

DS 1

Option informatique, deuxième année

Julien REICHERT

Pour les exercices 1 à 3, on impose le type suivant :

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

Exercice 1 : Écrire en Caml une fonction `minimum_ABR` qui prend en entrée un arbre binaire de recherche et qui retourne la valeur minimale y figurant.

Exercice 2 : Écrire en Caml une fonction qui prend en entrée un arbre binaire et un prédicat et qui détermine si toutes les valeurs dans l'arbre vérifient le prédicat.

On rappelle qu'un prédicat est assimilé à une fonction retournant un booléen. Ici, l'argument d'un prédicat est l'information associée à un nœud.

La fonction demandée dans l'exercice 2 peut débloquer dans l'exercice 3, mais s'en servir est une mauvaise idée si on cherche à optimiser la complexité, mieux vaut se servir du cours...

Exercice 3 : Écrire en Caml une fonction qui prend en entrée un arbre binaire et qui détermine s'il s'agit d'un ABR.

Exercice 3bis : Déterminer la complexité dans le pire des cas de la fonction précédente. (Il faut que la réponse soit au maximum quadratique en la taille de l'arbre.)

Exercice 4 : Écrire en Caml une fonction qui prend en entrée un arbre et qui détermine s'il s'agit d'un tas-min.

Pour l'exercice 4, on impose le type suivant :

```
type 'a arbre = Noeud of 'a * 'a arbre list;;
```

Il n'y a cette fois-ci pas d'arbre vide possible, mais cela simplifie le reste, et une feuille sera donc par exemple un nœud pour lequel la liste associée est vide.

Exercice 5 : Mettre les expressions booléennes suivantes en forme normale conjonctive et éventuellement simplifier.

- $(a \Rightarrow (b \Rightarrow c)) \Rightarrow (c \vee \neg a)$
- $\neg(a \wedge b \wedge c) \wedge (a \wedge b \vee a \wedge c \vee b \wedge c)$
- $(a \Rightarrow b) \Rightarrow ((a \Rightarrow \neg b) \Rightarrow \neg a)$

Exercice 6 : Résoudre la grille de Loopy suivante.

Règles du jeu : une boucle et une seule doit être dessinée le long des lignes du quadrillage (les lignes jaunes ici). Elle ne doit en particulier pas se croiser, donc par un point de coordonnées entières (les points noirs ici) passent exactement deux traits ou aucun. Les nombres entourés de quatre traits jaunes indiquent combien de ces traits font partie de la boucle.

3			3		3			2			3			2
3					2		2		3	1				
3	1		2						1	2		2	1	3
	1		1	2		3						1		
	1					1	3		2				2	3
			0	1	2							2	3	
			3		2	1		2	2			2		2
	2	2			3			3			1	1		
2	1	2				3				2		2	2	3
		2	2		2	2	2	3			2			