Informatique

Présentation du sujet

Le sujet 2014 de l'option informatique traite de la résolution du jeu de « Sudoku » par écriture des formules logiques déduites des contraintes, puis programmation afin d'effectuer la résolution du jeu. À notre connaissance, l'ensemble des outils étudiés permet la résolution de toutes les grilles. Le problème, sans être de difficulté croissante, offre de nombreuses questions de tous niveaux. Il est nécessaire néanmoins de bien comprendre la structure de liste utilisée pour traduire les formes normales conjonctives dans le langage choisi. Cette année encore la longueur est volontairement raisonnable pour laisser aux candidats le temps de faire explicitement la programmation.

Analyse globale des résultats

Le sujet a été globalement correctement traité. Quelques candidats cependant n'ont pas du tout compris la structure de données. Mais le plus souvent ils n'ont pas non plus écrit correctement les différentes formes normales conjonctives. Contrairement à la demande explicite dans le sujet, beaucoup de candidats composant en `Caml` oublient de préciser la signature des fonctions ou mettent des signatures fantaisistes. Certains confondent la signature avec le caractère itératif ou récursif de la fonction. Cela peut paraître anecdotique, puisque le compilateur donne la signature, mais c'est souvent significatif d'incompréhension dans les types utilisés. D'ailleurs, certains candidats confondent les listes et les vecteurs, ou convertissent les listes en vecteurs pour chercher des éléments. De même nous avons vu très souvent des itérations triples ou quadruples programmées avec des fonctions récursives à arguments multiples, ce qui n'est certes pas interdit, mais rend la lecture très complexe et les tests de validité beaucoup plus difficiles. Inversement des fonctions simples, « naturellement » récursives, comme l'ajout dans une liste, sont écrites de façon itérative. Dans les fonctions itératives, les listes sont parfois utilisées sans référence, donc non modifiées par la fonction.

Les candidats doivent se souvenir que les codes informatiques sont le plus souvent des travaux collectifs. Ils ont donc vocation à être lus, compris, et corrigés éventuellement, par des équipes de programmeurs. Ils doivent donc être corrects, mais aussi clairs ou au moins commentés. De même, il n'est pas idéal de multiplier les fonctions auxiliaires au sein d'une même fonction élémentaire. Les meilleurs ont traité correctement le problème, avec une bonne rédaction.

Commentaires sur les réponses apportées et conseils aux candidats

La première partie ne posait pas de problème. Questions de cours sur l'utilisation des listes. Quand on demande de programmer les fonctions, la réponse, consistant à se limiter à dire qu'elles existent déjà, est insuffisante. La question **I.D**, correspondance d'indices, a parfois été traitée par un « matching » complet heureusement très minoritaire.

La partie **II** permet de décrire les règles du jeu et de traduire la configuration initiale. Les formules logiques sont à écrire en formes normales conjonctives afin de correspondre à la programmation où ces formes sont des listes de clauses et les clauses des listes de littéraux. De nombreux candidats se sont révélés incapables de faire cette distinction ou d'utiliser correctement les constructeurs des types `X(i, j, k)` et `NonX(i, j, k)`. La programmation présente les mêmes difficultés avec confusion des clauses et des littéraux ce qui conduit à des confusions entre l'ajout en tête et la

concaténation d'une part, et entre les listes de littéraux et les listes de clauses d'autre part. Les gestions de listes sont parfois douteuses : modification sans affectation ou sans référence, concaténation d'une liste énorme par une toute petite, ce qui nuit à l'efficacité, même si ce point n'est pas explicitement demandé. Dans la traduction des formulations logiques, beaucoup n'écrivent pas des formes normales conjonctives, ce qui n'est pas obligatoirement faux du point de vue logique, mais totalement inopérant dans le cadre de ce sujet. De même, considérer les couples $i < j$ au lieu de tous les couples $i \neq j$, devrait être naturel pour les candidats.

La partie **III** explique comment résoudre la grille initiale. Bien entendu pour une grille correcte, il n'y a qu'une solution, mais la table de vérité, comme l'ont constaté la plupart des candidats, n'est pas faisable, en temps et en espace raisonnables, dans notre univers, d'où la description et la programmation des règles de propagation unitaire et du littéral infructueux. Rien de spécialement nouveau dans la gestion des listes, mais là encore des difficultés pour différencier la clause du littéral, le littéral de la variable, et la formule vide, satisfiable, d'une formule contenant la clause vide, non satisfiable. Les calculs de complexité sont comme toujours moyennement traités. La proximité de la fin de l'épreuve conduit à des programmations parfois très peu lisibles ; on peut espérer que ce soit plus par manque de temps, mais il convient de mettre au moins quelques explications sur ce que l'on cherche à faire.

Conclusions

Bien que le temps de préparation soit réduit, on attend des candidats des idées claires sur les bases du programme et de bonnes capacités d'adaptation aux situations proposées. Une bonne pratique de la programmation devant machine est indispensable.

Le niveau global des candidats est satisfaisant. Certaines copies sont tout à fait excellentes, claires, précises, ce qui, sans machine, est une vraie performance. Le jury félicite les candidats qui se sont investis dans l'informatique.