```
type expreg =
  | Vide
  | Epsilon
  | Caractere of char
  | Union     of expreg * expreg
  | Produit   of expreg * expreg
  | Etoile    of expreg

let rec contient_epsilon = function
  | Vide -> false
  | Epsilon -> true
  | Caractere _ -> false
  | Union (r1,r2) -> contient_epsilon r1 || contient_epsilon r2
  | Produit (r1,r2) -> contient_epsilon r1 && contient_epsilon r2
  | Etoile _ -> true

let rec residu r c = match r with
  | Vide | Epsilon ->
      Vide
  | Caractere d ->
      if c = d then Epsilon else Vide
  | Union (r1,r2) ->
      Union (residu r1 c, residu r2 c)
  | Produit (r1,r2) ->
      let r' = Produit (residu r1 c, r2) in
      if contient_epsilon r1 then Union (r', residu r2 c) else r'
  | Etoile r ->
      Produit (residu r c, Etoile r)

let reconnait r l =
  let res = List.fold_left residu r l in
  contient_epsilon res
```