



(Séquence 6.7

Spécifications des arbres binaires



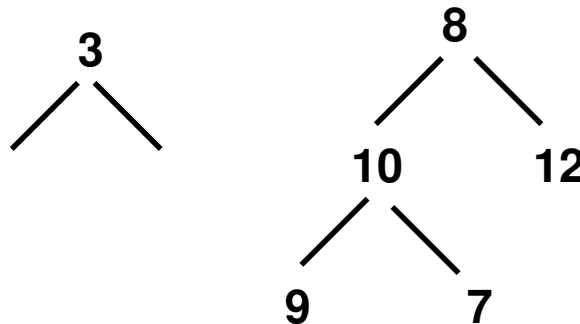
Spécification des constructeurs

```
;;; ab-vide:  -> ArbreBinaire[ $\alpha$ ]  
;;; (ab-vide) rend l'arbre binaire vide.  
  
;;; ab-noeud:  $\alpha$   
;;;          * ArbreBinaire[ $\alpha$ ]  
;;;          * ArbreBinaire[ $\alpha$ ]  
;;;          -> ArbreBinaire[ $\alpha$ ]  
;;; (ab-noeud e B1 B2) rend l'arbre binaire formé  
;;; d'une racine d'étiquette e, d'un sous-arbre  
;;; gauche B1 et d'un sous-arbre droit B2.
```



Exemples : arbres binaires de nombres

```
(ab-noeud 3 (ab-vide) (ab-vide))  
  
(let* ((b0 (ab-vide))  
      (b9 (ab-noeud 9 b0 b0))  
      (b7 (ab-noeud 7 b0 b0))  
      (b10 (ab-noeud 10 b9 b7))  
      (b12 (ab-noeud 12 b0 b0)) )  
  (ab-noeud 8 b10 b12) )
```



Un arbre est un objet opaque qui ne se manipule que via la barrière d'abstraction.



Spécification des accesseurs

```
;;; ab-etiquette: ArbreBinaire[ $\alpha$ ] ->  $\alpha$ 
;;; (ab-etiquette B) rend l'étiquette de la racine
;;; de l'arbre B
;;; ERREUR lorsque B ne satisfait pas ab-noeud?

;;; ab-gauche: ArbreBinaire[ $\alpha$ ] -> ArbreBinaire[ $\alpha$ ]
;;; (ab-gauche B) rend le sous-arbre gauche de B
;;; ERREUR lorsque B ne satisfait pas ab-noeud?

;;; ab-droit: ArbreBinaire[ $\alpha$ ] -> ArbreBinaire[ $\alpha$ ]
;;; (ab-droit B) rend le sous-arbre droit de B
;;; ERREUR lorsque B ne satisfait pas ab-noeud?
```



Propriétés algébriques

- ▶ Pour tout couple d'arbres binaires G et D , et toute valeur v

```
(ab-etiquette (ab-noeud  $v$   $G$   $D$ )) →  $v$   
(ab-gauche (ab-noeud  $v$   $G$   $D$ )) →  $G$   
(ab-droit (ab-noeud  $v$   $G$   $D$ )) →  $D$ 
```

- ▶ Pour tout arbre binaire non vide B

```
(ab-noeud (ab-etiquette  $B$ )  
          (ab-gauche  $B$ )  
          (ab-droit  $B$ )) →  $B$ 
```



Spécification du reconnaisseur

```
;;; ab-noeud?: ArbreBinaire[ $\alpha$ ] -> bool  
;;; (ab-noeud? B) rend vrai ssi B n'est pas  
;;; l'arbre vide.
```

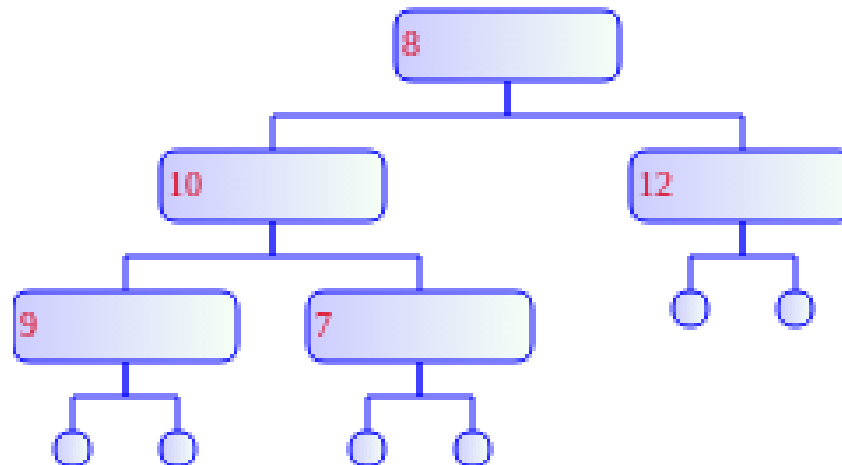
Propriété : Pour tout arbre binaire B,
 $(\text{ab-vide? } B) = (\text{not } (\text{ab-noeud? } B))$



Affichage d'arbre binaire

La barrière d'abstraction contient aussi une fonction d'affichage, qui permet de « visualiser » les objets arbres binaires par des S-expressions. En MrScheme :

```
(ab-affiche (let* ((b0 (ab-vide))  
                  (b9 (ab-noeud 9 b0 b0))  
                  (b7 (ab-noeud 7 b0 b0))  
                  (b10 (ab-noeud 10 b9 b7))  
                  (b12 (ab-noeud 12 b0 b0)) )  
            (ab-noeud 8 b10 b12) ))
```





Fin séquence)

