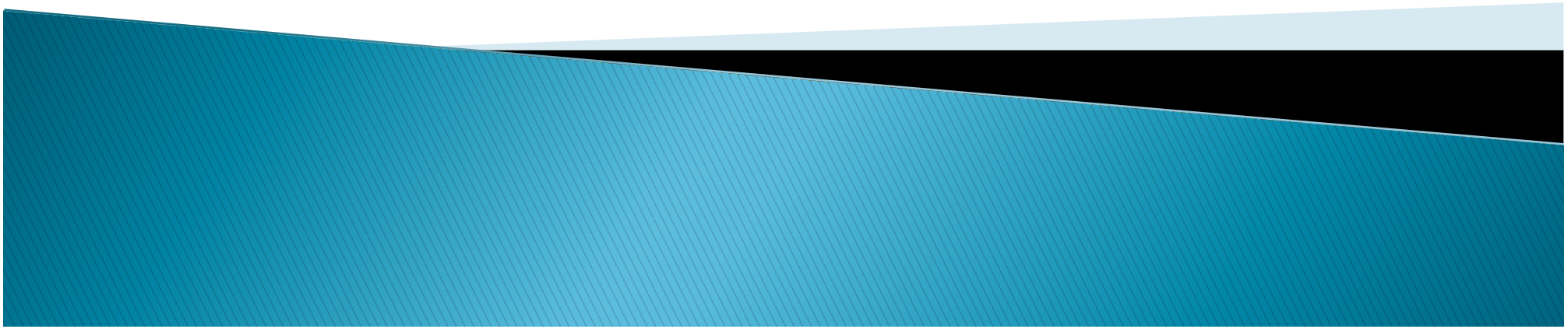
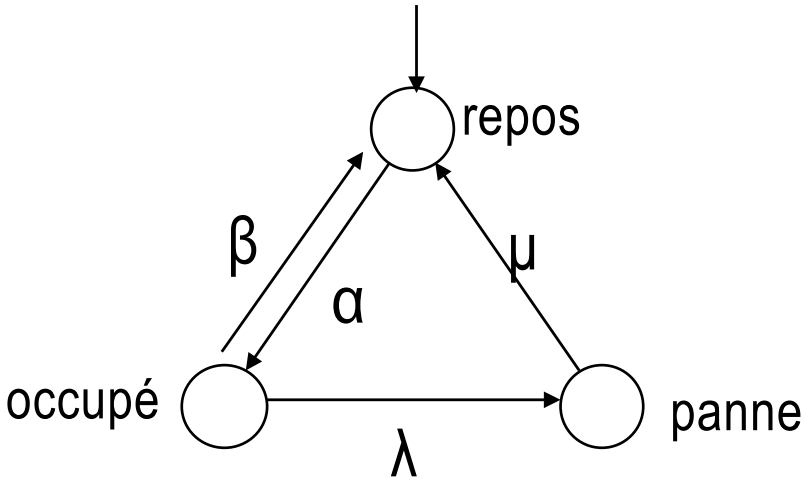
Chapitre 2
Commande des SED
La théorie de Contrôle par Supervision



Représentation des SED

Si on veut représenter une machine à 3 états on a besoin :



α : commencer une opération
β : fin de l'opération
λ : occurrence d'une panne
μ : réparer la machine

L={ε,α, α β, α λ, α λ μ, α λ μ α,…..}
L=(α β+ α λ μ)*(ε+ α + α λ)

Générateur d'événement :

G= {Q,∑, ,q_0,Q_m}

pour l'exemple on a :

∑ = {α, β, λ, μ} = $\sum_c$ $\sum_u$

$\sum_c$ = {α, μ} $\sum_u$ = {β,λ}

Q = {repos, occupé, panne}

q0 = repos Qm = repos

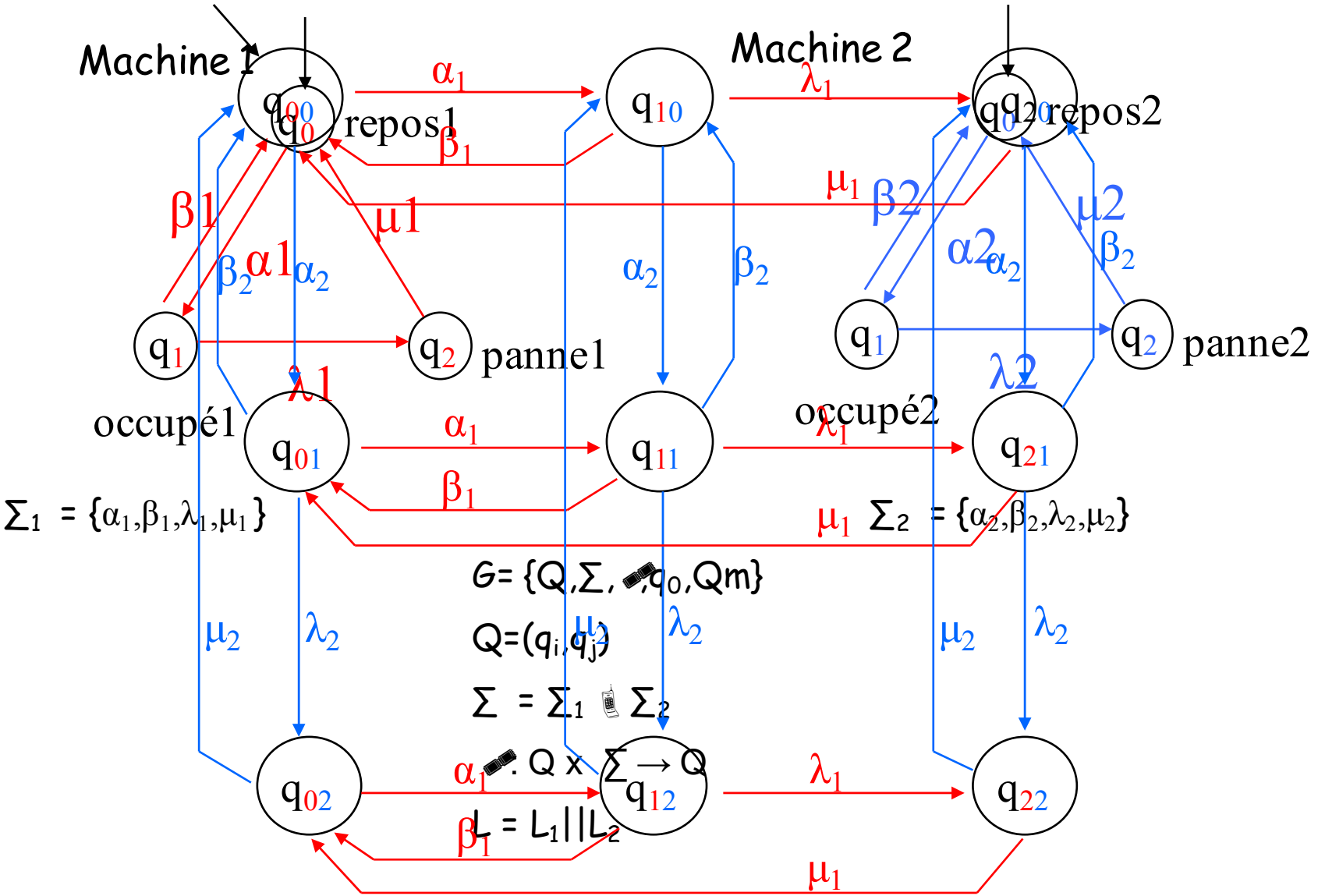
Fonctions de transition :

occupé= (repos, α)

panne = (occupé, λ)

repos = (occupé, β)

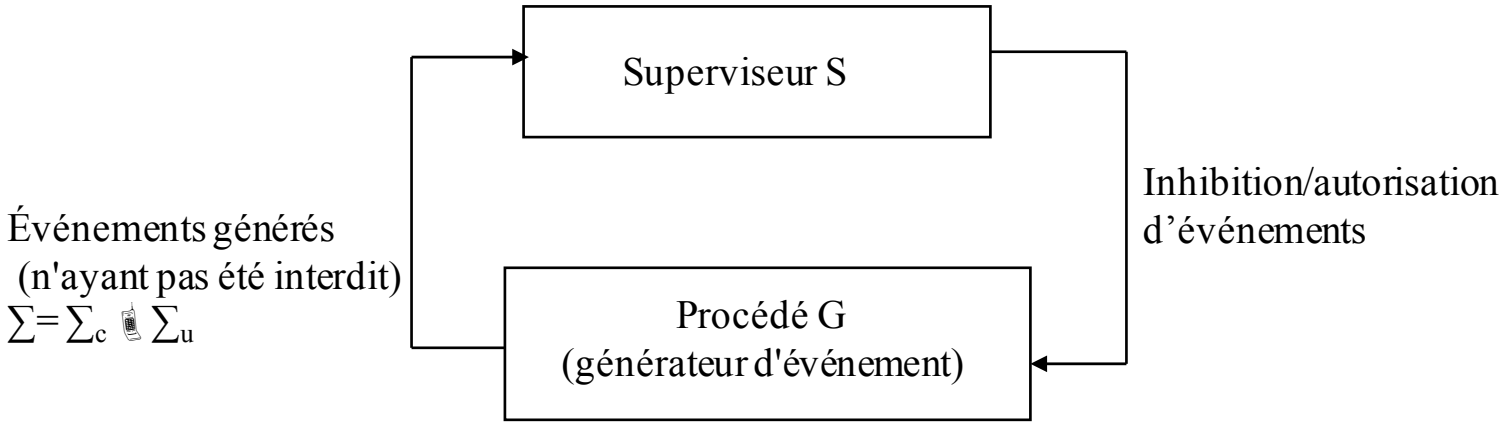
repos = (panne, μ)


Machine 1
Machine 2
repos1
repos2
panne1
panne2
occupé1
occupé2
q_{00}
q_{10}
q_{20}
q_1
q_2
q_{01}
q_{11}
q_{21}
q_{02}
q_{12}
q_{22}
α_1
β_1
λ_1
μ_1
α_2
β_2
λ_2
μ_2
$\Sigma_1 = \{\alpha_1,\beta_1,\lambda_1,\mu_1\}$
$\Sigma_2 = \{\alpha_2,\beta_2,\lambda_2,\mu_2\}$
$G = \{Q,\Sigma,\phi,q_0,Qm\}$
$Q = (q_i,q_j)$
$\Sigma = \Sigma_1 \cup \Sigma_2$
$\phi : Q \times \Sigma \rightarrow Q$
$L = L_1 || L_2$

Concept de supervision

Principe de la supervision :

- Le procédé génère des événements spontanément qui peuvent être commandable ou non



- ∑ est divisé en 2 sous-ensembles: ∑ = $\sum_c$ $\sum_u$

  $\sum_c$ événements commandables et $\sum_u$: événements non commandables

- $\sum_c$:le superviseur peut interdire ou autoriser ces événements à tout instant

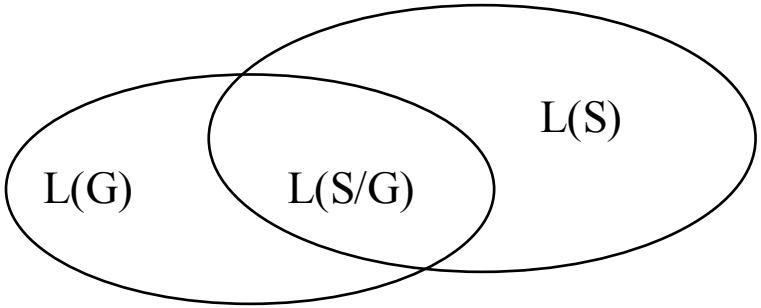
-

- $\sum_u$: le superviseur ne peut exercer d'action sur ces événements donc ils sont toujours autorisés quelque soit l'événement généré

Le superviseur est définit formellement par $S:L(G) \rightarrow \Gamma$

Le comportement généré par S/G est le comportement en boucle fermée et il est décrit par le langage L(S/G) définit :

- $\varepsilon \in L(S/G)$
- $w\sigma \in L(S/G)$ ssi $w \in L(S/G)$, $\sigma \in S(w)$ et $w\sigma \in L(G)$

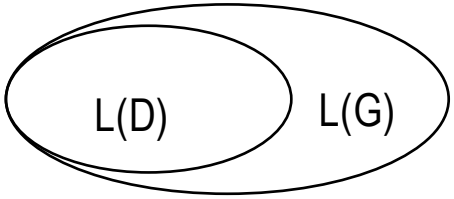


En pratique on représentera S et G par des automates fonctionnant en parallèle. Un événement σ peut avoir lieu lorsque SxG est dans l'état (x,q) ssi σ est possible dans S et G. Il en résulte un changement d'état (x',q') avec $x \rightarrow x'$ et $q \rightarrow q'$ les évolutions dans S et G sous l'occurrence de σ

Concept de commandabilité

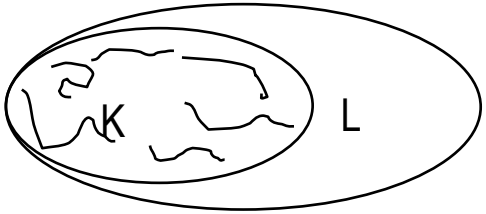
Problématique :

On veut limiter le comportement du système à celui désiré (par cahier des charges) à partir du comportement libre du procédé . Pour cela on va utiliser la commandabilité des événements

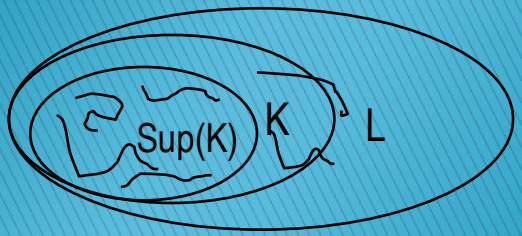


Principe de commandabilité :

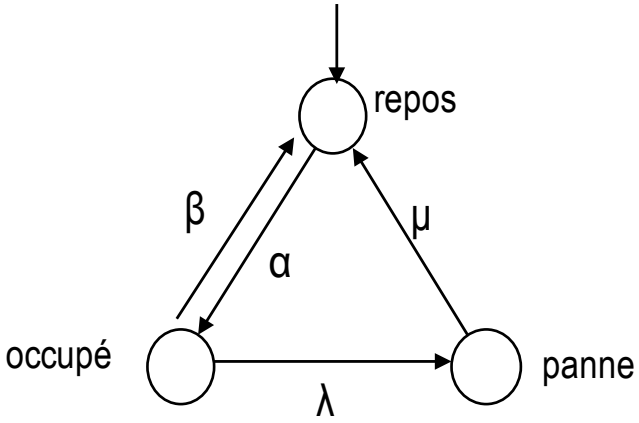
Un langage K est commandable ssi w ∈K, σ ∈∑u tel que wσ ∈L donc wσ ∈K



Si K n'est pas commandable il existe un sous ensemble sup(K) commandable par rapport à L. Il faut s'assurer que sup(K) ≥K_{min}



Application sur l'exemple de la machine :



$\sum_c = \{\alpha, \mu\}$

$\sum_u = \{\beta, \lambda\}$

$L(G) = (\alpha\beta + \alpha\lambda\mu)^* + (\varepsilon + \alpha + \alpha\lambda)$

Soit K= {αλμ} ce langage n'est pas commandable car on ne peut pas empêcher l'événement β. Plus formellement

$\overline{K} = \{\alpha, \alpha\lambda, \alpha\lambda\mu\}$

$\overline{K} \sum_u = \{\alpha\beta, \alpha\lambda, \alpha\lambda\beta, \alpha\lambda\lambda, \alpha\lambda\mu\beta, \alpha\lambda\mu\lambda\}$ l'intersection avec L(G) donne {αβ, αλ}

Soit K={αβ, αλ}

$\overline{K} = \{\alpha, \alpha\lambda, \alpha\beta\}$

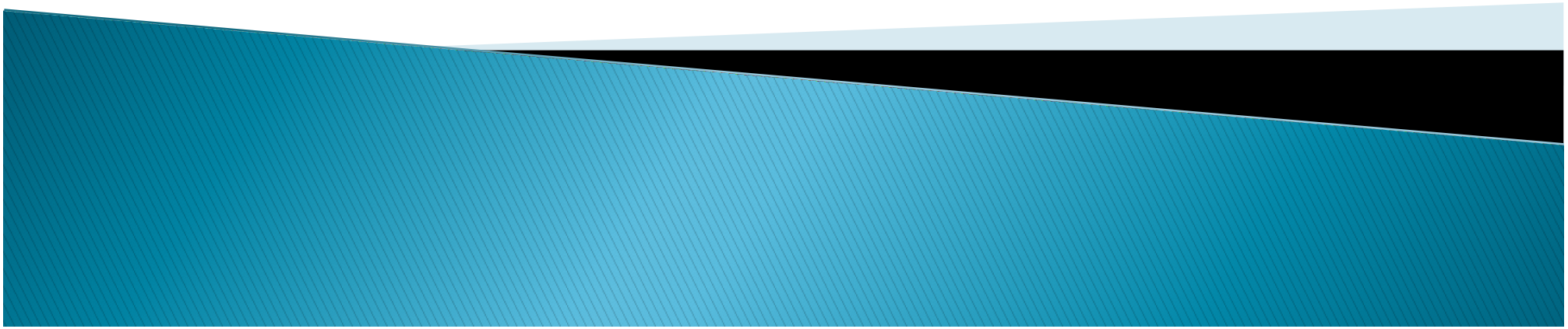
$\overline{K} \sum_u = \{\alpha\beta, \alpha\lambda, \alpha\lambda\beta, \alpha\lambda\lambda, \alpha\beta\lambda, \alpha\beta\beta\}$ l'intersection avec L(G) donne {αβ, αλ}

Donc K est commandable

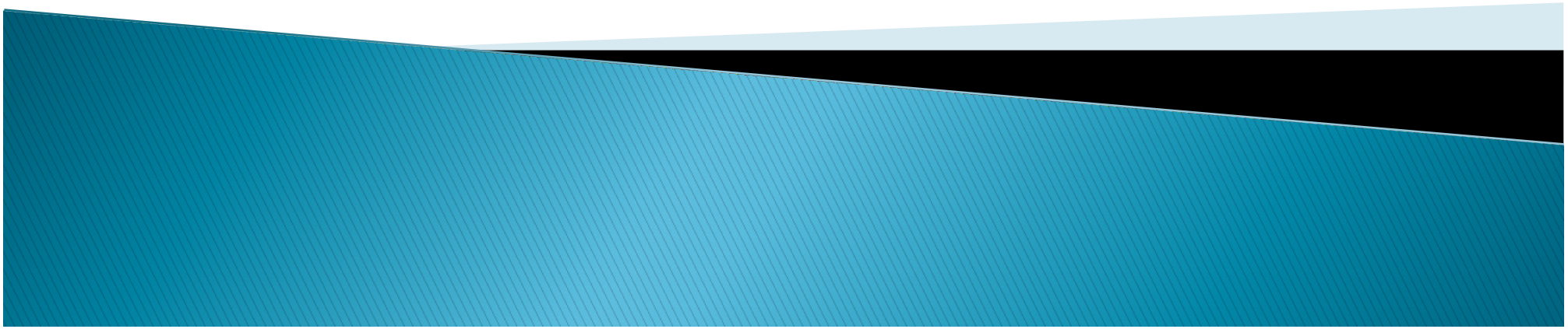
Existence d'un superviseur :

Soit un procédé G et un langage de spécification K . La théorie de supervision a établit le théorème suivant :

- Pour tout langage K c L(G) il existe un superviseur S ssi:
  - K est à préfixe fermé
  - K est commandable par rapport à L(G)



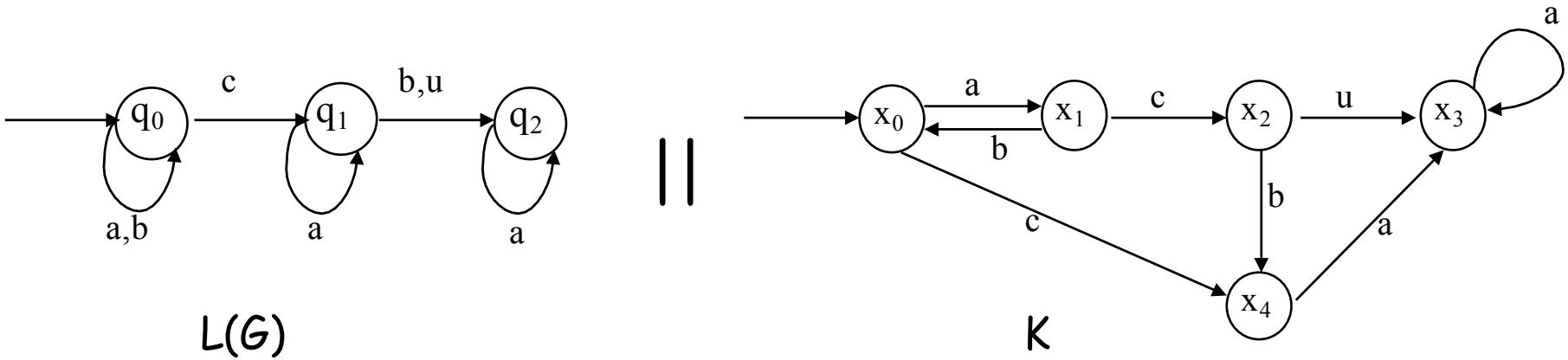
Synthèse du superviseur



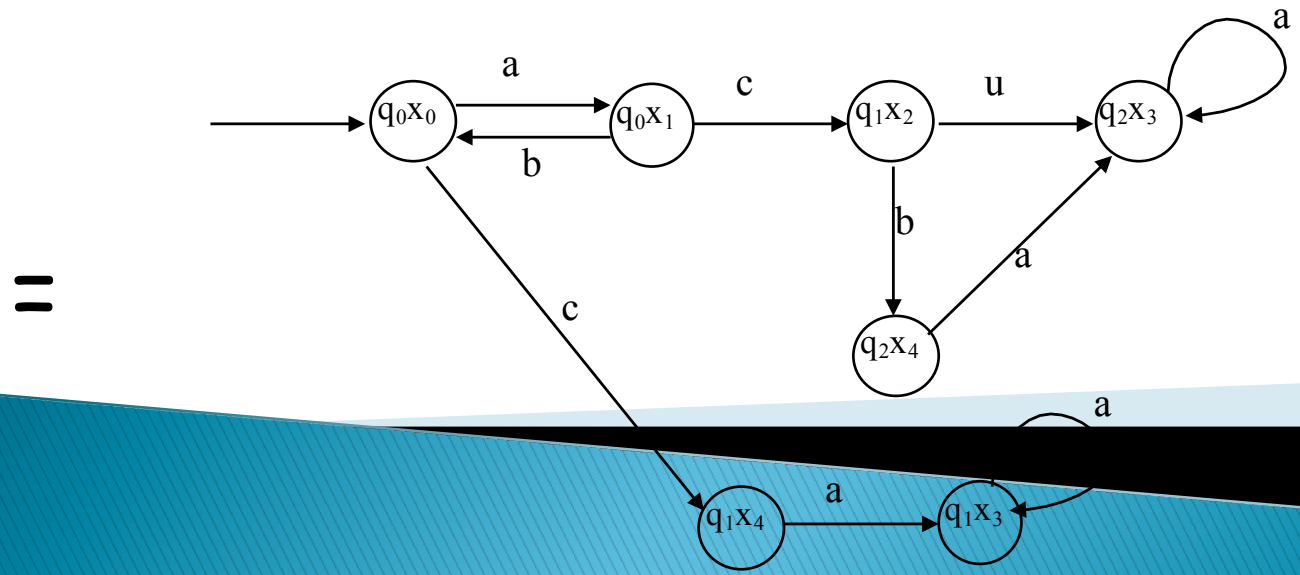
Pour faire la synthèse, on utilise l'algorithme de Kumar qui se décompose en 4 étapes :

- Construire le produit synchrone entre L(G) et K
- Déterminer les états défendus
- Déterminer les états faiblement défendus
- Éliminer les états défendus ,les états faiblement défendus ainsi que les transitions liées à ces états

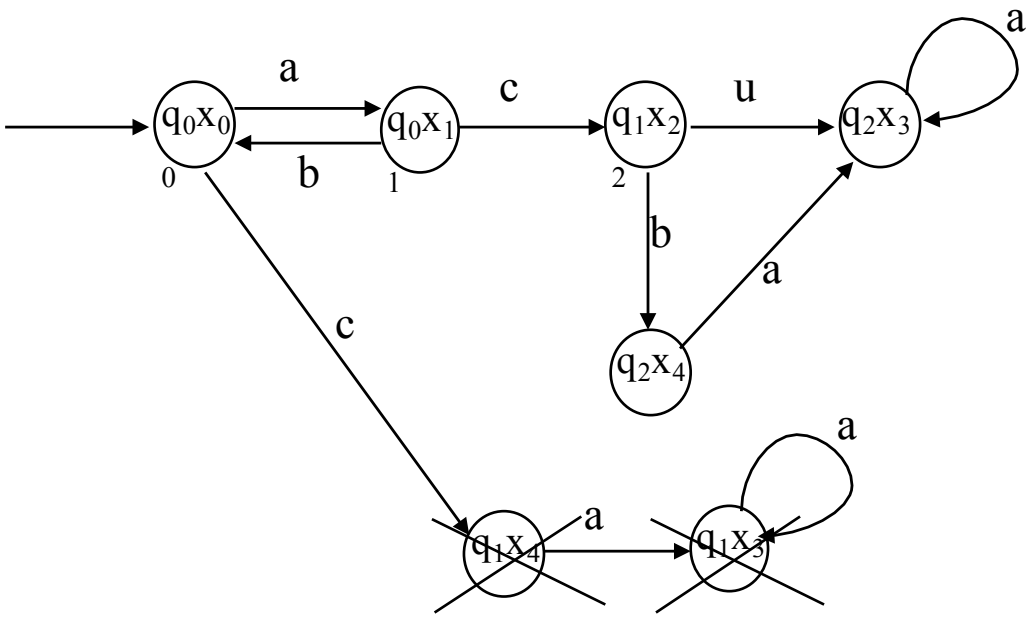
Étape 1 : Produit synchrone entre L(G) et K :



avec u événement incontrôlable

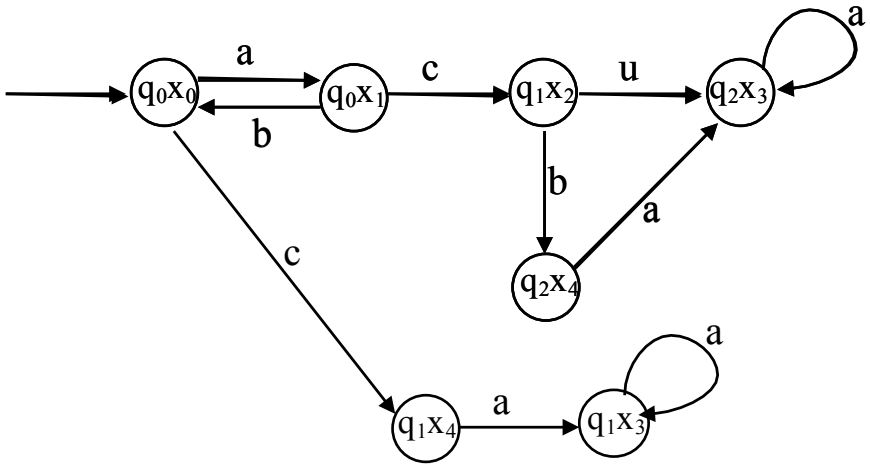


Étape 2 : Déterminer les états défendus :



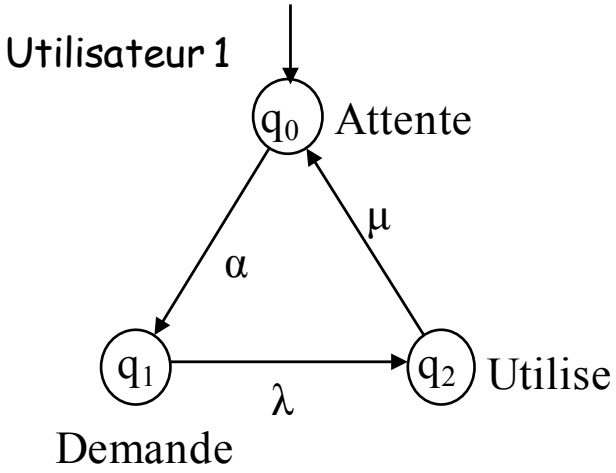
Étape 3 : Déterminer des états faiblement défendus

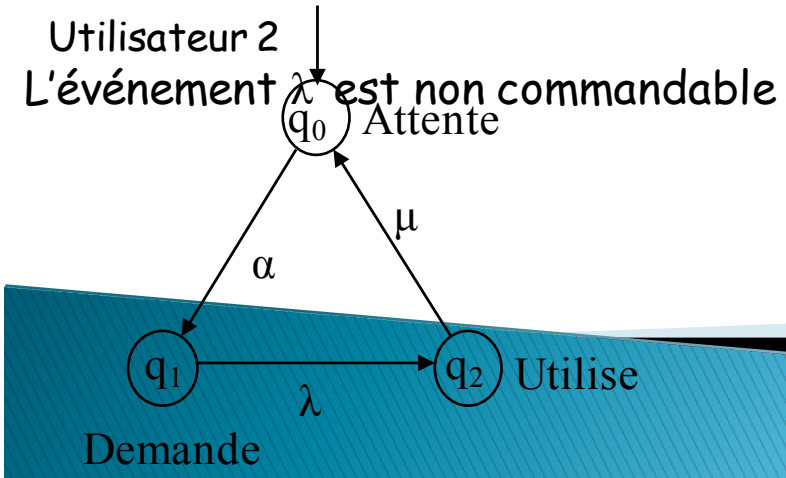
Étape 4 : Éliminer les états défendus ,les états faiblement défendus ainsi que les transitions liées à ces états

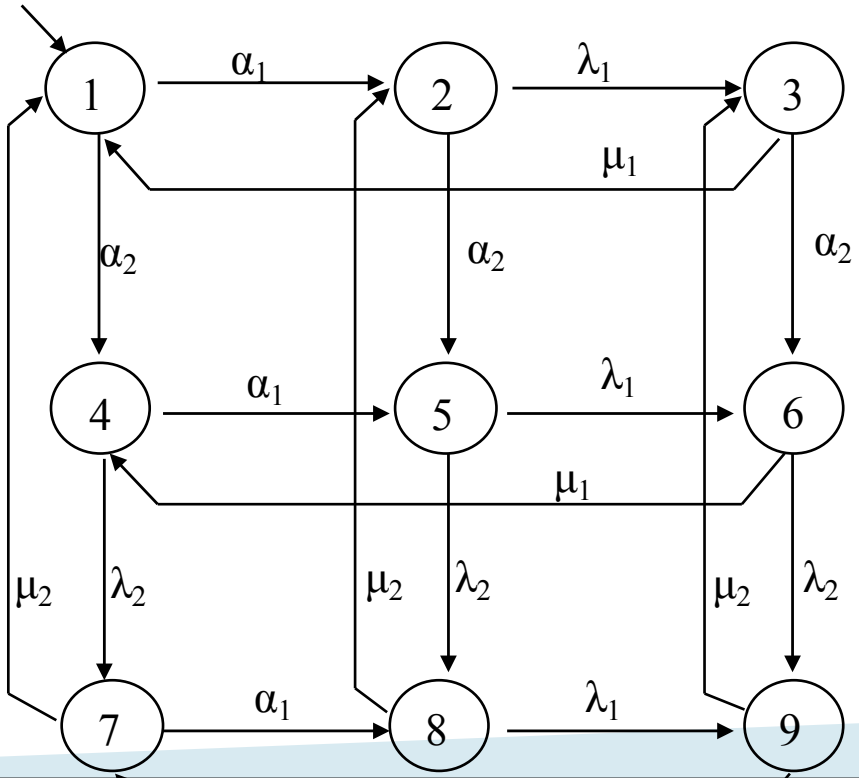


Sup(K) : Automate du superviseur

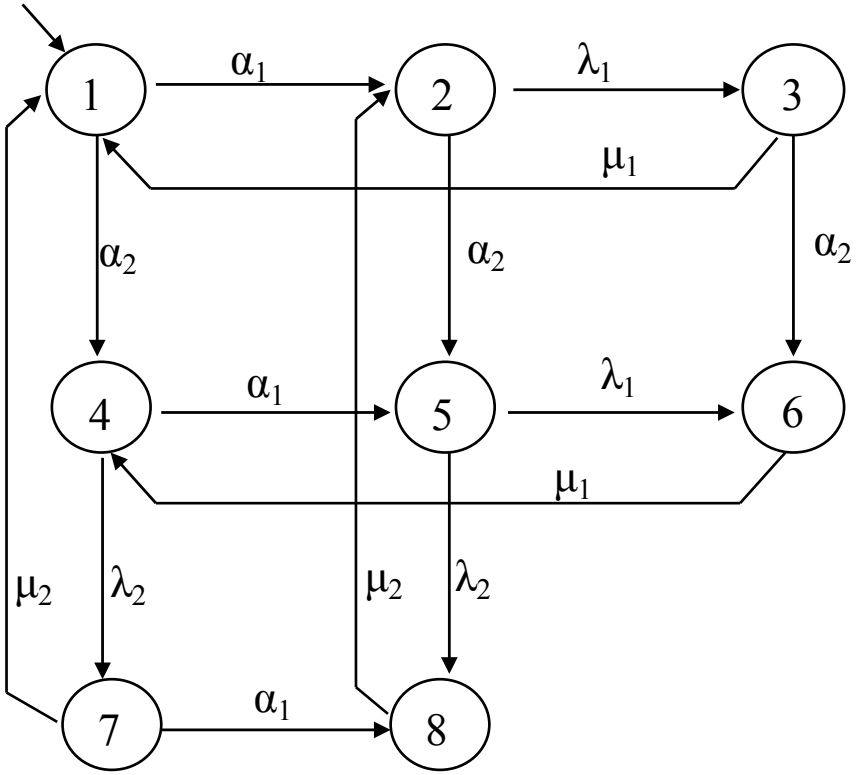
Autre exemple : On considère un système constitué de 2 utilisateurs partageant une ressource commune



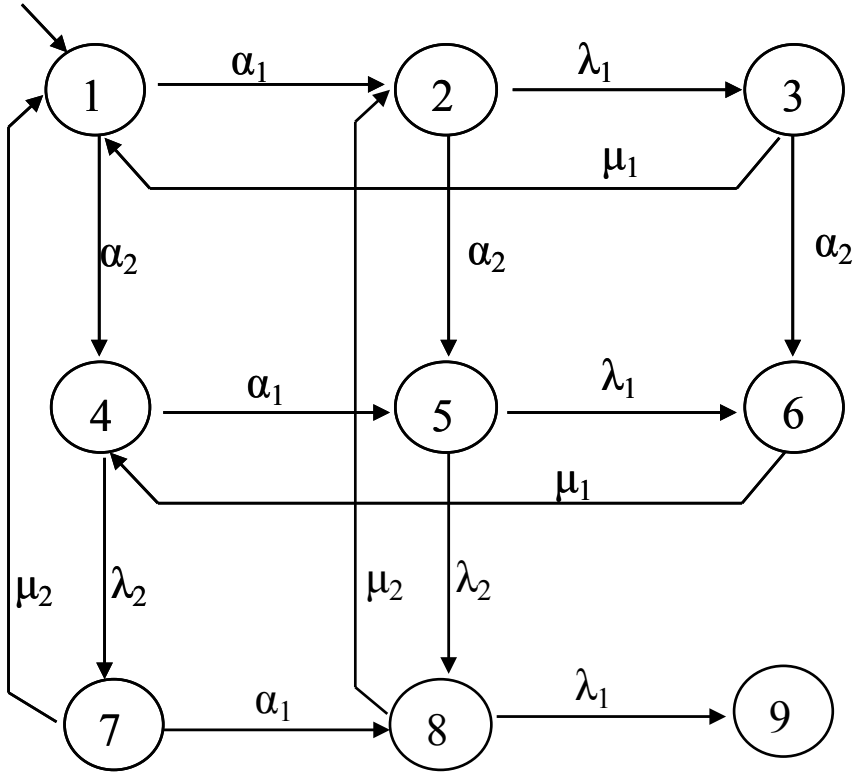




On souhaite que les 2 utilisateurs n'utilisent pas la ressource en même temps :



Au final, le graphe représente le processus et la spécification



Conclusions et perspectives

Conclusions :

- La théorie de supervision est basée sur l'autorisation ou l'interdiction de certains événements
- langage désiré n'est pas commandable, on utilise l'algorithme de Kumar pour réduire ce langage en un langage commandable

Perspectives :

- Voir d'autres méthodes de synthèse de la commande
- Utiliser d'autres exemples plus complexes