1. On suppose que l'on dispose des fonctions `contient_final(Q)`
   `stocke_new_delta(Q,x,Q')`
   Elles n'interfèrent pas avec l'algorithme. `U` dénote l'union ensembliste ; `in` l'appartenance ensembliste.

```
type
  etat : ...
  gd_etat : ensemble de etat
var
  avoir : file de gd_etat
  a     : lettre
  q     : etat
  cour, newS : gd_etat ;
  etats, final : ensemble de gd_etat
begin
  enfiler(avoir, initial()) ;
  etats := { initial() } ;
  final := {} ;

  while !vide(avoir) {
    cour := defiler(avoir) ;
    if ( contient_final(cour) )
       final := final U { cour } ;
    for a in X {
      newS := {} ;
      for q in cour
        newS := newS U { delta(q,a) } ;
      stocke_new_delta(cour,a,newS) ;
      if ( ! newS in etats ) {
        etats := etats U { newS } ;
        enfiler(avoir, newS) ;
      }
    }
  }
end
```

2. La première traduction directe n'est pas tout-à-fait régulière (à cause des règles de la forme $A \to \varepsilon B$).

   ① → a① | a② | b③ | c⑤ | ⑤
   ② → a③ | b② | c①
   ③ → c⑤ | ④ | ⑥ | ε
   ④ → a③ | c⑥ | ②
   ⑤ → a⑤ | b④ | b⑥ | c⑥
   ⑥ → ε

   Pour supprimer les règles non régulières, on remplace B par tous les α tels que $B \to \alpha$. On commence par ④ → ② :

   ① → a① | a② | b③ | c⑤ | ⑤
   ② → a③ | b② | c①
   ③ → c⑤ | ④ | ⑥ | ε
   ④ → a③ | c⑥ | a③ | b② | c①
   ⑤ → a⑤ | b④ | b⑥ | c⑥
   ⑥ → ε

   ... on aboutit finalement à :

   ① → a① | a② | b③ | c⑤
   | a⑤ | b④ | b⑥ | c⑥
   ② → a③ | b② | c①
   ③ → c⑤ | ε | a③ | c⑥ | b② | c①
   ④ → a③ | c⑥ | b② | c①
   ⑤ → a⑤ | b④ | b⑥ | c⑥
   ⑥ → ε

3. Calcul de $\varepsilon+$:

	a	b	c	ε	$\varepsilon+$
→ 1	1,2	3	5	5	5
2	3	2	1		
← 3			5	4,6	2,4,6
4	3		6	2	2
5	5	4,6	6		
← 6					

   Élimination des ε-transitions :

	a	b	c
→ 1	1,2,5	3,4,6	5,6
2	3	2	1
← 3	3	2	1,5,6
4	3	2	1,6
5	5	4,6	6
← 6			

4.

	a	b	c
→ 1	1		2
2	3	5	
3		4	
← 4			
← 5	5		

$a^*c(ab|ba^*)$