v)$ le sont. Par abus de notation, on note alors simplement $(u, v) \in I$. La relation $I$ induit une relation binaire $R$ sur $A^*$ définie par:

$$R = \{(uabv, ubav) \mid u, v \in A^*, (a, b) \in I\}$$

La clôture réflexive et transitive de la relation $R$ est notée $\sim$ et est donc définie par, pour tout $u, v \in A^*$:

$$u \sim v \iff \exists u_0 = u, u_1, \ldots, u_n = v \text{ tels que } (u_i, u_{i+1}) \in R \; \forall 0 \leq i < n$$

Deux mots $u, v$ de $A^*$ vérifiant $u \sim v$ sont dits $\sim$-équivalents.
Exemple 2: Soit $A = \{a, b, c, d\}$ et $I = \{(a, b), (b, a), (b, c), (c, b)\}$. Nous avons alors $aabcbcd \sim baaccbd \sim aabbccd$. Les mots $acaa$ et $bb$ sont indépendants, par contre les mots $acaa$ et $bd$ ne le sont pas.

Dans l'ensemble du problème, on se fixe un alphabet $A$ et une relation d'indépendance $I \subseteq A \times A$.

# 1 Mots $\sim$-équivalents

1. Montrer que $\sim$ est une congruence i.e. $\forall u, v, u', v' \in A^*$,

$$(u \sim u', v \sim v') \Longrightarrow uv \sim u'v'$$

2. Soit $(P)$ une propriété sur les mots de $A^*$ qui vérifie $\forall u, v \in A^*$ tels que $(u, v) \in R$,
$$P(u) \Longrightarrow P(v)$$
Montrer que $\forall u, v \in A^*$ tels que $u \sim v$,
$$P(u) \Longrightarrow P(v)$$

3. Soient $u, v$ deux mots $\sim$-équivalents de $A^*$. Montrer que $|u|_a = |v|_a$ pour toute lettre $a$ de $A$, $\text{alph}(u) = \text{alph}(v)$ et $|u| = |v|$.

4. Soient $u, v$ deux mots $\sim$-équivalents de $A^*$. Montrer que $\Pi_{\{a,b\}}(u) = \Pi_{\{a,b\}}(v)$ pour tout couple $(a, b) \in D$.

5. Soient $u, v$ deux mots de $A^*$ tels que $\Pi_{\{a,b\}}(u) = \Pi_{\{a,b\}}(v)$ pour tout couple $(a, b) \in D$.
   (a) Montrer que $|u| = |v|$.
   (b) On suppose $u$ non vide. Soit $a \in A$ la première lettre de $u$ i.e. $u = au'$ avec $u' \in A^*$.
      i. Montrer qu'il existe deux mots $v', v''$ de $A^*$ tels que $v = v'av''$ et $(a, v') \in I$.
      ii. Montrer que $\Pi_{\{a,b\}}(u') = \Pi_{\{a,b\}}(v'v'')$ pour tout couple $(a, b) \in D$.
   (c) Montrer par récurrence sur $|u|$ que $u \sim v$.

6. Déduire des deux questions précédentes un algorithme linéaire en la longueur du mot pour décider si deux mots $u, v$ de $A^*$ sont $\sim$-équivalents.

7. Montrer que si $u, v$ et $w$ sont trois mots de $A^*$ vérifiant $uw \sim vw$ ou $wu \sim wv$ alors $u \sim v$.

8. Soit $u$ un mot de $A^*$ et $a$ une lettre de $A$. La lettre $a$ est dite minimale dans $u$, on note alors $a \in \min(u)$, si $u = u'au''$ avec $(a, u') \in I$. Ainsi si $A = \{a, b, c, d\}$ et $I = \{(a, b), (b, a), (b, c), (c, b)\}$, les lettres minimales du mot *abbac* sont $a$ et $b$ alors que celles du mot *bbcaa* sont $b$ et $c$. Montrer l'équivalence des trois assertions suivantes:
   (a) $a \in \min(u)$.
   (b) $\exists v \in A^*$ tel que $v \sim u$ et $v$ commence par la lettre $a$.
   (c) $\forall b \in A$, $(a, b) \in D \Longrightarrow \Pi_{\{a,b\}}(u)$ commence par la lettre $a$.

9. Soit $u$ un mot de $A^*$. Donner un algorithme qui énumère tous les mots $v$ de $A^*$ qui sont $\sim$-équivalents à $u$.

# 2 Lemme de Levi

Soient $u, v, w$ et $t$ quatre mots de $A^*$.

1. Montrer que
$$xy = zt \Longleftrightarrow \exists u \in A^* \text{ tel que } (x = zu \text{ et } t = uy) \text{ ou } (z = xu \text{ et } y = ut)$$

2. La condition nécessaire et suffisante de la question précédente est-elle encore vérifiée si l'on n'a plus $xy = zt$ mais seulement $xy \sim zt$?

3. Soit $(P)$ la propriété sur $(A^*)^4$ définie par, pour $m, n, r, s \in A^*$,

$$P(m,n,r,s) \text{ si } \exists \alpha,\beta,\gamma,\delta \in A^* \text{ tels que } \begin{cases} m = \alpha\beta & r = \alpha\gamma \\ n = \gamma\delta & s = \beta\delta \\ (\beta,\gamma) \in I \end{cases}$$

(a) Montrer que si $P(x,y,z,t)$ alors $xy \sim zt$.

(b) Supposons que $xy \sim zt$ et $|x| > 0$. Soit $a$ la première lettre de $x$: $x = ax'$ avec $x' \in A^*$. Montrer que ou bien $z = z'az''$ avec $(a,z') \in I$ ou bien $t = t'at''$ avec $(a,zt') \in I$.

(c) Montrer par récurrence sur $|x|$ que

$$xy \sim zt \Longrightarrow P(x,y,z,t)$$

# 3 Forme Normale de Foata

On suppose dans cette partie que l'alphabet $A$ est totalement ordonné par une relation irréflexive, antisymétrique et transitive $<$. Un mot $u \in A^*$ est dit ordonné si $u = a_1 \dots a_k$ avec $a_i \in A$ pour tout $i \in \{1,\dots,k\}$ et $a_1 < a_2 < \dots < a_k$. Un mot non vide $u$ de $A^*$ est dit sous Forme Normale de Foata (FNF en abrégé) si $u = u_1 \dots u_n$ avec

- $\forall i \in \{1,\dots,n\}, u_i$ est un mot non vide ordonné
- $\forall i \in \{1,\dots,n\}$, les lettres de alph$(u_i)$ sont deux à deux indépendantes
- $\forall i \in \{2,\dots,n\}$, $\forall b \in \text{alph}(u_i)$, $\exists a \in \text{alph}(u_{i-1})$ tel que $(a,b) \in D$

Ainsi si $A = \{a,b,c,d\}$ avec $a < b < c < d$ et $I = \{(a,b),(b,a),(b,c),(c,b)\}$, le mot $(ab)(bc)(c)$ est sous FNF. Par contre le mot $aab$ ne l'est pas, en effet les seules décompositions vérifiant les deux premières conditions sont $(a)(ab)$ et $(a)(a)(b)$ qui ne vérifient pas la troisième condition.

1. Donner un algorithme qui, étant donné un mot non vide $u$ de $A^*$, dit s'il est sous FNF ou pas. Dans le cas où le mot $u$ est sous FNF, l'algorithme donnera alors une décomposition $(u_1,\dots,u_n)$ correspondante de $u$.
2. Soit $u$ un mot non vide de $A^*$ sous FNF et $a$ une lettre de $A$. Montrer qu'il existe un mot $v$ de $A^*$ sous FNF et tel que $v \sim ua$.
3. Montrer que pour tout mot non vide $u$ de $A^*$, il existe un unique mot $v$ de $A^*$ tel que $v \sim u$ et $v$ est sous FNF. Le mot $v$ est noté $FNF(u)$.
4. En déduire que deux mots $u$ et $v$ non vides de $A^*$ sont $\sim$-équivalents si et seulement si $FNF(u) = FNF(v)$.
5. Donner un algorithme qui, étant donné un mot non vide $u$ de $A^*$, calcule $FNF(u)$.

# 4 Graphes de dépendance

Un graphe est un couple $G = (V,E)$ où $V$ est l'ensemble fini de sommets et $E \subseteq V \times V$ est l'ensemble des arêtes. Soient $x_1$ et $x_n$ deux sommets de $V$, un chemin de $x_1$ à $x_n$ dans $G$ est une suite $(x_1,\dots,x_n)$ d'éléments de $V$ tels que $(x_i,x_{i+1}) \in E$ pour tout $i \in \{1,\dots,n-1\}$. Un chemin $(x_1,\dots,x_n)$ est un cycle si $n \geq 2$ et $x_1 = x_n$. La clôture transitive $G^+ = (V,E^+)$ de $G$ est définie par $(x,y) \in E^+$ s'il existe un chemin de $x$ à $y$ dans $G$.

Un graphe étiqueté est un triplet $G = (V, E, \lambda)$ où $(V, E)$ est un graphe et $\lambda : V \longrightarrow A$ un étiquetage des sommets dans $A$.

Un graphe de dépendance est un graphe étiqueté $G = (V, E, \lambda)$ tel que

- $(V, E)$ ne comporte pas de cycle
- $\forall x, y \in V, (\lambda(x), \lambda(y)) \in D \Longleftrightarrow ((x, y) \in E$ ou $(y, x) \in E$ ou $x = y)$

Pour écrire les algorithmes demandés dans cette partie, on considérera qu'un graphe $G = (V, E)$ est donné par sa matrice d'adjacence $M_G$ dont les indices sont les éléments de $V$, les valeurs dans $\{0, 1\}$ et telle que, pour tout couple $(x, y)$ de $V \times V$, $M_G(x, y) = 1$ si et seulement si $(x, y) \in E$. Pour définir un graphe étiqueté $G = (V, E, \lambda)$, on se donnera en plus l'étiquetage sous la forme d'un tableau $L_G$ à une dimension dont les indices sont les éléments de $V$ et les valeurs dans $A$.

1. Ecrire un algorithme qui, étant donné un graphe $G = (V, E)$, calcule la clôture transitive $G^+$ de $G$.

2. Ecrire un algorithme qui, étant donné un graphe étiqueté $G = (V, E, \lambda)$, vérifie si $G$ est un graphe de dépendance ou pas.

3. Un sommet $y$ d'un graphe $G = (V, E)$ est minimal s'il n'existe pas dans $E$ d'arête de la forme $(x, y)$. Montrer que les étiquettes des éléments minimaux d'un graphe de dépendance sont deux à deux indépendantes.

4. Soit $G = (V, E)$ un graphe sans cycle avec $V = \{x_1, \ldots, x_n\}$. Soit $\sigma$ une permutation de $\{1, \ldots, n\}$. La suite $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ est un tri topologique de $G$ si pour tout $i, j$ de $\{1, \ldots, n\}$:
$$(x_{\sigma(i)}, x_{\sigma(j)}) \in E^+ \Longrightarrow i < j$$
Un mot $u = a_1 \ldots a_n$ avec $a_i \in A$ pour $i \in \{1, \ldots, n\}$ est une linéarisation d'un graphe de dépendance $G = (V, E, \lambda)$ s'il existe un tri topologique $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ de $G$ tel que $a_i = \lambda(x_{\sigma(i)})$ pour tout $i \in \{1, \ldots, n\}$. L'ensemble des linéarisations d'un graphe de dépendance $G = (V, E, \lambda)$ est noté $\mathrm{lin}(G)$.

   On considère dans cette question un graphe de dépendance $G = (V, E, \lambda)$ fixé.

   (a) Ecrire un algorithme qui calcule un élément de $\mathrm{lin}(G)$.

   (b) Ecrire un algorithme qui calcule tous les éléments de $\mathrm{lin}(G)$.

   (c) Soient $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ et $(x_{\sigma'(1)}, \ldots, x_{\sigma'(n)})$ deux tris topologiques de $G$. Soit $i \in \{1, \ldots, n\}$ tel que $x_{\sigma'(i)} = x_{\sigma(1)}$.

      i. Montrer qu'il n'y a pas dans $G$ d'arête entre $x_{\sigma'(i)}$ et $x_{\sigma'(j)}$ pour $j \in \{1, \ldots, i\}$.

      ii. Montrer que
$$\lambda(x_{\sigma'(1)}) \ldots \lambda(x_{\sigma'(n)}) \sim \lambda(x_{\sigma'(i)})\lambda(x_{\sigma'(1)}) \ldots \lambda(x_{\sigma'(i-1)})\lambda(x_{\sigma'(i+1)}) \ldots \lambda(x_{\sigma'(n)})$$

      iii. En déduire que les linéarisations associées aux deux tris topologiques $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ et $(x_{\sigma'(1)}, \ldots, x_{\sigma'(n)})$ sont $\sim$-équivalentes.

5. Soit $u = a_1 \ldots a_n \in A^*$ avec $a_i \in A$ pour $i \in \{1, \ldots, n\}$. On définit l'ensemble de sommets $V_u = \{(a, i) \mid a \in A$ et $1 \leq i \leq |u|_a\}$ et l'ensemble des arêtes $E_u$ par $((a, i), (b, j)) \in E_u$ si $(a, b) \in D$ et si la $i^{ème}$ occurrence de $a$ est avant la $j^{ème}$ occurrence de $b$ dans $u$. On note $G_u = (V_u, E_u)$.

(a) Montrer que le graphe $G_u$ ainsi construit est un graphe de dépendance.

(b) Montrer que $u$ est une linéarisation de $G_u$.

(c) Montrer que si $v$ est un mot de $A^*$ tel que $v \sim u$ alors $G_u = G_v$.

# 5 Graphes et mots $\sim$-équivalents

1. Soient deux graphes de dépendance $G = (V, E, \lambda)$ et $G' = (V', E', \lambda')$ avec $V \cap V' = \emptyset$ (la définition de graphe de dépendance a été donnée dans les préliminaires de la Partie 4). On définit la concaténation de $G$ et de $G'$, notée $G \cdot G'$, par $G \cdot G' = (V \cup V', E \cup E' \cup F, \Lambda)$ avec $F = \{(x, x') \in V \times V' \mid (\lambda(x), \lambda'(x')) \in D\}$ et $\Lambda(x) =$ si $x \in V$ alors $\lambda(x)$ sinon $\lambda'(x)$.

   (a) Donner un algorithme qui, étant donnés $G$ et $G'$, calcule $G \cdot G'$.

   (b) Montrer que $G \cdot G'$ est un graphe de dépendance.

2. Etant donnés deux graphes de dépendance $G$ et $G''$, $G$ est dit préfixe de $G''$ s'il existe un graphe de dépendance $G'$ tel que $G \cdot G' = G''$. Ecrire un algorithme qui, étant donnés $G$ et $G''$, décide si $G$ est un préfixe de $G''$.

3. L'ensemble des graphes de dépendance associés à l'alphabet $A$ et à la relation d'indépendance $I$ est noté $\mathbb{G}(A, I)$. Soit $\varphi$ l'application de $A^*$ dans $\mathbb{G}(A, I)$ qui associe à tout mot $u$ de $A^*$ le graphe $G_u$ défini en 4.5.

   (a) Montrer que $\varphi$ est un homorphisme de $(A^*, \cdot)$ dans $(\mathbb{G}(A, I), \cdot)$.

   (b) Soit $\mathbb{M}(A, I)$ le quotient de l'ensemble $A^*$ par la congruence $\sim$. Montrer que $\varphi$ induit un isomorphisme de $(\mathbb{M}(A, I), \cdot)$ dans $(\mathbb{G}(A, I), \cdot)$.

# 6 Diagramme de Hasse

On définit une relation de récriture $\longrightarrow$ dans l'ensemble des graphes étiquetés sans cycles (la définition de graphe étiqueté a été donnée dans les préliminaires de la Partie 4) par $G = (V, E, \lambda) \longrightarrow G' = (V, E', \lambda)$ s'il existe une arête $(x, y) \in E$ telle que $E' = E \setminus \{(x, y)\}$ et qu'il existe un chemin de $x$ à $y$ dans $G$.

Un graphe $G$ est dit irréductible s'il n'existe pas de graphe $G'$ tel que $G \longrightarrow G'$.

1. Montrer que si $G, G'$ et $G''$ sont trois graphes étiquetés sans cycle tels que $G' \neq G''$, $G \longrightarrow G'$ et $G \longrightarrow G''$ alors il existe un graphe étiqueté sans cycle $\overline{G}$ tel que $G' \longrightarrow \overline{G}$ et $G'' \longrightarrow \overline{G}$.

2. La clôture transitive de la relation $\longrightarrow$ est notée $\xrightarrow{+}$. Soit $G$ un graphe étiqueté sans cycle qui ne soit pas irréductible. Montrer qu'il existe un unique graphe irréductible Hasse($G$), appelé le diagramme de Hasse de $G$, tel que $G \xrightarrow{+}$ Hasse($G$).

3. Ecrire un algorithme qui, étant donné un graphe de dépendance $G = (V, E, \lambda)$, calcule le diagramme de Hasse de $G$.

4. Ecrire un algorithme reconstruisant le graphe $G$ à partir de Hasse($G$).