
Feuille d'exercices n°16 - Exercices types Mines-Télécom

Exercice 1 : Grammaires

mines-télécom, 2023

On fixe l'alphabet $\Sigma = \{a, b, c, d\}$. On considère alors $L = \{a^n b^m c^p d^q \mid (n, m, p, q) \in \mathbb{N}^4, n+p = m+q\}$, $L_1 = \{u \in L \mid |u|_a \leq |u|_b\}$, $L_2 = \{u \in L \mid |u|_a \geq |u|_b\}$.

- Q. 1 Montrer $L_1 = \{a^n b^n b^m c^m c^p d^p \mid (n, m, p) \in \mathbb{N}^3\}$.
- Q. 2 Proposer une grammaire qui engendre L_1 .
- Q. 3 Proposer une grammaire qui engendre L_2 .
- Q. 4 Proposer une grammaire qui engendre L . Est-elle ambiguë ?

Exercice 2 : Dédution naturelle

mines-télécom, sujet 0

Cet exercice était accompagné d'une annexe rappelant les règles de la déduction naturelle : axiome, affaiblissement, absurde ($\Gamma, \bot \vdash A$ donne $\Gamma \vdash A$), élimination et introduction de l'implication, la conjonction, la disjonction, la négation.

- Q. 1 Prouver le séquent $A \wedge B \vdash B \wedge A$.
 - Q. 2 Prouver le séquent $A \wedge (B \vee C) \vdash (A \wedge B) \vee (A \wedge C)$.
- Soit $\Gamma \vdash C$ un séquent prouvable à l'aide d'un arbre de preuve Π .
- Q. 3 Montrer qu'il existe un arbre de preuve de $\Gamma \vdash C$ qui ne possède pas la succession de la règle d'élimination de la conjonction puis introduction de la conjonction.
 - Q. 4 Que peut-on dire pour les successions de règles de disjonction ?
 - Q. 5 Prouver le séquent $A \wedge B \vdash A \wedge B$.

Exercice 3 : Arbres et tri

mines-télécom, sujet 0

Soit A_n un arbre enraciné à n nœuds et de hauteur h .

- Q. 1 Donner le nombre minimal et maximal de nœuds en fonction de la hauteur h et le prouver.
- Soit T un tableau de taille n . On suppose qu'on ait un algorithme de tri qui puisse seulement comparer deux à deux les valeurs de ce tableau. On note a_n l'arbre de décisions associé à l'exécution de l'algorithme sur T . Les feuilles représentant le résultat de l'algorithme et les nœuds le choix de l'échange.
- Q. 2 Prouver que a_n a $n!$ feuilles.
 - Q. 3 Prouver que la hauteur de a_n est un $\Omega(n \log_2(n))$ c'est-à-dire qu'il existe C une constante telle que la hauteur de a_n est inférieure ou égale à $Cn \log_2(n)$.
 - Q. 4 Que peut-on en déduire sur la complexité pire cas d'un algorithme de tri par comparaison ?
 - Q. 5 Si on suppose que T ne contient que des entiers de $\llbracket 0, K \rrbracket$, donner un algorithme de tri de T en $\Theta(n + K)$.

Exercice 4 : Graphe biparti

mines-télécom, 2023

Soit $G = (V, E)$ un graphe biparti, c'est-à-dire qu'il existe V_1 et V_2 tels que E inclu dans $V_1 \times V_2$.

- Q. 1 Quel est le nombre minimal et maximal d'arêtes possibles dans un graphe biparti ?
 - Q. 2 Montrer qu'un graphe biparti l'est ssi il n'admet pas de cycle de longueur impaire.
 - Q. 3 Donner un algo permettant de déterminer si un graphe est biparti et donner sa complexité.
 - Q. 4 Donner une formule de la logique propositionnelle indiquant le fait qu'un graphe est biparti ?
-

Question subsidiaires

- Pouvez-vous mettre la formule proposée à la question 4 sous 2-sat ?
- Connaissez-vous un algorithme qui résout 2-SAT ?
- Quelle est la complexité de 2-SAT ?
- Quelle est la complexité de 3-SAT ?
- Connaissez-vous d'autres usages d'un parcours quelconque de graphe ?
- Combien de composantes connexes a un graphe biparti ?
- Dans quel algorithme au programme de MPI utilise-t-on des graphes bipartis ?

Exercice 5 : Concurrency

mines-télécom, sujet0

On s'intéresse à un bus touristique pouvant contenir C passagers. On suppose que l'on a $n > C$ passagers qui attendent leur tour, puis se remettent en attente à l'arrêt du bus dès qu'ils ont fini pour le revoir. Le bus ne démarre que lorsqu'il est plein.

Pour formaliser ce problème on utilise des fonctions fictives :

- board et unboard permettant au passager de monter et descendre ;
- le bus doit appeler les fonctions load lorsqu'il se remplit, run lorsqu'il démarre son tour et unload lorsqu'il se vide.

Attention les passagers ne peuvent pas descendre avant que le bus ait ouvert ses portes pour une fin de tour avec unload et ne peuvent pas monter avant que le bus ait ouvert ses portes pour un nouveau tour avec load.

- Q. 1 Écrire les fonctions en pseudo-code correspondant au bus et aux passagers **sans** prendre en compte les problèmes de synchronisation dans un premier temps.
- Q. 2 Proposer une solution en pseudo-code utilisant deux compteurs protégés par des mutex et quatre sémaphores.
- Q. 3 Votre solution peut-elle être utilisée dans le cas où l'on a plusieurs bus ? Justifier.
- Q. 4 Proposer une nouvelle solution pour le pseudo-code du bus dans le cas où il y a m bus numérotés respectant les règles suivantes.
 - un seul bus peut ouvrir ses portes aux passagers à la fois ;
 - un bus doit avoir fini de décharger avant qu'un autre vienne décharger ;
 - plusieurs bus peuvent faire un tour en même temps ;
 - les bus ne peuvent pas se doubler donc ils se chargent et déchargent toujours dans le même ordre.

La solution doit utiliser deux tableaux de sémaphores en plus des sémaphores utilisés dans la solution précédente permettant de gérer la coordination entre les bus.