Épreuve orale d'Informatique Fondamentale

ULC MP 2012

Exemples de sujets

Patrick Baillot, Anne Bouillard, Alexis Saurin

Ce document consiste en une sélection, à titre d'exemples, de 3 sujets proposés à l'épreuve d'informatique fondamentale ULC MP en 2012. Cette épreuve consiste en un oral de 45 minutes sans préparation. Le lecteur pourra se référer au rapport de jury pour plus d'informations sur le déroulement de l'épreuve.

1 Autour des mots de Lukasiewicz

Soit $A = \{a_0, a_1, \ldots, a_n, \ldots\}$, $n \in \mathbb{N}$ un alphabet infini et δ une fonction de pondération telle que pour tout $n \in \mathbb{N}$, $\delta(a_n) = n - 1$. Si $w = b_1 \ldots b_m \in A^*$ est un mot, le poids de w est la somme des poids des lettres qui le composent : $\delta(w) = \sum_{i=1}^{m} \delta(b_i)$.

Un mot $w \in A^*$ est dit de **Lukasiewicz** s'il vérifie les propriété suivantes :

1. $\delta(w) = -1$.
2. Pour tout préfixe propre y de w, $\delta(y) \geq 0$.

On note L l'ensemble de mots de Lukasiewicz.

Question 1. Les mots suivants sont-ils des mots de Lukasiewicz ?
- $a_2 a_1 a_0 a_2 a_0 a_1 a_0$
- $a_3 a_2 a_0 a_2 a_0 a_1 a_1 a_0$

Question 2. Caractériser $L \cap \{a_0, a_1\}^*$. Est-il rationnel ?

Question 3. Le langage $L \cap \{a_0, a_2\}^*$ est-il rationnel ?

On considère maintenant un système de réécriture. On se donne des variables X et des règles de transformation :

$$\forall n \in \mathbb{N},\ X \to a_n X \ldots X \qquad\qquad (= a_n X^n \text{ ; } X \text{ répété } n \text{ fois}).$$

Un mot est **engendré** par ce système de réécriture s'il peut s'obtenir à partir de X par application successives de règles de réécriture. Par exemple, le mot $a_2 a_0 a_1 a_0$ est obtenu par

$$X \to a_2 X X \to a_2 X a_1 X \to a_2 a_0 a_1 X \to a_2 a_0 a_1 a_0.$$

On note alors $X \to^* a_2 a_0 a_1 a_0$ pour signifier que $a_2 a_0 a_1 a_0$ peut être obtenu à partir de X en un nombre quelconque d'étapes.

Plus généralement, pour $u, v \in (A \cup \{X\})^*$, on peut appliquer une règle qui transforme u et v si $u = u_1 X u_2$ et $v = u_1 a_n X^n u_2$, et on note $u \to v$. On note aussi $u \to^* v$ si v est obtenu à partir de u en appliquant un nombre arbitraire de règles de réécriture. En particulier, on a toujours $u \to^* u$.

Question 4. Montrer que $a_2 a_1 a_0 a_2 a_0 a_1 a_0 \in L$ en donnant les règles de réécritures à appliquer à partir de X.

Question 5. Soit $w \in (A \cup \{X\})^*$. Montrer que si $XX \to^* w$, alors il existe w_1, w_2 tels que $w = w_1 w_2$, $X \to^* w_1$ et $X \to^* w_2$.

Question 6. Supposons que $X \to^* w \in A^*$. Montrer que $w \in L$.

Question 7. Montrer que si $w \in L$, alors $X \to^* w$.

Un **arbre binaire** est défini récursivement comme

$$\text{ArbreBin} = \text{Feuille} \;+\; \text{ArbreBin} \;\times\; \text{Nœud} \;\times\; \text{ArbreBin}.$$

Un **arbre planaire enraciné** est un arbre dont on distingue la racine r, dont les nœuds ont un nombre arbitraires de fils ordonnés (l'ordre dans lequel on les représente importe) :

$$\text{Arbre} = \text{Nœud} \;+\; \text{Nœud} \;\times\; \text{Arbre} + \;\text{Nœud} \;\times\; \text{Arbre} \times\; \text{Arbre}...$$

Question 8. Montrer que les mots de Lukasiewicz sont en bijection avec les arbres planaires enracinés.

Question 9. Donner une relation entre le nombre d'arbres planaires enracinés de taille n et le nombre d'arbres binaires de taille $n-1$ (la taille d'un arbre est son nombre de Nœuds).

Question 10. En déduire le nombre de mots de L de longueur n.

2 Facteurs itérants d'un langage rationnel

Soient A un alphabet fini et L un langage de A^*. Un mot v est un *facteur itérant* de L s'il existe deux mots u, w tels que : $uv^*w \subseteq L$. On désigne par $\mathcal{I}(L)$ l'ensemble des facteurs itérants de L. D'autre part, si $k \geq 2$ soit $\mathcal{R}_k(L) = \{w \in A^* \backslash\ w^k \in L\}$ le langage des racines k-èmes de L.

L'objet de cet exercice est de montrer que si L est rationnel alors $\mathcal{R}_k(L)$ et $\mathcal{I}(L)$ sont aussi des langages rationnels, et de construire des automates les reconnaissant.

On représentera ici les automates déterministes sous la forme $\mathcal{A} = (Q, A, \delta, q_1, F)$ où q_1 est l'état initial et F est l'ensemble des états acceptants. On rappelle qu'un état q de l'automate est dit *utile* s'il existe un chemin de l'état initial à q, et un chemin de q à un état acceptant.

Étant donné un tel automate, un état q et un mot w, on notera $q \cdot w$ l'état (s'il existe) atteint par exécution de $\mathcal{A}$ depuis q, avec le mot w. On dit que l'automate $\mathcal{A}$ est *régulier à gauche* si :

$$\forall u_1, u_2 \in A^*, q_1 \cdot u_1 = q_1 \cdot u_2 \Rightarrow (\forall v \in A^*,\ q_1 \cdot vu_1 = q_1 \cdot vu_2).$$

On va chercher à montrer que tout automate déterministe complet est équivalent à un automate déterministe complet régulier à gauche.

Soit $\mathcal{A} = (Q, A, \delta, q_1, F)$ un automate déterministe complet et $Q = \{q_1, \ldots, q_n\}$. On va définir un automate dont l'ensemble des états est inclus dans Q^n, l'ensemble des vecteurs de longueur n à valeurs dans Q. Si $r \in Q^n$ on écrira r_i ou r_{q_i} sa i-ème composante. On note I le vecteur défini par $\forall j \in \{1, \ldots, n\},\ I_j = q_j$.

On définit alors $\mathcal{A}_g = (Q^n, A, \gamma, I, F_g)$ par :

- pour $a \in A$ et $r \in Q^n$, $\gamma(a, r) = s$ avec $\forall j \in \{1, \ldots, n\},\ s_j = \delta(a, r_j)$,
- $F_g = \{r \in Q^n,\ r_1 \in F\}$.

On considérera en fait juste les états *accessibles* de $\mathcal{A}_g$ (c'est-à-dire qui peuvent être atteints depuis l'état initial) et on notera encore $\mathcal{A}_g$ l'automate correspondant.

Question 11. Soit $\mathcal{D} = (Q, A, \delta, q_1, F)$ défini par $Q = \{q_1, q_2, q_3\}$, $A = \{a, b\}$, $F = \{q_3\}$, et δ donné par :

	q_1	q_2	q_3
a	q_2	q_1	q_2
b	q_1	q_3	q_1

L'automate $\mathcal{D}$ est-il régulier à gauche ? Construire l'automate $\mathcal{D}_g$.

Question 12. Montrer que si $\mathcal{A}$ est un automate déterministe complet alors $\mathcal{A}_g$ est un automate déterministe complet équivalent à $\mathcal{A}$ et régulier à gauche.

Soit $\mathcal{A} = (Q, A, \delta, q_1, F)$ un automate déterministe reconnaissant L. On note V_2 le sous-ensemble de Q défini par :

$$V_2 = \{q \in Q \setminus \exists w \in A^*\ q_1 \cdot w = q \text{ et } q_1 \cdot ww \in F\}.$$

Soit $\mathcal{A}_{\mathcal{R}_2} = (Q, A, \delta, q_1, V_2)$, c'est-à-dire qu'il est obtenu à partir de $\mathcal{A}$ en prenant V_2 comme ensemble d'états acceptants.

Question 13. Montrer que si $\mathcal{A}$ est régulier à gauche, alors :

$$L(\mathcal{A}_{\mathcal{R}_2}) = \mathcal{R}_2(L(\mathcal{A})). \tag{1}$$

Question 14. Montrer qu'on peut calculer l'ensemble V_2.

Question 15. Généraliser le résultat précédent en donnant la construction, pour $k \geq 3$, d'un automate $\mathcal{A}_{\mathcal{R}_k}$ qui reconnaît le langage $\mathcal{R}_k(L(\mathcal{A}))$.

Soit L un langage rationnel. On en vient maintenant au sujet du langage $\mathcal{I}(L)$ des facteurs itérants de L.

Soit $\mathcal{A} = (Q, A, \delta, q_1, F)$ un automate déterministe reconnaissant L.

Question 16. Étant donné $p \in Q$, soit $\mathcal{A}_p = (Q, A, \delta, p, \{p\})$. Montrer que pour tout état p utile on a :

$$L(\mathcal{A}_p) \subseteq \mathcal{I}(L).$$

Question 17. Montrer que si v est un facteur itérant de L, alors il existe un entier $k \geq 1$ et un état p utile tels que v^k appartient à $L(\mathcal{A}_p)$.

Question 18. Montrer que $\mathcal{I}(L)$ est reconnu par un automate fini qu'on peut effectivement construire à partir de $\mathcal{A}$.

3 Logique et théorie des graphes

Dans ce problème, on va d'abord établir un résultat fondamental de la logique propositionnelle, le ***théorème de compacité***, puis on l'appliquera à la théorie des graphes.

Théorème de compacité du calcul des propositions.

Question 19. Rappeler la définition des formules de la logique propositionnelle, de la notion de valuation et de satisfaisabilité.

Le théorème de compacité affirme que :

Théorème 1 (de Compacité) *Soit $\mathfrak{T}$ un ensemble de formules propositionnelles. On a l'équivalence entre les deux assertions suivantes :*

1. *$\mathfrak{T}$ est satisfaisable ;*
2. *toute partie finie de $\mathfrak{T}$ est satisfaisable.*

Question 20. Montrer $1 \Rightarrow 2$.

Question 21. Le cas où $\mathfrak{T}$ est fini est évident. Pourquoi ?

Question 22. Dans cette question, on admet le théorème de compacité. Que peut-on dire d'un ensemble infini de formules propositionnelles $\mathfrak{T}$ qui n'est pas satisfaisable ?

On introduit la notion de ***valuation finie adaptée*** à un ensemble de formules : On considère un ensemble $\mathfrak{F}$ de formules propositionnelles, une énumération des variables propositionnelles $(P_i)_{i\in\mathbb{N}}$ et une valuation σ d'une partie finie $(P_i)_{i\leq n}$ des variables propositionnelles. On dit que σ est une ***valuation finie adaptée à*** $\mathfrak{F}$ si elle vérifie que toute partie finie $\mathfrak{F}' \subseteq \mathfrak{F}$ peut être satisfaite par un prolongement de σ à l'ensemble des variables propositionnelles apparaissant dans $\mathfrak{F}'$.

Question 23. Montrer que si $\mathfrak{T}$ est satisfaisable, toute valuation finie adaptée à $\mathfrak{T}$, définie sur n variables propositionnelles peut se prolonger en une valuation finie définie sur $n+1$ variables propositionnelles.

Question 24. Démontrer le théorème de compacité.

La suite du problème va consister à appliquer le théorème de compacité à la théorie des graphes.

Graphes. Un ***graphe*** $\mathcal{G}$ est la donnée d'un ensemble au plus dénombrable $\mathcal{S}_\mathcal{G}$, les sommets du graphe, et d'un ensemble $\mathcal{A}_\mathcal{G} \subseteq \{a \in \mathcal{P}(\mathcal{S}_\mathcal{G}), \#a = 2\}$ de paires non orientées de sommets, les arêtes du graphe. Dans le cas où $\mathcal{S}_\mathcal{G}$ est fini, on parlera de ***graphe fini*** et de ***graphe infini*** sinon.

On représentera les sommets d'un graphe $\mathcal{G}$ par des points du plan et les arêtes $a = \{s, t\}$ par une ligne joignant les deux sommets s et t.

Un graphe ***complet*** est un graphe $\mathcal{G}$ tel que toute paire de sommets disjoints est reliée par une arête (c'est-à-dire $\mathcal{A}_\mathcal{G} = \{a \in \mathcal{P}(\mathcal{S}_\mathcal{G}), \#a = 2\}$).

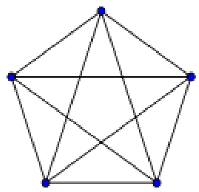

FIGURE 1 – K_5, graphe complet à 5 sommets.

Question 25. Donner des exemples illustrant les notions de graphe fini, infini et de graphe complet.

Graphes rouge et bleu. Dans la suite, on ne considèrera plus que des graphes complets dont les arêtes sont colorées soit en bleu, soit en rouge. Formellement, on se donne une application c de $\mathcal{A}_\mathcal{G}$ dans $\{\text{rouge}, \text{bleu}\}$. Un graphe sera dit monochrome si toutes ses arêtes ont la même couleur.

On va s'intéresser à l'existence de sous-graphes complets monochromes (on dira SGCM pour faire court) dans des graphes finis ou infinis.

Question 26. Montrer que dans tout graphe, l'intersection d'un SGCM rouge et d'un SGCM bleu est au plus réduite à un sommet.

Question 27. Montrer que tout graphe à 6 sommets contient un SGCM de taille 3.

Existence de SGCM. Dans la suite du problème, on va s'intéresser à deux généralisations du résultat de la question précédente :

Théorème 2 (SGCM infinis) *Tout graphe rouge et bleu infini contient un SGCM infini.*

Théorème 3 (SGCM finis) *Pour tout entier $n \in \mathbb{N}$, il existe un entier $k \in \mathbb{N}$ tel que tout graphe ayant au moins k sommets, contient un SGCM de taille n.*

La question précédente nous indique donc que pour $n = 3$, on peut choisir $k = 6$. On va d'abord démontrer le théorème pour les graphes finis se déduit du cas infini par le théorème de compacité puis on démontrera le cas infini.

Question 28. En admettant le théorème des SGCM infinis, démontrer le théorème des SGCM finis en utilisant le théorème de compacité.

Question 29. Démontrer le théorème des SGCM infinis.