MINISTÈRE
DE L'ÉDUCATION
NATIONALE
Liberté
Égalité
Fraternité

EAE INF 3

SESSION 2025

AGREGATION
CONCOURS EXTERNE

Section : INFORMATIQUE

ÉPREUVE SPÉCIFIQUE SELON L'OPTION CHOISIE :
- **ÉTUDE DE CAS INFORMATIQUE**
- **FONDEMENTS DE L'INFORMATIQUE**

Durée : 6 heures

L'usage de tout ouvrage de référence, de tout dictionnaire et de tout matériel électronique (y compris la calculatrice) est rigoureusement interdit.

Il appartient au candidat de vérifier qu'il a reçu un sujet complet et correspondant à l'épreuve à laquelle il se présente.

Si vous repérez ce qui vous semble être une erreur d'énoncé, vous devez le signaler très lisiblement sur votre copie, en proposer la correction et poursuivre l'épreuve en conséquence. De même, si cela vous conduit à formuler une ou plusieurs hypothèses, vous devez la (ou les) mentionner explicitement.

NB : Conformément au principe d'anonymat, votre copie ne doit comporter aucun signe distinctif, tel que nom, signature, origine, etc. Si le travail qui vous est demandé consiste notamment en la rédaction d'un projet ou d'une note, vous devrez impérativement vous abstenir de la signer ou de l'identifier.
Le fait de rendre une copie blanche est éliminatoire.

Tournez la page S.V.P.

INFORMATION AUX CANDIDATS

Vous trouverez ci-après les codes nécessaires vous permettant de compléter les rubriques figurant en en-tête de votre copie.

Ces codes doivent être reportés sur chacune des copies que vous remettrez.

► **Etude de cas informatique** :

Concours	Section/option	Epreuve	Matière
EAE	6200A	103	9424

► **Fondement de l'informatique** :

Concours	Section/option	Epreuve	Matière
EAE	6200A	103	9425

IMPRIMERIE NATIONALE – 25 0115 – D’après documents fournis

Transducteurs finis

Les transducteurs finis sont des machines théoriques similaires aux automates finis permettant de transformer des mots donnés en entrée. Ils sont utilisés pour l'analyse morphologique et phonologique dans le domaine de la linguistique, mais peuvent également avoir des applications en analyse syntaxique. Certains cas particuliers de transducteurs peuvent également avoir des applications dans des machines simples, tels que des distributeurs automatiques.

Le sujet est découpé en trois parties. La première présente les transducteurs séquentiels et contient quelques exemples et implémentations. La deuxième généralise le concept en introduisant les transducteurs sous-séquentiels. Enfin, la troisième et dernière partie établit deux théorèmes qui caractérisent les fonctions séquentielles et sous-séquentielles et leurs démonstrations.

Préliminaires

Notations Si Σ est un alphabet, on note Σ^* l'ensemble des mots sur Σ. Pour $u \in \Sigma^*$, on note $|u|$ la taille de u. L'unique mot de taille 0 sera noté ε, indépendamment de l'alphabet. Pour n un entier naturel, on note :

- Σ^n l'ensemble des mots de taille **exactement** n, défini par induction par $\Sigma^0 = \{\varepsilon\}$ et $\Sigma^{n+1} = \Sigma\Sigma^n$;
- $\Sigma^{\leqslant n}$ l'ensemble des mots de taille **au plus** n, c'est-à-dire $\Sigma^{\leqslant n} = \{\varepsilon\} \cup \Sigma^1 \cup \Sigma^2 \cup \ldots \cup \Sigma^n$.

On définit un **automate fini déterministe complet** comme un quintuplet $A = (Q, \Sigma, \delta, q_0, F)$ où :

- Q est un ensemble fini dont les éléments sont appelés **états** ;
- Σ est un alphabet appelé **alphabet d'entrée** ;
- δ est une fonction de $Q \times \Sigma$ dans Q appelée **fonction de transition** ;
- $q_0 \in Q$ est appelé **état initial** ;
- $F \subseteq Q$ est l'ensemble des **état finaux**.

On étend inductivement l'ensemble de définition de δ à $Q \times \Sigma^*$ par :

- $\delta(q, \varepsilon) = q$;
- si $u \in \Sigma^*$ et $a \in \Sigma$, alors $\delta(q, ua) = \delta(\delta(q, u), a)$.

On dit qu'un mot $u \in \Sigma^*$ est **reconnu** par l'automate A si et seulement si $\delta(q_0, u) \in F$. On note $L(A)$ l'ensemble des mots reconnus par A. Un langage L est dit **rationnel** (ou régulier) s'il existe un automate A tel que $L = L(A)$.

Dépendances. Ce sujet contient plusieurs parties. Chaque partie utilise des définitions et des résultats des parties précédentes. Sauf mention explicite du contraire, les questions restent néanmoins indépendantes, au sens où toute question peut être traitée en admettant les résultats énoncés dans les questions précédentes.

Attendus. Les questions de programmation doivent être traitées en langage OCaml. On pourra utiliser toutes les fonctions des modules `Array` et `List`, ainsi que les fonctions de la bibliothèque standard (celles qui s'écrivent sans nom de module, comme `max`, `incr` ainsi que les opérateurs comme `@`). L'utilisation d'autres modules est interdite.

Il est attendu des candidates et des candidats des réponses construites. Ils seront aussi évalués sur la précision, le soin et la clarté de la rédaction.

Partie I. Transducteurs séquentiels

Informellement, un transducteur séquentiel est un automate fini déterministe complet qui écrit un mot en sortie lors de la lecture d'un mot. Dans la définition suivante, il n'y a pas de notion d'état final.

On définit un **transducteur séquentiel** (ou simplement transducteur quand il n'y a pas d'ambiguïté) comme un sextuplet $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ tel que :

- Q est un ensemble fini dont les éléments sont appelés **états** ;
- Σ est un alphabet appelé **alphabet d'entrée** ;
- Γ est un alphabet appelé **alphabet de sortie** ;
- δ est une fonction de $Q \times \Sigma$ dans Q appelée **fonction de transition** ;
- λ est une fonction de $Q \times \Sigma$ dans Γ^* appelée **fonction de sortie** ;
- $q_0 \in Q$ est appelé **état initial**.

Comme pour un automate, on étend l'ensemble de définition de δ à $Q \times \Sigma^*$. On fait de même pour λ :

- $\lambda(q, \varepsilon) = \varepsilon$;
- si $u \in \Sigma^*$ et $a \in \Sigma$, alors $\lambda(q, ua) = \lambda(q, u)\lambda(\delta(q, u), a)$.

On appelle **transition de** T et on note $q \xrightarrow{a|v} q'$ un quadruplet (q, a, q', v) tel que $(q, a) \in Q \times \Sigma$, $\delta(q, a) = q'$ et $\lambda(q, a) = v$. L'interprétation d'une telle transition est « si, depuis l'état q, on lit la lettre a en entrée, alors on va vers l'état q', et on écrit le mot v en sortie ».

Un **calcul dans** T est une suite de transitions de la forme $p_0 \xrightarrow{a_0|v_0} p_1 \xrightarrow{a_1|v_1} \dots \xrightarrow{a_{k-1}|v_{k-1}} p_k$ telle que $p_i \xrightarrow{a_i|v_i} p_{i+1}$ est une transition de T pour $i \in [\![0, k-1]\!]$. On dit qu'un tel calcul est d'origine p_0, d'extrémité p_k, d'entrée $a_0 a_1 \dots a_{k-1}$ et de sortie $v_0 v_1 \dots v_{k-1}$.

Enfin, on définit la **fonction associée** à T par $\varphi_T : \begin{array}{rcl} \Sigma^* & \longrightarrow & \Gamma^* \\ u & \longmapsto & \lambda(q_0, u) \end{array}$. Autrement dit, $\varphi_T(u) = v$ si un calcul d'entrée u est de sortie v. On dit qu'une fonction $f : \Sigma^* \to \Gamma^*$ est **séquentielle** s'il existe un transducteur séquentiel T tel que $f = \varphi_T$.

Schématiquement, on représente un transducteur séquentiel comme dans la figure 1, qui représente le transducteur $T_0 = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ où $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{0, 1\}$ et δ et λ sont définies par les tableaux en figure 2. Sur ce transducteur, on a $\varphi_{T_0}(aabba) = 001010$ et $\varphi_{T_0}(baaab) = 01$.

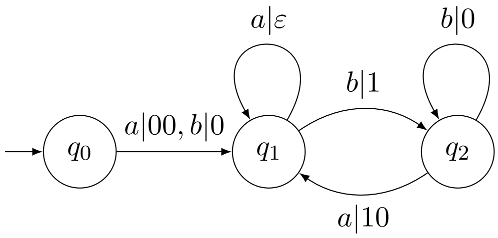

FIGURE 1 – Le transducteur séquentiel T_0

δ	a	b
q_0	q_1	q_1
q_1	q_1	q_2
q_2	q_1	q_2

λ	a	b
q_0	00	0
q_1	ε	1
q_2	10	0

FIGURE 2 – La table de transition et la table de sortie de T_0

I.1 Premiers exemples

Question 1 Pour T_0 le transducteur défini par la figure 1, déterminer $\varphi_{T_0}(bbbaaa)$. Déterminer un antécédent par φ_{T_0} de 01101.

Question 2 On considère le transducteur T_1 de la figure 3. Pour un mot $u \in \{a,b\}^*$ quelconque, décrire en français comment est construit $\varphi_{T_1}(u)$.

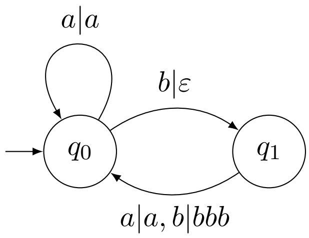

FIGURE 3 – Le transducteur séquentiel T_1

Question 3 Représenter graphiquement un transducteur séquentiel T sur $\Sigma = \Gamma = \{a,b\}$ dont la fonction séquentielle remplace toute séquence de a consécutifs en un seul a et ne modifie pas les séquences de b consécutifs. Justifier brièvement la correction de la construction.

Par exemple, on doit avoir $\varphi_T(aaaaabaaabbba) = ababbba$ et $\varphi_T(babba) = babba$.

Pour $b \in \mathbb{N} \setminus \{0,1\}$ et $n \in \mathbb{N}^*$, une représentation en base b de n est une suite notée $[a_0 a_1 \ldots a_{m-1}]_b$ telle que $m \in \mathbb{N}^*$, $\forall i \in [\![0, m-1]\!]$, $a_i \in [\![0, b-1]\!]$ et :

$$n = \sum_{i=0}^{m-1} a_i b^i$$

Si de plus on impose $a_{m-1} \neq 0$, alors cette suite est unique.

Par exemple, la représentation en base 4 de 237 est $[1323]_4$ car $237 = 1+3\times4+2\times16+3\times64$. Notons que nous représentons ici les chiffres de poids faible à gauche.

Question 4 Représenter graphiquement un transducteur sur $\Sigma = \{0,1,2,3\}$ et $\Gamma = \{0,1\}$ qui transforme un mot u représentant une écriture de $n \in \mathbb{N}$ en base 4 en un mot v représentant une écriture de n en base 2. Justifier brièvement la correction de la construction.

Question 5 Représenter graphiquement un transducteur sur $\Sigma = \{0,1\}$ et $\Gamma = \{0,1,2,3\}$ qui transforme un mot u représentant une écriture de $n \in \mathbb{N}$ en base 2, tel que u est de taille paire (terminant éventuellement par un 0), en un mot v représentant une écriture en base 4 de n. Justifier brièvement la correction de la construction.

Par exemple, l'image de 011110 doit être 132, car $[011110]_2 = [132]_4$.

I.2 Implémentation

On représente un transducteur séquentiel en OCaml en utilisant les types suivants :

```
type lettre = int
type mot = lettre list
type etat = int
type transducteur = (etat * mot) array array
```

On choisit de représenter un alphabet Σ par $\Sigma = \{0, 1, \ldots, |\Sigma| - 1\}$ (de même pour Γ). Un mot de Σ^* sera représenté par une liste d'entiers. Par exemple, l'objet `[0; 1; 1; 2; 1; 0]` représente le mot 011210 sur l'alphabet $\{0,1,2\}$.

Lorsque le transducteur séquentiel $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ est représenté par la matrice `t`, dont la première dimension est n et la deuxième dimension est p, on a :

- $|Q| = n$;
- $|\Sigma| = p$;
- $q_0 = 0$;
- pour tous $q \in Q$ et $a \in \Sigma$, `t.(q).(a)` est un couple `(q', v)` où `q'` représente $q' = \delta(q, a)$ et `v` représente $v = \lambda(q, a)$.

Par exemple, le transducteur séquentiel T_0 de la figure 1 peut être représenté par le morceau de code suivant :

```
let t0 = [| [|(1, [0; 0]); (1, [0])|];
            [|(1, []); (2, [1])|];
            [|(1, [1; 0]); (2, [0])|] |]
```

On a assimilé ici la lettre a à `0` et la lettre b à `1`.

Question 6 Écrire une fonction `calcul (t : transducteur) (u : mot) : mot` telle que si `t` est un transducteur séquentiel représentant $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ et `u` est un mot $u \in \Sigma^*$, alors `calcul t u` renvoie le mot v représentant $\lambda(q_0, u)$.

À T fixé, cette fonction devra avoir une complexité linéaire en $|u|$ et on demande de le justifier.

Question 7 Écrire une fonction `liste_mots (p : int) (m : int) : mot list` qui prend en argument un entier p et un entier m et renvoie une liste contenant tous les mots de taille m sur l'alphabet $\{0, 1, \ldots, p-1\}$ dans un ordre arbitraire.

Question 8 En déduire une fonction `antecedent (t : transducteur) (m : int) (v : mot) : mot option` telle que si `t` est un transducteur séquentiel représentant $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$, `m` est un entier naturel et `v` est un mot $v \in \Gamma^*$, alors `transducteur t m v` renvoie `Some u` où u est un mot le plus court possible vérifiant $|u| \leqslant m$ et $\varphi_T(u) = v$ s'il en existe un, et `None` sinon.

On n'impose aucune restriction sur la complexité temporelle de cette fonction.

I.3 Morphismes de mots

Soit $\Phi : \Sigma^* \longrightarrow \Gamma^*$. On dit que Φ est un **morphisme de mots** si et seulement si pour tous mots u et v de Σ^*, $\Phi(uv) = \Phi(u)\Phi(v)$.

Question 9 Soit $\Phi : \Sigma^* \longrightarrow \Gamma^*$ un morphisme de mots. Déterminer en justifiant la valeur de $\Phi(\varepsilon)$.

Question 10 Soit Φ et Ψ deux morphismes de mots de Σ^* dans Γ^*. Montrer que si pour tout $a \in \Sigma$, $\Phi(a) = \Psi(a)$, alors $\Phi = \Psi$.

Question 11 Soit $\Phi : \Sigma^* \longrightarrow \Gamma^*$ un morphisme de mots. Montrer que Φ est une fonction séquentielle.

Question 12 Soit T un transducteur séquentiel dont tous les états sont **accessibles** (c'est-à-dire que pour tout état $q \in Q$, il existe $u \in \Sigma^*$ tel que $\delta(q_0, u) = q$). Montrer qu'il y a équivalence entre :

1. φ_T est un morphisme de mots
2. pour tout $q \in Q$ et $a \in \Sigma$, $\lambda(q, a) = \lambda(q_0, a)$.

I.4 Machines de Mealy et de Moore

Un transducteur séquentiel $(Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ est appelé **machine de Mealy** si pour toute transition $q \xrightarrow{a|v} q'$, on a $|v| = 1$. C'est-à-dire que pour chaque lettre lue en entrée, la machine écrit une lettre en sortie. La fonction de sortie peut être vue comme définie de $Q \times \Sigma$ dans Γ.

Une machine de Mealy est appelée **machine de Moore** si de plus pour toute paire de transitions $q_1 \xrightarrow{a_1|b_1} q$ et $q_2 \xrightarrow{a_2|b_2} q$, on a $b_1 = b_2$. Autrement dit, la lettre écrite en sortie ne dépend que de l'état dans lequel on arrive lors de l'exécution d'une transition.

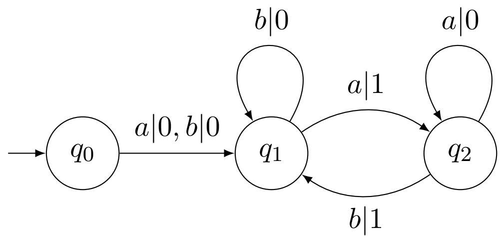

FIGURE 4 – Une machine de Mealy

On rappelle qu'une représentation en base b commence par les chiffres de poids faible.

Question 13 Représenter graphiquement une machine de Mealy sur $\Sigma = \Gamma = \{0, 1\}$ qui transforme un mot u représentant l'écriture de $x \in \mathbb{N}$ en base 2 en le mot v représentant l'écriture en base 2 de $2x$ modulo $2^{|u|}$. Justifier brièvement la correction de la construction.

Question 14 Représenter graphiquement une machine de Moore sur $\Sigma = \Gamma = \{0, 1\}$ qui transforme un mot u représentant l'écriture de $x \in \mathbb{N}$ en base 2 en le mot v représentant l'écriture en base 2 de $x + 1$ modulo $2^{|u|}$. Justifier brièvement la correction de la construction.

Question 15 Écrire une fonction `est_mealy (t : transducteur) : bool` qui détermine si un transducteur est une machine de Mealy ou non. Déterminer sa complexité temporelle dans le pire cas.

Question 16 Soit M une machine de Mealy. Montrer qu'il existe une machine de Moore M' telle que $\varphi_{M'} = \varphi_M$.

Une machine de Mealy est appelée **machine de Moore alternative** si pour toute paire de transitions $q \xrightarrow{a_1|b_1} q_1$ et $q \xrightarrow{a_2|b_2} q_2$, on a $b_1 = b_2$. Autrement dit, la lettre écrite en sortie ne dépend que de l'état duquel **on part** lors de l'exécution d'une transition. La définition est similaire à celle d'une machine de Moore, mais c'est cette fois-ci l'état de départ qui détermine la lettre en sortie.

Question 17 Montrer qu'il existe une machine de Mealy M à moins de deux états telle qu'aucune machine de Moore alternative n'est équivalente à M.

Partie II. Transducteurs sous-séquentiels

Dans cette partie, on étend la définition des transducteurs séquentiels en autorisant l'écriture d'un préfixe et d'un suffixe lors du calcul de l'image d'un mot.

II.1 Définition et exemples

On définit un transducteur sous-séquentiel comme un septuplet $(Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$ tel que $(Q, \Sigma, \Gamma, \delta, \lambda, q_0)$ est un transducteur séquentiel et ρ est une fonction de Q dans Γ^* appelée **fonction de suffixe**.

Si $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$ est un transducteur sous-séquentiel, on définit la fonction associée à T comme la fonction $\begin{array}{rcl}\varphi_T : \Sigma^* & \longrightarrow & \Gamma^* \\ u & \longmapsto & \lambda(q_0, u)\rho(\delta(q_0, u))\end{array}$. Autrement dit l'image de u est constituée de la sortie du calcul d'entrée u suivie du suffixe associé à l'état final du calcul.

On représente graphiquement un transducteur sous-séquentiel de la même façon qu'un transducteur séquentiel, en rajoutant des flèches sortantes étiquetées par les suffixes. Par exemple, la figure 5 représente un transducteur sous-séquentiel T_2 vérifiant $\rho(q_0) = 00$, $\rho(q_1) = \varepsilon$ et $\rho(q_2) = 1$.

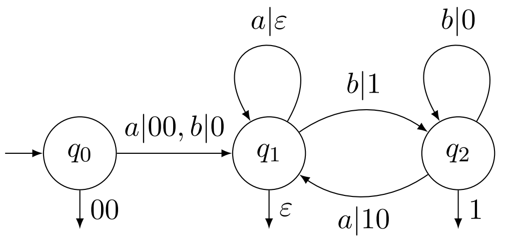

FIGURE 5 – Le transducteur sous-séquentiel T_2

On a par exemple $\varphi_{T_2}(\varepsilon) = 00$ et $\varphi_{T_2}(aabbaa) = 001010$.

Enfin, si f est une fonction de Σ^* dans Γ^*, on dit que f est une **fonction sous-séquentielle** s'il existe un transducteur sous-séquentiel T tel que $f = \varphi_T$.

Question 18 Expliquer comment modifier la machine de Moore de la question 14 en un transducteur sous-séquentiel qui fait une addition exacte, c'est-à-dire sans modulo.

Question 19 On considère le transducteur sous-séquentiel T_3 de la figure 6. Montrer que si u est un mot non vide qui représente une écriture de $x \in \mathbb{N}$ en base 2, alors $\varphi_{T_3}(u)$ représente une écriture de $3x$ en base 2.

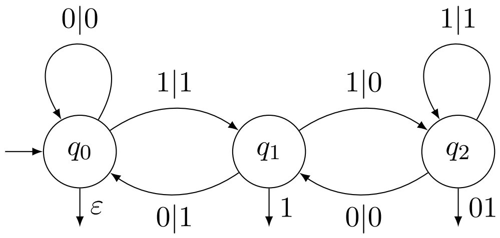

FIGURE 6 – Le transducteur sous-séquentiel T_3

Question 20 Représenter graphiquement sans justifier un transducteur sous-séquentiel à 5 états sur $\Sigma = \Gamma = \{0, 1\}$ qui transforme un mot u non vide représentant une écriture de $x \in \mathbb{N}$ en base 2 en un mot v représentant une écriture en base 2 de $5x$.

II.2 Langages rationnels

Dans cette sous-partie, on considère $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$ un transducteur sous-séquentiel.

Question 21 Montrer que si L est un langage rationnel sur Σ, alors $\varphi_T(L)$ est un langage rationnel sur Γ.

Indication : on pourra admettre qu'un automate fini dont les transitions sont étiquetées par des mots (plutôt que des lettres) reconnaît aussi un langage rationnel.

Question 22 Montrer que la réciproque de la question précédente est fausse.

Soit L un langage sur Γ. On appelle image réciproque de L par T, notée $\varphi_T^{-1}(L)$, l'ensemble des mots dont l'image par φ_T est dans L, c'est-à-dire $\varphi_T^{-1}(L) = \{u \in \Sigma^* \mid \varphi_T(u) \in L\}$.

Question 23 Montrer que si L est un langage rationnel sur Γ, alors $\varphi_T^{-1}(L)$ est un langage rationnel sur Σ. On explicitera un automate fini, construit à partir de T et d'un automate reconnaissant L, et on montrera qu'il reconnait exactement $\varphi_T^{-1}(L)$.

Indication : on commencera par traiter le cas où T est séquentiel.

II.3 Fonctions sous-séquentielles

Question 24 Soit $f : \Sigma^* \longrightarrow \Gamma^*$ et $g : \Gamma^* \longrightarrow \Lambda^*$ deux fonctions sous-séquentielles. Montrer que $g \circ f$ est une fonction sous-séquentielle.

On pourra commencer par traiter le cas de fonctions séquentielles.

Soit $f : \Sigma^* \longrightarrow \Gamma^*$ une fonction. On dit que f **conserve les préfixes** si $f(\varepsilon) = \varepsilon$ et si pour tous mots u et v de Σ^*, si u est un préfixe de v, alors $f(u)$ est un préfixe de $f(v)$.

Question 25 Montrer par un exemple qu'il existe une fonction sous-séquentielle qui ne conserve pas les préfixes.

Soit $f : \Sigma^* \longrightarrow \Gamma^*$ une fonction. On cherche à montrer l'équivalence entre les deux propositions suivantes :

(a) f est une fonction séquentielle.

(b) f est une fonction sous-séquentielle et f conserve les préfixes.

Question 26 Montrer l'implication (a) $\Rightarrow$ (b) de l'équivalence précédente.

Question 27 Soit $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$ un transducteur sous-séquentiel tel que φ_T conserve les préfixes. On suppose que tous les états de Q sont accessibles. Montrer que pour tout $q \in Q$ et $a \in \Sigma$, il existe un mot qu'on notera $\lambda'(q, a) \in \Gamma^*$ tel que $\rho(q)\lambda'(q, a) = \lambda(q, a)\rho(\delta(q, a))$.

Question 28 Avec les notations de la question précédente, on étend l'ensemble de définition de λ' à $Q \times \Sigma^*$ comme pour les fonctions de sortie des transducteurs. Montrer que pour $q \in Q$ et $u \in \Sigma^*$, $\rho(q)\lambda'(q, u) = \lambda(q, u)\rho(\delta(q, u))$.

Question 29 En déduire l'implication (b) $\Rightarrow$ (a) de l'équivalence précédente.

Partie III. Théorèmes de Ginsburg-Rose et Choffrut

III.1 Distance préfixe

Pour u et v deux mots de Σ^*, on note $u \wedge v$ le plus long préfixe commun à u et v. On définit la **distance préfixe** entre u et v par $d(u,v) = |u| + |v| - 2|u \wedge v|$. Par exemple, si $u = abbaba$ et $v = abbbbaaab$, alors $u \wedge v = abb$ et $d(u,v) = |u| + |v| - 2|u \wedge v| = 6 + 9 - 2 \times 3 = 9$.

Question 30 Montrer que la fonction de distance préfixe est une distance, c'est-à-dire que pour tous u, v, w mots de Σ^*, d vérifie les quatre propriétés suivantes :
- $d(u,v) = d(v,u)$;
- $d(u,v) \geqslant 0$;
- $d(u,v) = 0 \Leftrightarrow u = v$;
- $d(u,w) \leqslant d(u,v) + d(v,w)$.

Question 31 Soit $X \subseteq \Sigma^*$ un ensemble de mots non vide. Soit u le plus long préfixe commun à tous les mots de X. Montrer que pour $v \in X$, $d(u,v) \leqslant \max_{(x,y)\in X^2} d(x,y)$.

On dit qu'une fonction $f : \Sigma^* \longrightarrow \Gamma^*$ est **lipschitzienne** s'il existe un entier K strictement positif tel que :

$$\forall (u,v) \in \Sigma^* \times \Sigma^*, d(f(u), f(v)) \leqslant K d(u,v)$$

On cherche dans cette partie à montrer les théorèmes suivants :

- **Théorème de Ginsburg et Rose (1966)** : Soit $f : \Sigma^* \longrightarrow \Gamma^*$. La fonction f est une fonction séquentielle si et seulement si elle vérifie les trois propriétés suivantes :
  (a) f conserve les préfixes ;
  (b) pour tout langage rationnel L sur Γ^*, $f^{-1}(L)$ est rationnel ;
  (c) f est lipschitzienne.
- **Théorème de Choffrut (1978)** : Soit $f : \Sigma^* \longrightarrow \Gamma^*$. La fonction f est une fonction sous-séquentielle si et seulement si elle vérifie les deux propriétés suivantes :
  (a) pour tout langage rationnel L sur Γ^*, $f^{-1}(L)$ est rationnel ;
  (b) f est lipschitzienne.

Question 32 On suppose vrai le théorème de Choffrut. Montrer le théorème de Ginsburg et Rose.

Question 33 Soit $f : \Sigma^* \longrightarrow \Gamma^*$ une fonction sous-séquentielle. Montrer que f est lipschitzienne. En déduire le sens direct du théorème de Choffrut.

Indication : si $f = \varphi_T$ avec $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$, on posera $K = K_1 + 2K_2$, où K_1 et K_2 sont définies par $K_1 = \max\{|\lambda(q,a)| \mid q \in Q, a \in \Sigma\}$ et $K_2 = \max\{|\rho(q)| \mid q \in Q\}$.

La suite de cette partie a pour objectif de montrer le sens réciproque du théorème de Choffrut. On considère pour la suite une fonction $f : \Sigma^* \longrightarrow \Gamma^*$ qui vérifie :

(a) pour tout langage rationnel L sur Γ^*, $f^{-1}(L)$ est rationnel ;

(b) f est lipschitzienne.

On cherche à montrer que f est une fonction sous-séquentielle.

III.2 Découpage de $f(uv)$

Pour $u \in \Sigma^*$, on définit $\pi(u)$ comme le plus long préfixe commun aux mots $f(uv)$, pour $v \in \Sigma^{\leqslant 1}$. Ainsi, pour tout $v \in \Sigma^{\leqslant 1}$, il existe $\sigma_u(v)$ tel que :

$$f(uv) = \pi(u)\sigma_u(v)$$

Question 34 Montrer qu'il existe un entier naturel M tel que pour tout $u \in \Sigma^*$ et tous v_1, v_2 dans $\Sigma^{\leqslant 1}$, on a :

$$d(f(uv_1), f(uv_2)) \leqslant M$$

Question 35 En déduire que pour tout $u \in \Sigma^*$ et $v \in \Sigma^{\leqslant 1}$, $|\sigma_u(v)| \leqslant M$ puis que l'ensemble $S = \{\sigma_u \mid u \in \Sigma^*\}$ est fini.

Pour $s \in S$, on pose alors $X_s = \{u \in \Sigma^* \mid \sigma_u = s\}$.

III.3 Construction d'un transducteur sous-séquentiel

On admet temporairement que pour $s \in S$, l'ensemble X_s est un langage rationnel sur Σ.

Question 36 Soit L_1 et L_2 deux langages rationnels sur Σ. Montrer qu'il existe un automate fini déterministe complet $A = (Q, \Sigma, \delta, q_0, F)$ ainsi que deux sous-ensembles $Q_1 \subseteq Q$ et $Q_2 \subseteq Q$ tels que $L_1 = \{u \in \Sigma^* \mid \delta(q_0, u) \in Q_1\}$ et $L_2 = \{u \in \Sigma^* \mid \delta(q_0, u) \in Q_2\}$.

Un automate fini déterministe est dit **standard** si aucune transition ne pointe vers l'état initial.

Question 37 Montrer qu'il existe un automate fini déterministe complet et standard $A = (Q, \Sigma, \delta, q_0, F)$ tel qu'en posant, pour tout $s \in S$, $Q_s = \{\delta(q_0, u) \mid u \in X_s\}$, alors on a :

- $\forall s, s' \in S, Q_s \cap Q_{s'} \neq \emptyset \Leftrightarrow s = s'$;
- $\bigcup_{s \in S} Q_s = Q$.

Pour $q \in Q$, on pose $U_q = \{u \in \Sigma^* \mid \delta(q_0, u) = q\}$. On pose de plus $\beta(q)$ le plus long suffixe des $\pi(u)$, pour $u \in U_q$. Il existe donc une fonction α telle que pour $u \in U_q$:

$$\pi(u) = \alpha(u)\beta(q)$$

On veut montrer, par disjonction de cas, que pour tout $u \in U_q$ et $a \in \Sigma$, $\alpha(u)$ est un préfixe de $\alpha(ua)$. Pour cela, on note $s \in S$ tel que $q \in Q_s$, $q' = \delta(q, a)$ et s' tel que $q' \in Q_{s'}$.

Question 38 Montrer que $\alpha(u)\beta(q)s(a) = \alpha(ua)\beta(q')s'(\varepsilon)$.

On suppose pour les deux questions suivantes que $|s(a)| \leqslant |s'(\varepsilon)|$.

Question 39 Montrer qu'il existe $v \in \Gamma^*$ tel que $\pi(u) = \pi(ua)v$, et que v ne dépend pas du mot u choisi dans U_q.

Question 40 Montrer que nécessairement $|\beta(q')v| \leqslant |\beta(q)|$, puis qu'il existe $w \in \Gamma^*$ tel que $\alpha(u)w = \alpha(ua)$, et que w ne dépend pas du mot u choisi dans U_q.

On admet que le cas $|s(a)| > |s'(\varepsilon)|$ se traite de manière similaire. En conclusion, il existe une fonction $\lambda : Q \times \Sigma \to \Gamma$ telle que pour tout $q \in Q$, $u \in U_q$ et $a \in \Sigma$, $\alpha(u)\lambda(q, a) = \alpha(ua)$.

On pose, pour $q \in Q$ tel que $q \in X_s$, $\rho(q) = \beta(q)s(\varepsilon)$.

Question 41 En considérant le transducteur sous-séquentiel $T = (Q, \Sigma, \Gamma, \delta, \lambda, q_0, \rho)$, montrer le théorème de Choffrut.

III.4 Rationalité des X_s

On cherche dans cette partie à montrer que pour tout $s \in S$, l'ensemble X_s est un langage rationnel. Pour cela, on définit successivement plusieurs familles intermédiaires de langages permettant de reconstruire X_s.

Pour $i \in \{0, 1, \ldots, 2M\}$ et $z \in \Gamma^M$, l'entier M étant celui définit à la question 34, on définit $B(i, z)$ par :

$$B(i, z) = \{x \in \Gamma^* \mid |x| \equiv i \pmod{2M + 1} \text{ et } (x \in \Gamma^* z \text{ ou } z \in \Gamma^* x)\}$$

c'est-à-dire que $x \in B(i, z)$ si et seulement si la taille de x est de la forme $|x| = (2M+1) \times k + i$ avec k entier, et x est un suffixe de z ou que z est un suffixe de x.

Question 42 Montrer que $B(i, z)$ est un langage rationnel sur Γ.

Pour $i \in \{0, 1, \ldots, 2M\}$, $z \in \Gamma^M$, $s \in S$ et $v \in \Sigma^{\leqslant 1}$, on définit $C_s(i, z, v)$ par :

$$C_s(i, z, v) = \{u \in \Sigma^* \mid f(uv) \in B(i, z)s(v)\}$$

Question 43 Montrer que $C_s(i, z, v)$ est un langage rationnel sur Σ.

Enfin, pour $s \in S$, on définit Y_s par :

$$Y_s = \bigcup_{i=0}^{2M} \bigcup_{z \in \Gamma^M} \bigcap_{v \in \Sigma^{\leqslant 1}} C_s(i, z, v)$$

Le résultat de la question 43 et les propriétés de stabilité des langages rationnels garantissent que Y_s est un langage rationnel sur Σ.

Question 44 Montrer que pour $s \in S$, $X_s \subseteq Y_s$.

On veut maintenant démontrer que $Y_s \subseteq X_s$. On considère $y \in Y_s$, et $i \in \{0, 1, \ldots, 2M\}$, $z \in \Gamma^M$ tels que $y \in \bigcap_{v \in \Sigma^{\leqslant 1}} C_s(i, z, v)$.

Question 45 Soient $v_1, v_2 \in \Sigma^{\leqslant 1}$ et $b_1, b_2 \in B(i, z)$ tels que $f(yv_1) = b_1 s(v_1)$ et $f(yv_2) = b_2 s(v_2)$. Montrer que $|b_1| = |b_2|$. Pour la suite, on pose n_s cette taille commune.
Indication : on pourra utiliser, après l'avoir montrée, l'inégalité $||u| - |v|| \leqslant d(u, v)$.

Question 46 Montrer qu'il existe $v_1, v_2 \in \Sigma^{\leqslant 1}$ tels que $s(v_1) \wedge s(v_2) = \varepsilon$. En déduire que $|\pi(y)| \leqslant n_s$.

On admet que, de même, il existe $v_1, v_2 \in \Sigma^{\leqslant 1}$ tels que $\sigma_y(v_1) \wedge \sigma_y(v_2) = \varepsilon$.

Question 47 En considérant $v_1, v_2 \in \Sigma^{\leqslant 1}$ tels que $\sigma_y(v_1) \wedge \sigma_y(v_2) = \varepsilon$, montrer que $s(v_1)$ est un suffixe de $\sigma_y(v_1)$ et que $s(v_2)$ est un suffixe de $\sigma_y(v_2)$. En déduire que $|\pi(y)| = n_s$.

Question 48 En déduire que $Y_s \subseteq X_s$.

* *
*