▶ La manière naturelle de procéder consiste à poser une dalle en bas et à gauche du rectangle, puis à essayer d'en placer une à sa droite, puis une autre à la droite de celle-ci et ainsi de suite jusqu'à la fin de la rangée. On recommence alors avec la rangée immédiatement supérieure. Si aucune dalle ne convient, on recule d'un pas, et on remplace la dernière dalle placée par la prochaine dalle convenant à cette place, dans le catalogue ; si l'on a épuisé les possibilités, on recule encore...

```
let pavage_rectangle p n catalogue =
 let pavage = make_matrix n p (hd catalogue) in
 let convient t i j = (* vérifie que la dalle t convient *)
  (j=0 or t.g = pavage.(i).(j-1).d) &&
  (i=0 or t.h = pavage.(i-1).(j).b) in
 let rec pave i j dispo pile = (* tente de poursuivre *)
  match dispo with
  | [] -> recule pile (i,j)
  | t::q -> if convient t i j then avance i j t q pile
                              else pave   i j   q pile
 and avance i j t q pile = (* poursuit le pavage *)
   begin
    pavage.(i).(j) <- t;
    if j<p-1 then pave i (j+1) catalogue (q::pile) else
    if i<n-1 then pave (i+1) 0 catalogue (q::pile) else ()
   end
 and recule pile = function   (* on ne peut continuer *)
  | (0,_) -> failwith "pavage impossible"
  | (i,0) -> pave (i-1) (p-1) (hd pile) (tl pile)
  | (i,j) -> pave i (j-1) (hd pile) (tl pile)
in pave 0 0 catalogue []; pavage;;
```

▶ Il est clair que l'on effectue un parcours en profondeur d'un arbre...