

```

type etat = {final: bool; transitions: (char*int) list};;

type AFND = AFND of etat vect;;

let reconnait (AFND v) w =
  let rec reunit = function (* reunit une liste de listes *)
    | [] -> []
    | t::r -> union t (reunit r)
  and cherche_tous a = function
    (* recherche toutes les associations *)
    | [] -> []
    | (x,y)::r -> if x=a then y::(cherche_tous a r)
    else cherche_tous a r
  and cherche_bon = function (* recherche un état final *)
    | [] -> false
    | t::r -> v.(t).final || cherche_bon r

```

```
in
let delta partie c =
  let f = function
    x -> cherche_tous c v.(x).transitions in
  reunit (map f partie)
in let rec delta_barre partie = function
  | [] -> partie
  | t::r -> delta_barre (delta partie t) r
in
let dernier = delta_barre [0] w
in cherche_bon dernier;;
```