Correction du DS 1

Option informatique, deuxième année

Julien Reichert

Exercice 1

Il est envisageable de se limiter à des arbres d'entiers, mais pour cette correction nous allons rester généraux.

```
type 'a ABR = V | N of 'a ABR * 'a * 'a ABR;;

let rec minimum_ABR = function
| V -> failwith "Arbre vide"
| N(Vide,a,_) -> a
| N(g,_,_) -> minimum_ABR g;;
```

La signature est ici `'a ABR -> 'a`.

Exercice 2

```
let rec insertion_ABR elt = function
| V -> Noeud(V,elt,V)
| N(g,x,d) -> if elt < x then N(insertion_ABR elt g,x,d) else N(g,x,insertion_ABR elt d);;
```

Exercice 3

Le détail étapes par étapes est préférablement fait au tableau.

L'écriture en Caml de l'arbre obtenu est la suivante (l'arbre peut différer si la fonction d'insertion ne choisit pas d'envoyer chaque nouvel élément dans une feuille) :

```
N (
  N (V, 5, N (V, 94, V)),
  167,
  N (
    N (N (V, 258, V), 578, V),
    600,
    N (
      V,
      654,
      N (
        N (
          V, 663, N (V, 706, V)),
        977,
        V))))
```

Cet arbre a été obtenu automatiquement par la fonction suivante :

```
let construit_ABR liste =
  let rec construit_aux accu = function
  | [] -> accu
  | a::q -> construit_aux (insertion_ABR a accu) q
  in construit_aux V liste;;
```

Le parcours en profondeur infixe (soit ici la lecture dans l'ordre naturel du code Caml) d'un ABR donne la version croissante de la liste de ses étiquettes. Cela permet d'obtenir un algorithme de tri dont la complexité est excellente si on peut garantir que la hauteur des ABR est logarithmique en sa taille, car sinon les appels à `insertion_ABR` deviennent trop coûteux.

Exercice 4

```
let pluspetitecart l =
  let rec ppe mini = function
  | [] -> failwith "Liste trop courte" (* Cas n'arrivant jamais *)
  | [a] -> mini
  | a::b::q -> ppe (min mini (b-a)) (b::q)
  in ppe (hd (tl l) - hd l) (tl l);;
```

En pratique, si la liste n'est pas triée, la fonction calcule la plus petite différence entre deux éléments consécutifs (le suivant moins le précédent à chaque fois). Bien entendu, pour des listes de taille inférieure ou égale à un, une erreur s'impose (on peut tolérer une réponse fixée à l'infini, soit `max_int`).

Exercice 5

```
type 'a arbre_hauteur = Vide | Noeud of 'a arbre_hauteur * ('a * int) * 'a arbre_hauteur;;

let hauteur = function
| Vide -> 0
| Noeud(_, (_, h), _) -> h;;

let rec inserer_hauteur elt = function
| Vide -> Noeud(Vide, (elt, 1), Vide)
| Noeud(g, (x, ht), d) when hauteur g < hauteur d ->
    let nouvarb = inserer_hauteur elt g in
      let nouvht = max ht (1 + hauteur nouvarb) in
      Noeud(nouvarb, (x, nouvht), d)
| Noeud(g, (x, ht), d) ->
    let nouvarb = inserer_hauteur elt d in
      let nouvht = max ht (1 + hauteur nouvarb) in
      Noeud(g, (x, nouvht), nouvarb);;
```

On n'obtient pas nécessairement un arbre de hauteur optimale car en cas d'égalité de hauteur des fils on privilégie l'insertion à droite, ce qui peut poser problème si par exemple le sous-arbre de droite est complet alors que le sous-arbre de gauche ne l'est pas, car dans ce cas la hauteur totale de l'arbre augmentera alors qu'on aurait pu passer de l'autre côté. Ceci étant, il y a un certain équilibre dans les arbres obtenus, et la hauteur sera logarithmique en la taille.

Une meilleure façon de procéder serait de stocker la taille en tant qu'information et de partir du côté de la plus petite taille, ce qui a d'ailleurs l'avantage de simplifier les mises à jour.

Exercice 6

Remarque : Les deux premiers énoncés correspondent à des énigmes où la règle est différente : les deux écriteaux énoncent simultanément la vérité ou simultanément un mensonge. Il n'était pas nécessaire de fournir des variantes différentes sur trois questions, la confusion pouvait déjà régner assez facilement.

Pour cet exercice et le suivant, la correction fait honneur aux formules booléennes.

Notons P_i, T_i et V_i les variables propositionnelles indiquant que la cellule numéro $i \geq 1$ en partant de la gauche contient respectivement une princesse, un tigre ou rien. Dans les deux premiers cas P_i et T_i sont mutuellement exclusifs pour $i = 1$ et $i = 2$, et dans le troisième cas nous avons $(P_1 \wedge \neg T_1 \wedge \neg V_1) \vee (\neg P_2 \wedge T_2 \wedge \neg V_2) \vee (\neg P_3 \wedge \neg T_3 \wedge V_3)$.

Premier énoncé

L'écriteau de gauche annonce $P_1 \vee P_2$ et l'écriteau de droite annonce T_1.

Pour simplifier avant de compliquer les choses, le fait qu'il n'y ait pas de cellule vide nous permet de remplacer T_1 par $\neg P_1$.

Maintenant, puisqu'un écriteau dit vrai si la cellule contient une princesse et faux si elle contient un tigre, nous pouvons affirmer d'après l'écriteau de gauche que $P_1 \Leftrightarrow (P_1 \vee P_2)$ et d'après l'écriteau de droite que $P_2 \Leftrightarrow \neg P_1$.

Examinons la première affirmation : on a toujours une des deux implications, de sorte que l'information pertinente obtenue est la réciproque, à savoir que $P_1 \Leftarrow (P_1 \vee P_2)$, donc que $\neg P_1 \Rightarrow (\neg P_1 \wedge \neg P_2)$, ce qui se résume à $\neg P_1 \Rightarrow \neg P_2$.

En combinant cette information avec la deuxième porte, nous avons que $P_2 \Rightarrow \neg P_2$ (à ne pas écrire $P_2 \Leftrightarrow \neg P_1 \Rightarrow \neg P_2$, car ceci serait vu comme une seule formule, ambiguë de surcroît), ce qui implique $\neg P_2$, soit en français : il y a un tigre dans la cellule de droite.

Finalement, de par l'équivalence, on obtient P_1 et il y a une princesse dans la cellule de gauche.

Deuxième énoncé

Réutilisons les remarques précédentes. Nous avons deux informations ici : par la cellule de gauche, $P_1 \Leftrightarrow (\neg P_1 \vee P_2)$, et par la cellule de droite, $P_2 \Leftrightarrow P_1$.

Grâce au premier énoncé, nous savons que $\neg P_1$ est impossible, car il implique $\neg P_1 \vee P_2$ qui est équivalent à P_1. Dans ce cas, P_1 est vrai, de même que P_2 qui lui est équivalent, et il y a donc deux princesses !

Troisième énoncé

Cette fois-ci, il faut utiliser des implications, combinées à la formule énoncée en début de correction.

L'énoncé précise qu'on a de plus $(P_1 \wedge \neg P_2 \wedge \neg P_3) \vee (\neg P_1 \wedge P_2 \wedge \neg P_3) \vee (\neg P_1 \wedge \neg P_2 \wedge P_3)$, et de même pour les deux autres lettres.

La première information est $P_1 \Rightarrow V_3 \wedge T_1 \Rightarrow \neg V_3$, la deuxième information est $P_2 \Rightarrow T_1 \wedge T_2 \Rightarrow \neg T_1$ (ceci étant, on savait déjà que $T_2 \Rightarrow \neg T_1$ car il n'y a qu'un tigre), et la troisième information est $P_3 \Rightarrow V_3 \wedge T_3 \Rightarrow \neg V_3$, donc on apprend que la princesse n'est pas à droite.

À ce stade, un tableau de vérité pourrait contribuer à trouver la solution, en profitant du fait que seules six combinaisons (dont deux qu'on vient d'exclure) sont possibles. Autant finir avec des formules. Supposons P_2, alors on a T_1, ce qui entraîne V_3 d'après les conditions d'unicité, ce qu'un calcul violent permet de déduire des quatre grosses formules (un langage logique le dirait rapidement !). Mais alors d'après la première information, on tombe sur une contradiction car T_1 impliquait $\neg V_3$. Ainsi P_2 est faux et il ne reste que P_1, ce qui entraîne V_3 et donc T_2 et tout est cohérent. En d'autres termes, il y a une princesse à gauche, un tigre au milieu et rien à droite.

Exercice 7

Nous allons utiliser les variables H_A et H_B indiquant respectivement si A et B sont humains (donc $\neg H_A$ signifie que A est un vampire), et les variables F_A et F_B indiquant respectivement si A et B sont fous. Pour aller plus loin, on peut utiliser des raccourcis R_A et R_B, indiquant respectivement si A et B ont raison, sachant que par exemple $R_A \Leftrightarrow ((H_A \wedge \neg F_A) \vee (\neg H_A \wedge F_A))$.

L'énoncé nous permet d'écrire $(H_A \wedge \neg H_B) \vee (\neg H_A \wedge H_B)$ pour tous les cas.

Premier énoncé

Nous écrivons avec les variables-raccourcis $R_A \Leftrightarrow (F_A \wedge F_B)$ et $R_B \Leftrightarrow \neg(F_A \wedge F_B)$.

À ce stade, cela commence à chauffer, et mieux vaut sortir une bonne vieille déduction logique : soit A soit B a raison [1], et soit A soit B est un vampire. Ainsi, A et B sont dans le même état mental, puisqu'étant d'espèces différentes si de plus l'un était fou et l'autre non, ils affirmeraient la même chose.

Dans ce cas, si les deux sont fous, c'est le vampire qui dit la vérité, donc A, alors que si aucun n'est fou, c'est l'humain qui dit la vérité, donc B.

Quoi qu'il en soit, le vampire est A, et aucune déduction supplémentaire ne peut être faite sans expertise supplémentaire d'un psychiatre.

Deuxième énoncé

Nouvel essai avec des formules : $R_A \Leftrightarrow H_A$ et $R_B \Leftrightarrow H_B$ et finalement $R_A \Leftrightarrow \neg F_B$.

Il est intéressant d'étudier $R_A \Leftrightarrow H_A$, car en redéveloppant R_A nous observons que cela implique $\neg F_A$. On peut tout aussi bien le concevoir avec des phrases : un fou ne peut pas se dire humain, car s'il est humain et fou il se prendra pour un vampire et s'il est vampire et fou il se prendra pour un humain et se déclarera également vampire.

Nous savons donc $\neg F_A$, ainsi que $\neg F_B$, donc que R_A et de ce fait H_A : A est humain donc le vampire est B.

Troisième énoncé

Cette fois-ci, nous avons les formules $R_A \Leftrightarrow (F_A \vee F_B)$, $R_B \Leftrightarrow (F_A \vee F_B)$ et $R_A \Leftrightarrow H_A$, et on sait que la dernière implique que $\neg F_A$.

Pour une raison similaire à ce que l'on a vu avec le premier énoncé, si les deux personnages disent la même chose, c'est qu'exactement l'un est fou (donc c'est B). Nous en déduisons que R_A et R_B sont vraies et celui qui est fou est le vampire, donc B.

Quatrième énoncé

Nous terminons avec les formules $R_A \Leftrightarrow \neg H_B$ et $R_B \Leftrightarrow F_A$, avec de plus la précision $(F_A \wedge \neg F_B) \vee (\neg F_A \wedge F_B)$.

C'est le moment pour une distinction de cas (sans doute combinable avec un raisonnement par l'absurde).

Supposons F_A. Alors nous avons $\neg F_B$ mais aussi R_B, d'où H_B, d'où $\neg R_A$ et finalement H_A. Il n'y a alors pas de vampire et c'est impossible.

Nous avons donc $\neg F_A$ et donc F_B. Dans ce cas nous avons $\neg R_B$, d'où aussi H_B, donc R_A et finalement $\neg H_A$. Alors A est un vampire sain d'esprit et B est un humain fou.

1. Notez le singulier en raison de l'exclusion mutuelle !