

Introduction aux Méthodes Formelles

(Partie 3_2)

Cours de Master: M1 GLSD
2016-2017

Test 3

- Que signifie l'indéterminisme d'un automate? C'est quoi l'inconvénient de cette qualité? À quoi ça sert les AFN?
- Formellement c'est quoi un automate?
- Que signifie un état accessible, co-accessible, et utile? Donnez des exemples?
- À quoi sert le calcul de l'union? Formellement l'union de A_1 et A_2 ?
- Que signifie le mot vide? À quoi ça sert un tel mot?

automates finis (12):

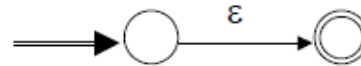
Expressions rationnelle

- Des expressions qui décrivent les **langages reconnaissables**
- Des expressions construites à base de :
 - **briques** : ε , les éléments de l'alphabet
 - **opérations** sur ces briques: $e+e'$ (**choix $e|e'$**), **concaténation** ab ($a.b$), **les répétitions** e^* et e^+
- Pour ces briques, on associe des **automates** et pour ces opérations, on associe des **constructions d'automates**
- Les **automates** obtenus ont **un seul état final**

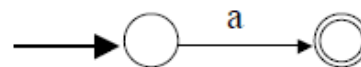
automates finis (5): rappel

Construction d'automates à partir d'une expression

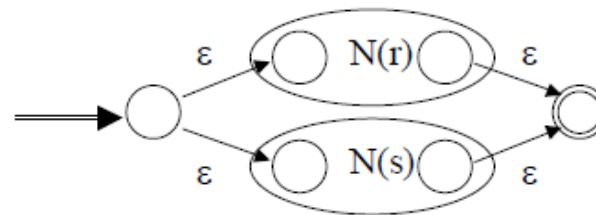
• pour ε : $N(\varepsilon)$



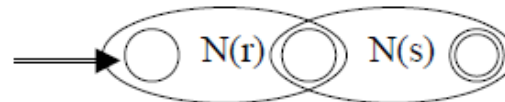
• pour a : $N(a)$



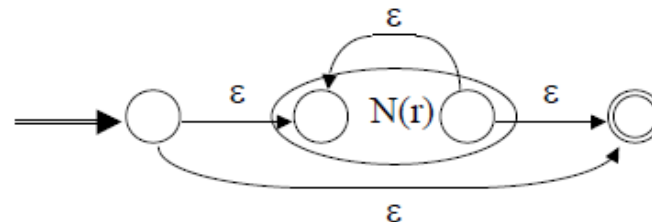
• pour $r + s$: $N(r + s)$



• pour rs : $N(rs)$



• pour r^* : $N(r^*)$

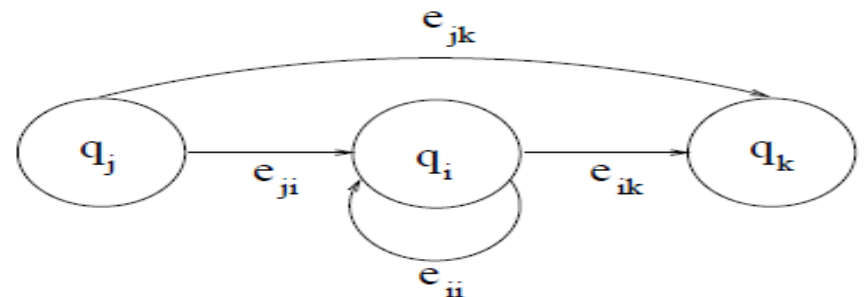


automates finis (13): rappel

Construction d'une expression à partir d'automates

- De proche en proche on **supprime des états intermédiaires**: si nous avons $s_1 \xrightarrow{e} s_2 \xrightarrow{e'} s_3$ on supprime s_2
- et le reste des **transitions seront étiquetés par des concaténations**: $s_1 \xrightarrow{ee'} s_3$
- jusqu'à la phase où nous allons **avoir uniquement deux états (initial et final)**.
- L'étiquette de la dernière transition est donc **l'expression reconnaissable**

Exemple :



Automates d'états finis: Spécification & Vérification

- 1) Éléments pour la Spécification;
- 2) Communication
- 3) Produit synchrone
- 4) Limites

Automates finis

spécification et vérification(1)

- **Alphabet**= des événements qui peuvent se produire, **des actions** que le système peut exécuter (une entrée, une sortie, un appel de méthode, etc),
- **Les états**= les états par lesquels le système transite durant son exécution
- **Calculs réussis**= représenter les **séquences d'évènements valides** d'un système;

Une abstraction de l'état du système:

- **Impossible de représenter tous** les états du système avec un automate
- Idée: avoir un **point de vue partiel**

Automates finis

spécification et vérification(2)

Exemple pour les circuits digitaux:

- L'entrée, l'état et le résultat sont des vecteurs de bits de taille fixe
- Les action sont liées aux transitions, (Mealy)
- Les actions sont liées aux états, (Moore)

Automate Partiel: la réceptivité

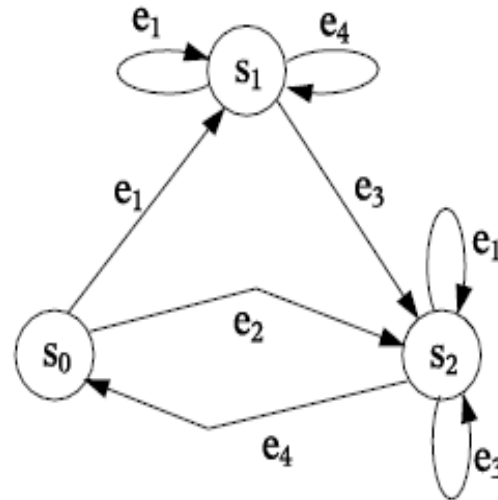
- Un état s **n'est pas réceptif** à un événement e ssi, il n'y a **aucun arc** sortant de s qui est **étiqueté e** .

$E = \{e_1, \dots, e_4\}$

s_0 n'est pas réceptif pour $\{e_3, e_4\}$

s_1 n'est pas réceptif pour $\{e_2\}$

s_2 n'est pas réceptif pour $\{e_2\}$



Si le système dans un état s se trouve **face à un événement e non réceptif** par s ,



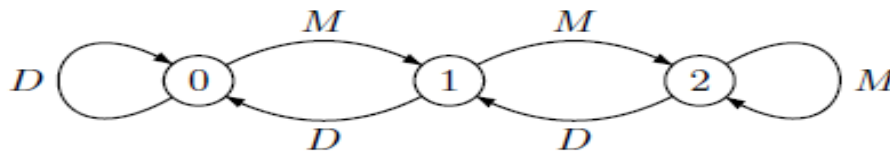
ceci exprime une **erreur ou une anomalie**

Exemple 1:

On cherche à spécifier un **assenseur**:

L'assenseur serve deux étages. Proposer une spécification avec les automate finis

?



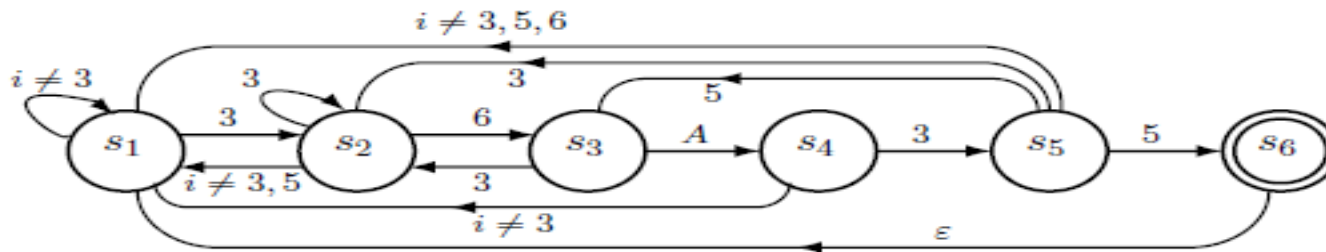
Exemple (2)

Digicode:

Un clavier alpha-numérique permet d'ouvrir une porte si le bon code est introduit:

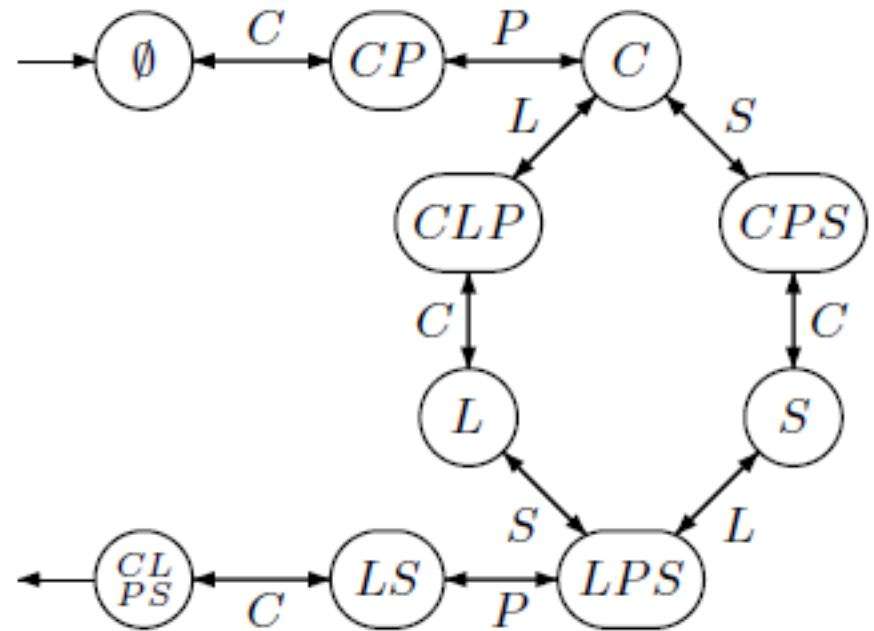
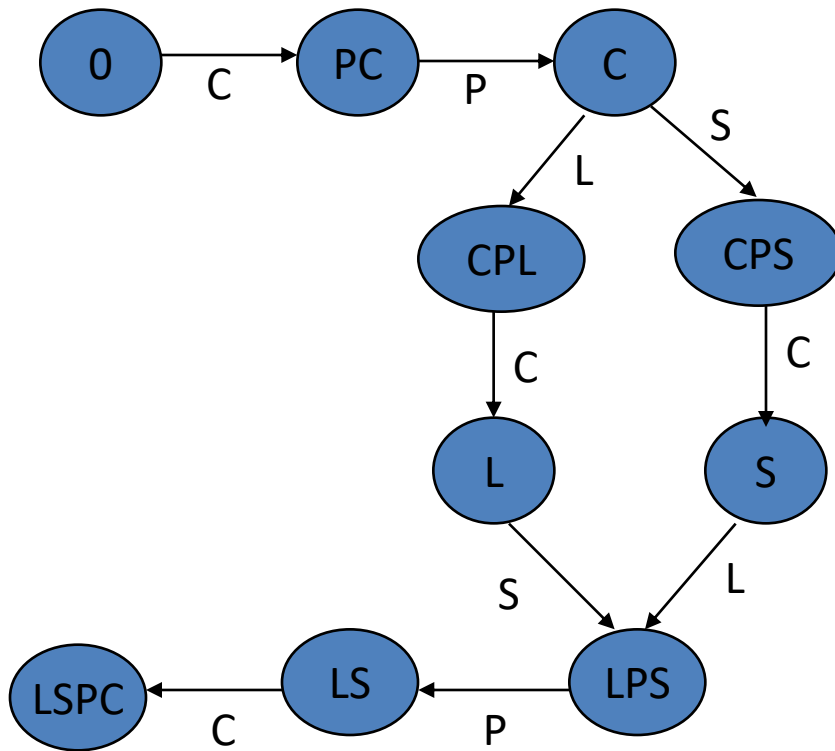
le code est : 36A35

?



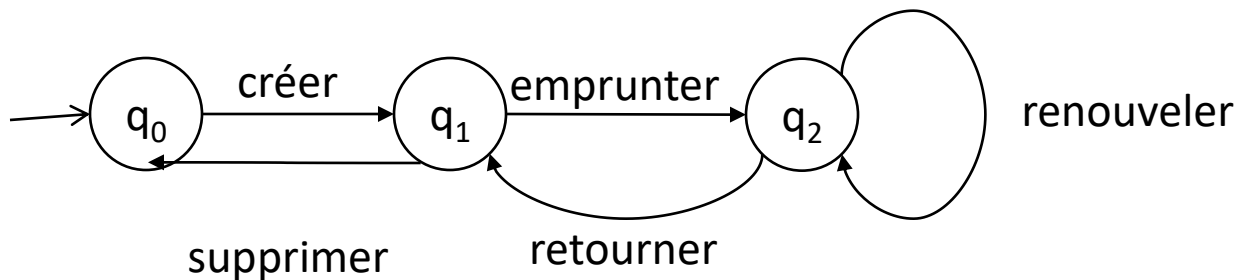
Exemple (3)

Problème du passeur, loup, chèvre, salade



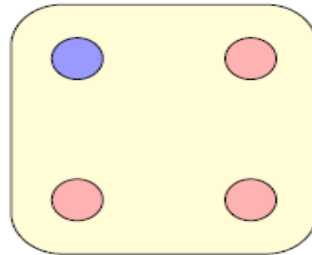
Exemple (4)

- Utilisez un automate pour spécifier le **comportement d'une bibliothèque** :
seulement **les prêts** de livres aux membres,
 - Créer le prêt
 - Emprunter le livre
 - Renouveler le prêt
 - Retourner le livre
 - Supprimer le prêt



Exemple (5)

- Un serveur aveugle joue au jeu suivant avec un client.
- Il a devant lui un plateau avec 4 verres.



A chaque tour le serveur fait retourner au client 1 ou 2 verres.
S'il s'agit de 2 verres, il précise s'ils sont adjacents ou opposés.
Si tous les verres sont dans le même sens, le serveur a gagné.

Exemple (5)

Les actions :

- 1) Retourner un verre
- 2) Retourner deux verres opposés
- 3) Retourner deux verres adjacents

Les états:

?

- On considère que l'état est la disposition des quatre verres,

Exp : **1, 0, 0, 0** représente 1 verre à l'envers, 3 verres à l'endroit, ou l'inverse,

Dans ce raisonnement, combien il y'a d'états?

Example (5)

Exemple (7)

Deux processus P_0 , P_1 . Variables $req[0]$, $req[1]$ et $turn$ partagées.

Code pour P_i :

```
req[i] = true
```

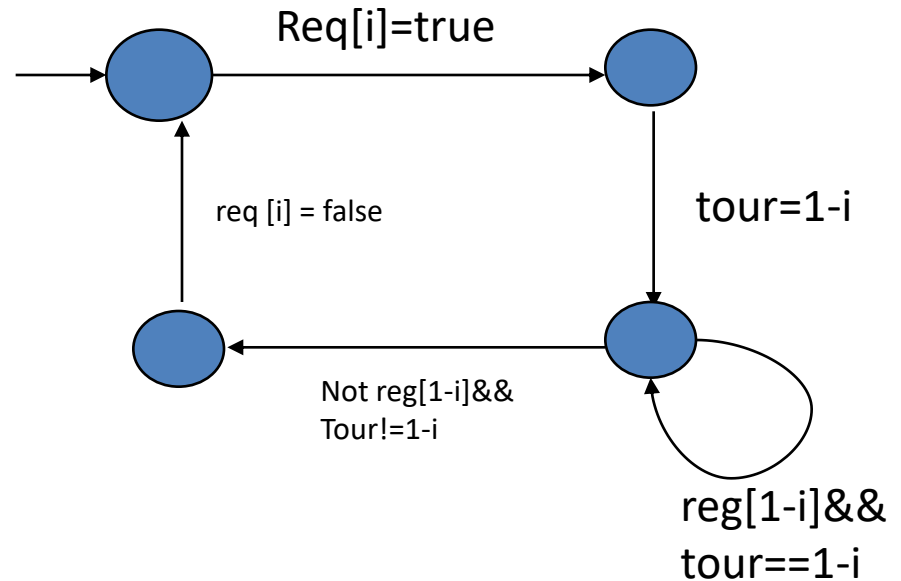
```
tour = 1-i
```

```
while (req[1-i] && tour == 1-i)
```

```
; // attente
```

```
section_critique ()
```

```
req[i] = false
```



Communication entre automates d'états finis (1)

Communications

→ **Synchrones** :

comme le téléphone,

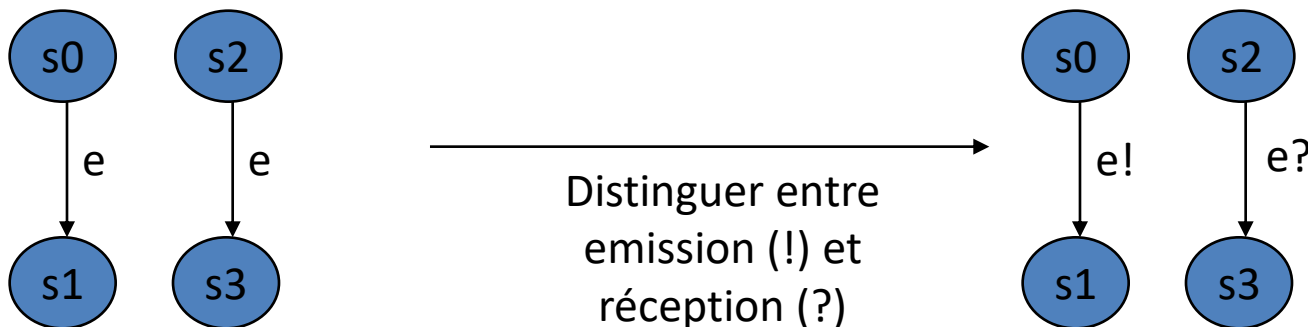
les communicants ont la même horloge,
émission/réception en même temps

→ **Asynchrone**:

comme un sms, une lettre, ...

les communicants sont indépendants, ils doivent utiliser
un tampon

Synchrone : utiliser un seul événement **e** pour l'émission et la réception
dans les deux automates.



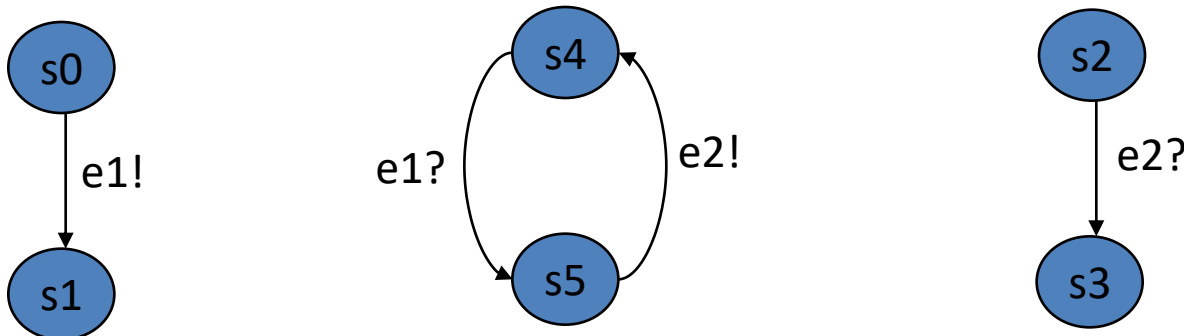
Communication entre automates d'états finis (2)

Asynchrone :

- utiliser deux événements e1 et e2.

e1 (émission) dans le premier automate,
e2 (réception) dans le deuxième automate.

- utiliser un troisième automate comme tampon



Produit Synchrone d'automates (1)

- Les sous-systèmes modélisés **ne sont pas** indépendants complètement.
- Ces systèmes communiquent et **certaines actions sont synchronisées**
- Une action A dans le sous-système 1 **doit s'exécuter avec** une certaine action B dans le sous-système 2.
- Quand on fait la composition des deux sous-systèmes, on doit faire attention à ces actions, et donc on doit les synchroniser
On parle de **produit synchronisé**

Produit Synchrone d'automates (5)

Dans le produit synchrone $A1 \otimes A2$, il y'a trois types d'événements :

- 1) Des événements qui sont de l'automate A1 (événements **inobservables par A2**)
- 2) Des événements qui sont dans l'automate A2 (événements **inobservables par A1**)
- 3) Des événements qui sont des couples d'événements dans A1 et A2 (**événements de synchronisation**)

Produit Synchrone d'automates (2)

Formellement :

Soit $A_1=(S_1, E_1, T_1, s_{01}, F_1)$ un automate

Et $A_2=(S_2, E_2, T_2, s_{02}, F_2)$ un automate.

On définit un ensemble d'alphabet E et une fonction Syn de synchronisation comme suit:

$$\text{Syn: } E_1 \cup \{ _ \} \times E_2 \cup \{ _ \} \rightarrow E$$

Le symbole $_$ signifie que rien n'est exécuté dans l'automate

Le produit synchronisé de A_1 et A_2 par rapport à la fonction Syn noté par $A_1 \otimes_{\text{Syn}} A_2$ est défini comme suit:

$$A_1 \otimes_{\text{Sync}} A_2 = (S_1 \times S_2, E, T, (s_{01}, s_{02}), F_1 \times F_2)$$

Produit Synchrone d'automates (3)

La transition $(s_{11}, s_{21}) \xrightarrow{\text{Syn}(a,b)} (s_{12}, s_{22})$ existe dans T ssi:

1) $s_{11} \xrightarrow{a} s_{12}$ et $s_{21} \xrightarrow{b} s_{22}$

2) ou : $s_{11} = s_{12}$ et $s_{21} \xrightarrow{b} s_{22}$ et $a = _$

3) Ou bien : $s_{11} \xrightarrow{a} s_{12}$ et $s_{21} = s_{22}$ et $b = _$

Produit Synchrone d'automates (4)

- La fonction Syn peut être vue comme un sous ensemble:

$$\text{Syn} \subseteq E_1 \cup E_2 \cup (E_1 \times E_2)$$

- Selon cette idée, l'ensemble T des **transitions du produit synchrone** est l'ensemble de transitions étiquetées par les événements de Syn

Produit Synchrone d'automates (6)

Exemple:

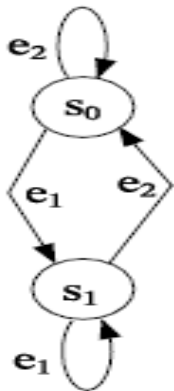
- 1) Proposez une modélisation d'un compteur de minutes
- 2) Proposez une modélisation d'un compteur d'heures
- 3) Calculer le produit cartésien des deux automates
- 4) Que pensez vous du produit synchrone des deux modèles précédents

Limite des automates d'états finis(1)

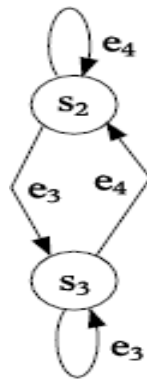
cas de systèmes complexes

- Systèmes complexes $\rightarrow$ **conception modulaire** $\rightarrow$ plusieurs composantes $\rightarrow$ **plusieurs automates** $\rightarrow$ composer ces automates à la fin

$E1=\{e1,e2\}$
 $S=\{s0,s1\}$

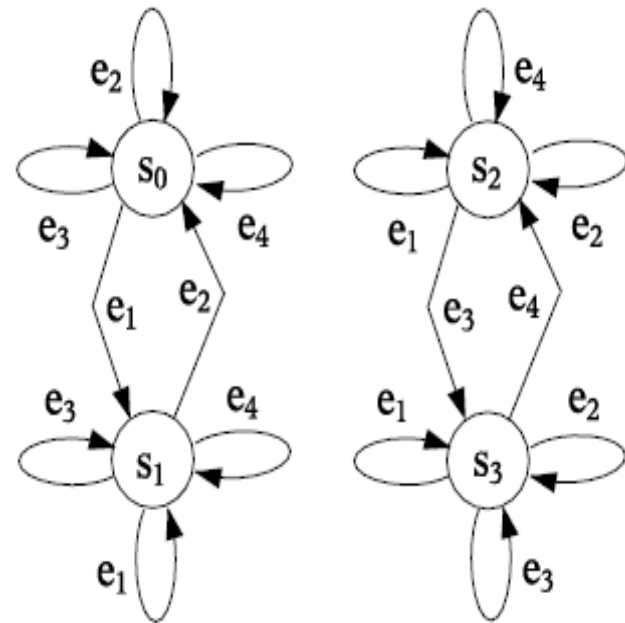


$E2=\{e2,e3\}$
 $S=\{s2,s3\}$



$E=\{e1,e2,e3,e4\}$

Plonger les
automates
dans E

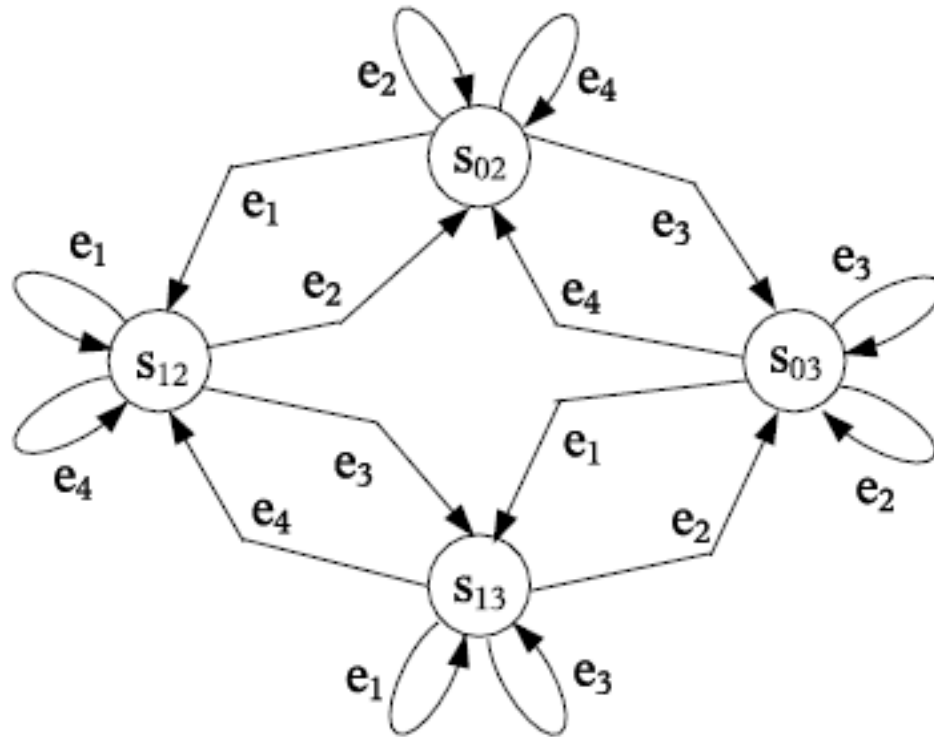


Problème1? Ambiguïté de la réceptivité dans les automates après plongement

Limites des automates d'états finis(2)

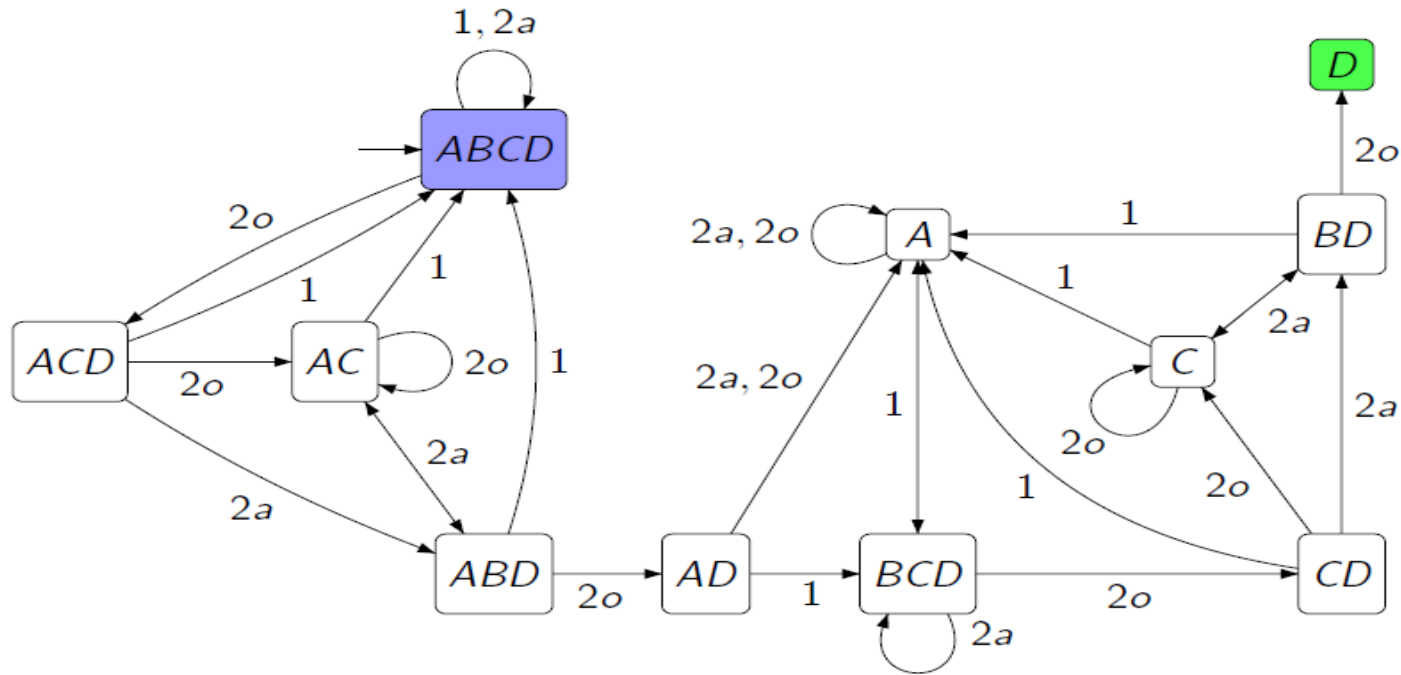
$S = S_1 * S_2 = \{e_{13}, e_{14}, e_{23}, e_{24}, e_{31}, e_{32}, e_{41}, e_{42}\}$

→
Définir le
produit
cartésien des
deux
ensembles
d'états



Problème2? Nombre important d'états → explosion combinatoire
→ étude complexe

Un autre exemple: détermination de l'automate de l'exemple 7 (4 verres)



Problème : Explosion exponentielle du nombre d'états

Limites des automates d'états finis(3)

- Problème pour représenter le parallélisme