Option Informatique
Arbres couvrants minimaux

Corrigé

23 décembre 2005

1 Utilisation d'un tableau

```
let est_terminal T =
  let r = T.racine and tab = T.sommet in
  let A = tab.(r) in
  ((A.droit = -1) && (A.gauche = -1)) ; ;

let racine T =
  if T.racine = (-1) then failwith "arbre vide"
  else ((T.sommet).(T.racine).info) ; ;

let fils_gauche T =
  racine = ((T.sommet).(T.racine)).gauche ; sommet = T.sommet ; ;

let fils_droit T =
  racine = ((T.sommet).(T.racine)).droit ; sommet = T.sommet ; ;

let cons G r D =
  let tailleg = (vect_length G.sommet) and tailled = (vect_length D.sommet) in
  let A = make_vect (tailleg+tailled+1)
    droit = (tailleg+D.racine) ; info = r ; gauche = G.racine in begin
    for i = 0 to (tailleg - 1) do
      A.(i+1) <- (G.sommet).(i) ;
    done ;
    for i = 0 to (tailled - 1) do
      A.(i+1+tailleg)<- (D.sommet.i) ;
    done ;
    racine = 0 ; sommet = A ;
  end ; ;
```

2 Utilisation d'une liste

```
let est_terminal T =
  match T with
      []-> false ;
```

```
    | t : : q -> ((not t.droit) && (not t.gauche)) ; ;

let racine T =
  match T with
      [] -> failwith "arbre vide" ;
    |t : :q -> t.info ; ;

let fils_gauche T =
  let rec next_arbre T =
    match T with
        [] -> ([],[]) ;
      | t : :q -> if t.gauche && t.droit then
          let gauche,reste = (next_arbre q) in
          let droite,reste2 = (next_arbre reste) in
          t : :(gauche@droite),reste2 ;
        else if t.gauche || t.droit then
          let fils,reste = (next_arbre q) in
          t : :fils,reste ;
        else [t],q ;
  in
  match T with
      [] -> failwith "pas de fils gauche"
    | t : :q -> if not t.gauche then failwith "pas de fils gauche !" else
        let g,reste = (next_arbre q) in
        g ; ;

let cons G r D =
  match G,D with
      [],[] -> [droit = false ; info = r ; gauche = false] ;
    | [],_ -> droit = true ; info = r ; gauche = false : :D ;
    | _,[] -> droit = false ; info = r ; gauche = true : :G ;
    | _,_ -> droit = true ; info = r ; gauche = true : :(G@D) ; ;
```

. . . et on en tire facilement une écriture de `fils_droit`