

REQUIN

REcueil de QUestions d'INformatique

Niveaux de difficultés

- 🐼 : Question d'introduction. Prend entre quelques secondes et 1 minutes. Faisable de tete, il s'agit souvent de trouver un exemple ou un contre-exemple.
- 🐱 : Question demandant des méthodes classique, mais sans autre difficulté apparante. Très bien pour réviser le cours.
- 🧠 : Question relativement difficile, qui peut nécessiter un peu de travail et/ou des astuces relativement classique. Toutefois, avec un peu de persévérance, tout le monde y est capable. Très bien pour réviser des exercices. Il est fortement conseiller de les poser sur feuille.
- 🗨️ : Question difficile du sujet, qui peut prendre entre une dizaine de minute pour les plus fort et plusieurs heures. Parfois, elles demandent de sortir des astuces peu communes. On ce doit de les poser sur feuille.
- 🕒 : Questions très très difficiles demandant beaucoup de recherche / connaissances externes. Demande plusieurs jours de travail.
- 🌐 : Questions ouvertes à la connaissance des autres.

Sommaire

Chapitre I. Algorithmique	3
Chapitre II. Arbres & Graphes	4
Problème II.1: Mots univers	5
Chapitre III. Langages formels	6
Problème III.1: Language permuté et inclusions	7
Problème III.2: Language continuables	8
Problème III.3: Puissance et racine de langages	9
Chapitre IV. Théorie des jeux	10
Problème IV.1: Nim à choix	11
Chapitre V. Calculabilité	12
Problème V.1: Calculabilité et représentation d'ensembles infinis	13
Chapitre VI. Logique	14
Problème VI.1: Compacité	15
Chapitre VII. Langages fonctionnels	16
Chapitre VIII. Mathématiques pour l'informatique	17
Problème VIII.1: Monoïdes libres, langages et actions	18


Chapitre I. Algorithmique

c'est si vide ici...


Chapitre II. Arbres & Graphes

Problème II.1: Mots univers

On dit qu'un mot $w \in \Sigma^*$ est n -univers si tout les mots de Σ^n sont des facteurs de w . On s'intéresse à créer les plus petits mots n -univers.


 **Question 0** Montrer qu'un mot n -univers sur un alphabet à k lettres à au moins une longueur de $k^n + n - 1$


Soit $G = (V, E)$ un graphe **orienté**, on définit $L(G)$ le *graphe ligne* de G par le graphe orienté (E, E') avec E' l'ensemble des arêtes de la forme $((x, y), (y, z))$ pour $(x, y), (y, z) \in E$.

 **Question 1** Donner le graphe ligne du cycle à 4 éléments et d'un arbre binaire parfait de hauteur 2.


On construit alors la famille des graphes de Bruijn $(DB(n))_{n \in \mathbb{N}^*}$ par $DB(1) = (\{0, 1\}, \{0, 1\}^2)$ et $DB(n+1) = L(DB(n))$.


 **Question 2** Construire $DB(2)$

 **Question 3** Montrer que pour tout $n \in \mathbb{N}^*$, chaque sommet de $DB(n)$ à autant d'arêtes sortantes que entrantes. Combien de sommets et d'arêtes $DB(n)$ possède t'il ?

 **Question 4** Montrer que pour tout graphe orienté fortement connexe tel que pour tout sommet le degré entrant est le même que le degré sortant, il existe un cycle eulérien (un cycle passant par toutes les arêtes du graphe).

En déduire que pour tout $n \in \mathbb{N}^*$, $DB(n)$ possède un cycle eulérien.

 **Question 5** En voyant les sommets de $DB(n)$ comme des mots dans $\{0, 1\}^{n-1}$, et en étiquetant les arêtes par 0 ou 1, montrer qu'il existe un mot n -univers sur l'alphabet $\{0, 1\}$ de taille $2^n + n - 1$


 **Question 6** Généraliser la question précédente pour des alphabets plus grand.


Chapitre III. Langages formels


Problème III.1: Language permuté et inclusions


Soit Σ un alphabet. Pour $w \in \Sigma^*$ un mot et $\alpha \in \Sigma$ une lettre, on note $|w|_\alpha$ le nombre d'occurrence de α dans w . Pour L un langage sur Σ , on pose $\sigma(L) = \{w \in \Sigma^* \mid \exists u \in L, \forall \alpha \in \Sigma, |u|_\alpha = |w|_\alpha\}$ la *permutation* de L .


On dira que L est strictement hors-contexte si L est hors-contexte et n'est pas régulier.

 **Question 0** Montrer que $L_1 = \{a^n b^n : n \in \mathbb{N}\}$ n'est pas régulier mais qu'il est hors-contexte. Que vaut $\sigma(L_1)$?

 **Question 1** Donner un langage L tel que L soit strictement hors contexte mais $\sigma(L)$ régulier.

 **Question 2** Donner un langage L' tel que L' soit régulier mais $\sigma(L')$ strictement hors-contexte.

 **Question 3** Montrer que si L est tel que $L_1 \subseteq L \subseteq \sigma(L_1)$, alors L n'est pas régulier.

 **Question 4** Est-ce qu'il existe L' un langage strictement hors-contexte tel que $\sigma(L')$ est strictement hors-contexte et tel qu'il existe L régulier tel que $L' \subseteq L \subseteq \sigma(L')$?

Problème III.2: Language continuables

Soit $\Sigma = \{a, b\}$. On dit que $w \in \Sigma^*$ est *primitif* s'il n'existe pas de mot $u \in \Sigma^*$ et de $p > 1$ tels que $w = u^p$

Un langage L est dit *continuable* si pour tout $u \in \Sigma^*$, il existe un $v \in \Sigma^*$ tel que $uv \in L$



Question 0 Pour chacun des cas suivants donner des exemples de langage sur $\Sigma = \{a, b\}$:

- Un langage infini régulier ne reconnaissant aucun mot primitif
- Un langage infini régulier ne reconnaissant que des mots primitifs
- Un langage infini régulier continuable
- Un langage infini algébrique continuable ne reconnaissant que des mots primitifs



Question 1 Proposer un algorithme pour tester si un mot est primitif en $O(|w|^{\frac{3}{2}})$. C'est possible de le faire en $O(|w|)$



Question 2 Étant donné un automate A , proposer un algorithme pour déterminer si le langage reconnu par A est continuable.



Question 3 Montrer que tout langage régulier continuable sur $\Sigma = \{a, b\}$ contient une infinité de mots primitifs. Quelle est la condition sur Σ pour que cela soit vrai ?



Question 4 Existe-t'il un langage infini rationnel continuable ne comportant que des mots primitifs?


Indication: Considérer l'ensemble des applications partielles $\{\delta^(u, _) : u \in \Sigma^*\}$, et trouver un bon groupe pour la composition inclut dans cet ensemble.*


δ^* est définie par $\delta^*(ua, q) = \delta(a, \delta^*(u, q))$ et $\delta^*(\varepsilon, q) = q$ pour $(a, u, q) \in \Sigma \times \Sigma^* \times Q$

Problème III.3: Puissance et racine de langages

Soit Σ un alphabet possédant au moins 2 lettres. Pour L un langage sur Σ et $k > 1$ on pose


$$L^{(k)} = \{w^k \mid w \in L\} \text{ et } L^{1/k} = \{w \mid w^k \in L\}$$

 **Question 0** Calculer $L^{1/2}$ pour L reconnu par l'expression régulière $ab(\Sigma\Sigma)^*$


 **Question 1** Pour $k, l \geq 1$, montrer que


- $(L^{(k)})^{(l)} = L^{(kl)}$
- $(L^{1/k})^{1/l} = L^{1/kl}$
- $(L^{1/k})^{(k)} \subseteq L$


 **Question 2** Donner un langage rationnel L tel que $\forall k \geq 2$, $L^{(k)}$ n'est pas rationnel


 **Question 3** Montrer que pour L reconnu par un automate $A = (\Sigma, Q, q_i, \delta, F)$, on a


$$w \in L^{1/2} \Leftrightarrow \exists q \in Q, \begin{cases} \delta(q_i, w) = q \\ \delta(q, w) \in F \end{cases}$$

 **Question 4** Montrer que si L rationnel, alors $L^{1/2}$ aussi.

 **Question 5** Montrer que soit $k \in \mathbb{N}$, si L est rationnel, alors $L^{1/k}$ aussi.

 **Question 6** Si L est rationnel, est-ce forcément aussi le cas de $\bigcup_{k \geq 1} L^{1/k}$?

 **Question 7** Donner un algorithme qui détermine si un langage rationnel L respecte $L = (L^{1/2})^{(2)}$


 **Question 8** Montrer que si L est rationnel, alors $\text{Root}(L) = \{w \in \Sigma^* : w^{|w|} \in L\}$ l'est aussi.

Chapitre IV. Théorie des jeux


Problème IV.1: Nim à choix

Soit $A \subseteq \mathbb{N}^*$ avec $1 \in A$. On considère un jeu à deux joueurs où $N > 0$ objets sont disposés sur une table, et chaque joueur doit à tour de rôle retirer $t \in A_{>N}$ objets de la table. Le joueur qui retire le dernier objet perd.

On appellera Alice le joueur qui commence et Bob le deuxième joueur.

 **Question 0** Pour les valeurs de A suivantes, pour quel N Alice possède-t-elle une stratégie gagnante ?

- $A = \mathbb{N}$
- $A = \{1\}$
- $A = \{2k + 1 \mid k \in \mathbb{N}\}$
- $A = \{1, 2, \dots, p\}$ pour p fixé

 **Question 1** Si $\max(A) \leq N$, proposer un algorithme qui décide si Alice possède une stratégie gagnante pour un N donné.

Chapitre V. Calculabilité

Problème V.1: Calculabilité et représentation d'ensembles infinis

On se propose ici de créer une structure de donnée permettant de représenter un ensemble infini sous la forme d'une fonction de `'a -> bool` déterministe et qui **termine toujours**. On fixe dans cet exercice X un ensemble quelconque. On définit donc le type suivant en OCaml :

```
1 type 'a set = 'a -> bool;;
```

ml

Question 0 Expliciter la bijection entre $\mathcal{P}(X)$ et $\{0, 1\}^X$



Question 1 Donner une fonction `make_finite: 'a list -> 'a set` qui à une liste finie renvoie l'ensemble de ses valeurs



Question 2 Construire des objets de type `int set` pour représenter les ensembles suivants :

- L'ensemble des nombres pairs
- \mathbb{P} , l'ensemble des nombres premiers
- L'image d'une fonction fixée f positive et strictement croissante



Question 3 Donner le code d'une fonction `union` qui réalise l'union de deux ensembles.



Question 4 Donner un ensemble que l'on ne pourra pas représenter par notre structure. Le nombre d'ensembles non représentable est-il fini ? Dénombrable ? Indénombrable ?



Question 5 Soit `val P: int set set`, montrer qu'il existe un $N \in \mathbb{N}$ tel que pour tout `val x: int set`, `P x` ne regarde que les entrées inférieure à N dans `x`. Est-ce vrai avec `val P: (int -> int) set` ?



Question 6 Écrire une fonction `val f: (int set set) -> int set` qui à un `int set set` non vide associe un de ces éléments.

Chapitre VI. Logique

Problème VI.1: Compacité

Pour A un ensemble de formules de la logique propositionnelle, on dit qu'une valuation μ *satisfait* A si $\forall F \in A, \mu \models F$. On note cela $\mu \models A$.

On dit que A est *finement satisfiable* si pour tout $E \subseteq A$ fini, E est satisfiable.

On pose $(X_n)_n$ une suite de variables propositionnelles.



Question 0 Les ensembles suivants sont-ils satisfiables ?

- $A_1 = \{X_{2n} : n \in \mathbb{N}\} \cup \{\neg X_{2n+1} : n \in \mathbb{N}\}$
- $A_2 = \{X_i \vee \neg X_{i+1} : i \in \mathbb{N}\}$
- $A_3 = \{X_i \wedge \neg X_{i+1} : i \in \mathbb{N}\}$

On cherche à montrer le *théorème de compacité*: A est satisfiable si et seulement si A est finement satisfiable



Question 1 Montrer que si A est satisfiable alors il est finement satisfiable.



Question 2 Dans le cas où l'ensemble des variables propositionnelles de A est dénombrable, démontrer le théorème de compacité. On pourra chercher à construire une valuation par récurrence sur les variables propositionnelles.

On dit qu'un graphe non orienté $G = (S, E)$ avec $E \subseteq S^2$ et S potentiellement infini est N -coloriable s'il existe une fonction $c : S \rightarrow \llbracket 1; N \rrbracket$ tel que $\forall (x, y) \in E, c(x) \neq c(y)$



Question 3 Utiliser le théorème de compacité pour montrer que un graphe infini est N coloriable si et seulement si tout ses sous-graphes fini sont N coloriables.

Chapitre VII. Langages fonctionnels

Chapitre VIII. Mathématiques pour l'informatique

Problème VIII.1: Monoïdes libres, langages et actions

I Demi-groupes, monoïdes et groupes

Soit un *demi-groupe* $(\mathbb{E}, +)$, c'est-à-dire que


- \mathbb{E} est stable par $+$
- La loi $+$ est associative

On dira de plus que \mathbb{E} est un *monoïde* si il existe $e \in \mathbb{E}$ tel que

$$\forall x \in \mathbb{E}, xe = ex = x$$

On dira enfin que \mathbb{E} est un *groupe* si il existe $\cdot^{-1} : \mathbb{E} \rightarrow \mathbb{E}$ tel que

$$\forall x \in \mathbb{E}, xx^{-1} = x^{-1}x = e$$

 **Question 0** Donner un groupe, puis un monoïde qui n'est pas un groupe, et enfin un demi-groupe qui n'est pas un monoïde.


Si \mathbb{E} est un monoïde, soit $\sim \in (\mathbb{E}^2)^2$ telle que $(a, b) \sim (c, d) \iff a + d = b + c$.

 **Question 1** Que dire de \mathbb{E}^2 / \sim ?


Soit Σ un ensemble fini. On appelle Σ^* le plus petit monoïde contenant Σ et tel que

$$\forall u, v, w, x \in \Sigma^*, (u, v) = (w, x) \iff uv = wx$$

On note son neutre ε .

 **Question 2** Justifier que Σ^* est l'ensemble des mots finis sur Σ


On pose $\mathcal{A} := \{x \mapsto xw, w \in \Sigma^*\}$, que l'on munit de la loi de composition usuelle des fonctions.

 **Question 3** Justifier que Σ^* et \mathcal{A} sont isomorphes comme monoïdes.


II Associativité ?

Dans cette partie, $(S, +)$ est un demi-groupe.


Soit $n \in \mathbb{N}$ puis $a \in S^n$ un n -uplet.

 **Question 4** Donner le langage des expressions calculant $\sum a$. Est-il rationnel ?

Ind: Par exemple, pour $n = 3$, $\mathcal{L} = \{a_1 + (a_2 + a_3), (a_1 + a_2) + a_3\}$.

 **Question 5** Mettre en bijection \mathcal{L} et l'ensemble des arbres binaires à n noeuds. Dénombrer \mathcal{L} .


On considère maintenant posséder une machine capable d'exécuter $\omega \in \mathbb{N}^*$ opérations “+” simultanées.

 **Question 6** Donner un mauvais ordre de calcul de $\sum a$, puis un choix plus raisonnable.


III Retouches

Soient \mathcal{L} un langage rationnel et $M \in \Sigma^*$ un mot de longueur $n \in \mathbb{N}^*$. On appelle une *requête* un couple $1 \leq i \leq j \leq n$ et sa *taille* est $r := j - i$.

On *satisfait* une requête en renvoyant si $M[i : j] \in \mathcal{L}$. On note $q \in \mathbb{N}$ le nombre d'états d'un automate qui reconnaît \mathcal{L} .

 **Question 7** Donner un algorithme satisfaisant une requête.

Moyennant un précalcul,

 **Question 8** Donnez un algorithme efficace satisfaisant une requête en temps $\mathcal{O}(1)$ *Ind*: On pourra introduire un ensemble de fonctions similaire à \mathcal{A} agissant sur l'automate

Une *modification* est une opération de la forme $M[i] \leftarrow a$ avec $a \in \Sigma$.

 **Question 9** Modifier l'algorithme précédent pour permettre des modifications en temps $\mathcal{O}(q \log r)$.