

Introduction à la théorie du contrôle par supervision

Laurent Piétrac

INSA Lyon – Laboratoire Ampère

3 juin 2019



Plan de l'exposé

- 1 Fondements de la théorie du contrôle par supervision
 - Principe
 - Concept d'événement
 - Formalismes
 - Modèles
 - Propriétés utilisées
- 2 Problèmes traités dans l'approche centralisée
 - Problème 1 : existe-t-il un procédé sous contrôle ?
 - Exemple
 - Problème 2 : ce procédé sous contrôle existe-t-il ?

Principe du contrôle par supervision

C'est l'application :

- sur des Systèmes à Événements Discrets (SED)
- du principe de bouclage
- pour la prise de décisions
- en fonction de l'objectif à atteindre

Le concept d'événement

Définition

Un événement peut correspondre à tout changement d'état du système.
Un événement est instantané.

Exemples

- Il peut être identifié par rapport à une action :
Appuyer sur un bouton, mise en route d'une pompe. . .
- Il peut être considéré comme spontanément généré par la nature :
L'occurrence d'un éclair d'orage, la panne d'un ordinateur
- Il peut être le résultat de plusieurs conditions subitement vraies :
Pompe 1 arrêtée et pompe 2 en marche

Systèmes conduits par les événements

Systèmes conduits par le temps :

- Le temps peut être discret ou continu
- La variable temps évolue de manière indépendante
- L'état du système, ses entrées, ses sorties évoluent en fonction du temps

Systèmes conduits par les événements :

- Pas de variable temps
- L'état du système évolue en fonction des événements
- Les événements peuvent correspondre aussi bien à une évolution de l'état, des entrées ou des sorties

- 1 Fondements de la théorie du contrôle par supervision
 - Principe
 - Concept d'événement
 - **Formalismes**
 - Modèles
 - Propriétés utilisées
- 2 Problèmes traités dans l'approche centralisée

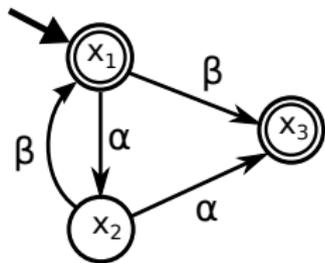
Automates déterministes

$$G = \langle Q, \Sigma, \delta, q_0, Q_m \rangle$$

- Q : ensemble appelé espace d'états de G
- Σ : ensemble appelé alphabet d'entrée de G
- $\delta : Q \times \Sigma \rightarrow Q$ fonction partielle de transition
- q_0 : état initial, $q_0 \in Q$
- Q_m : ensemble des états marqués, $Q_m \subseteq Q$

Exemple

- États : $Q = \{x_1, x_2, x_3\}$, $q_0 = x_1$,
 $Q_m = \{x_1, x_3\}$
- Alphabet : $\Sigma = \{\alpha, \beta\}$
- Transitions : $\delta =$
 $\{(x_1, \alpha, x_2), (x_1, \beta, x_2), (x_2, \beta, x_1), (x_2, \alpha, x_3)\}$



Mots

Comportement du système

A partir de l'état initial, un comportement possible du système se traduit par une succession d'événements, c'est-à-dire un mot.

Exemples : $\alpha.\beta$, $\alpha.\beta.\alpha.\beta$, $\alpha.\beta.\alpha\dots$

ε

Un mot particulier est le mot vide : ε (il ne se passe rien).

Σ^*

L'ensemble des mots sur l'alphabet Σ est noté Σ^* .

Fonction de transition étendue

La fonction de transition δ définie sur Σ peut être étendue à Σ^* :

$$\begin{aligned}\delta(q, \varepsilon) &= q \\ \delta(q, s\sigma) &= \delta(\delta(q, s), \sigma) \text{ pour } s \in \Sigma^* \text{ et } \sigma \in \Sigma\end{aligned}$$

Langages

Comportements des systèmes

Il est exprimé par les langages :

- Le langage $L(G)$ représente l'ensemble des mots générés par G à partir de l'état initial :

$$L(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \neq !\}$$

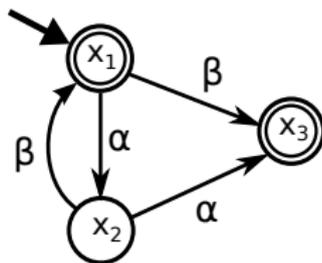
- Le langage $L_m(G) \subseteq L(G)$ marqué par G contient tous les mots tels qu'à partir de l'état initial, la fonction de transition étendue contienne ce mot et mène à un état marqué :

$$L_m(G) = \{s \in \Sigma^* \mid \delta(q_0, s) \in Q_m\}$$

Langages

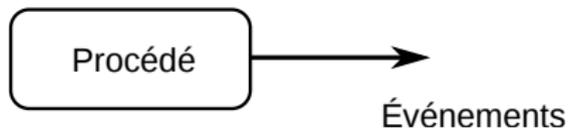
Exemple

- Le langage généré : $L(G) = \{\varepsilon, \alpha, \beta, \alpha.\alpha, \alpha.\beta, \alpha.\beta.\alpha, \alpha.\beta.\beta, \dots\} = (\alpha.\beta)^* . (\alpha^2 + \alpha + \beta + \varepsilon)$
- Le langage marqué : $L_m(G) = (\alpha.\beta)^* . (\alpha^2 + \beta + \varepsilon)$



- 1 Fondements de la théorie du contrôle par supervision
 - Principe
 - Concept d'événement
 - Formalismes
 - **Modèles**
 - Propriétés utilisées
- 2 Problèmes traités dans l'approche centralisée

Le procédé



Modélisation

- Le procédé est constitué de plusieurs composants
- Chaque composant est modélisé par un automate à états, avec son alphabet
- Le procédé global est construit par composition des automates des composants

La composition d'automates

Modèles de composants

$$G_1 = (Q_1, \Sigma_1, \delta_1, q_{0,1}, Q_{m,1})$$

$$G_2 = (Q_2, \Sigma_2, \delta_2, q_{0,2}, Q_{m,2})$$

Modèle du procédé

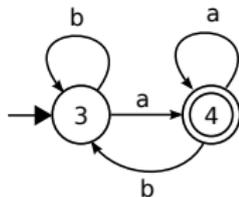
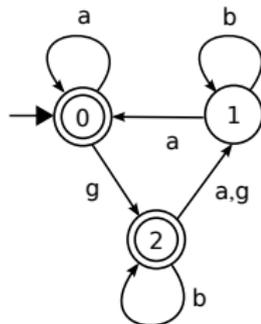
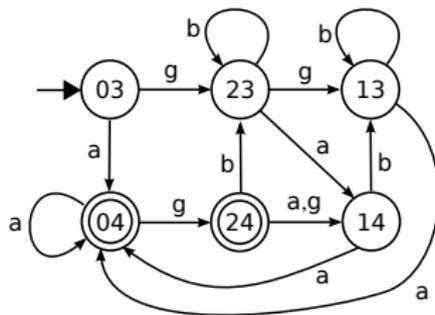
$$G_1 \parallel G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, (q_{0,1}, q_{0,2}), Q_{m,1} \times Q_{m,2})$$

avec δ définie de la façon suivante :

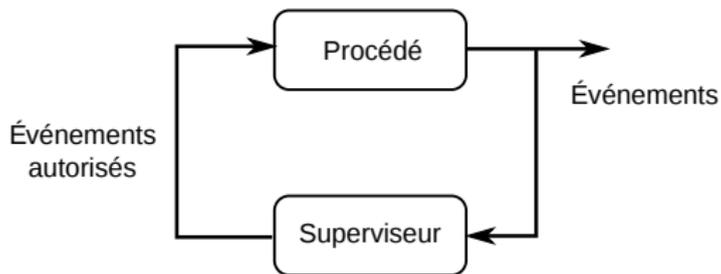
$$\delta((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)) & \text{si } \delta(q_1, \sigma)! \text{ et } \delta(q_2, \sigma)! \\ (\delta_1(q_1, \sigma), q_2) & \text{si } \delta(q_1, \sigma)! \text{ et } \sigma \notin \Sigma_2 \\ (q_1, \delta_2(q_2, \sigma)) & \text{si } \sigma \notin \Sigma_1 \text{ et } \delta(q_2, \sigma)! \\ \text{non défini} & \text{sinon} \end{cases}$$

Note : « $\delta(q_1, \sigma)!$ » se lit « $\delta(q_1, \sigma)$ existe ».

Exemple de composition d'automates

 G_1

 G_2

 $G_1 \parallel G_2$


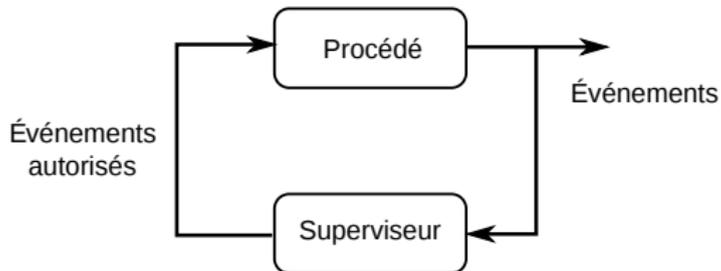
Le procédé sous contrôle : S/G



- Le procédé évolue spontanément
- Le superviseur restreint les comportements possibles :

$$L(S/G) \subseteq L(G)$$

Comment le superviseur restreint-il le comportement ?



Σ : ensemble des événements générés par le procédé

$$\Sigma = \Sigma_c \cup \Sigma_u \quad \text{et} \quad \Sigma_c \cap \Sigma_u = \emptyset$$

avec :

- Σ_c : ensemble des événements contrôlables c'est-à-dire qui peuvent être interdits par le superviseur
- Σ_u : ensemble des événements incontrôlables

Définition formelle du superviseur

Définition

$S : L(G) \rightarrow 2^\Sigma$ avec 2^Σ ensemble de tous les sous-ensembles de Σ

Interprétation

Quel que soit s généré par G , $S(s)$ est l'ensemble des événements autorisés par S . Donc : $\forall s \in L(G) : \Sigma_u \subseteq S(s)$

Langage généré $L(S/G)$

Le langage $L(S/G)$ est défini par :

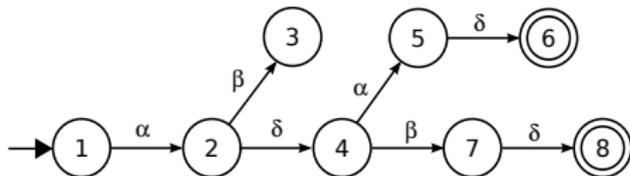
- ① $\varepsilon \in L(S/G)$
- ② $[s \in L(S/G) \text{ et } \sigma \in S(s) \text{ et } s\sigma \in L(G)] \Leftrightarrow s\sigma \in L(S/G)$

Langage marqué $L_m(S/G)$

$$L_m(S/G) = L(S/G) \cap L_m(G)$$

Exemple de superviseur

Le procédé G



$$\Sigma = \{\alpha, \beta, \delta\}, \Sigma_c = \{\alpha, \beta\}, \Sigma_u = \{\delta\}$$

Exigence

Est-il possible d'autoriser tous les comportements mais d'interdire les états 5 et 6 ? oui (ouf!)

Le superviseur

$$S(\varepsilon) = \Sigma \quad S(\alpha) = \Sigma \quad S(\alpha.\delta) = \{\beta, \delta\} \quad S(\alpha.\delta.\beta) = \Sigma$$

- 1 Fondements de la théorie du contrôle par supervision
 - Principe
 - Concept d'événement
 - Formalismes
 - Modèles
 - Propriétés utilisées
- 2 Problèmes traités dans l'approche centralisée

Contrôlabilité

Contrôlabilité des événements

Un événement peut être contrôlable ou pas : $\Sigma = \Sigma_u \cup \Sigma_c$

Contrôlabilité d'un langage

Le langage K est contrôlable par rapport au procédé G et à l'ensemble des événements incontrôlables Σ_u si et seulement si : $\overline{K} \cdot \Sigma_u \cap L(G) \subseteq \overline{K}$

Existence d'un superviseur

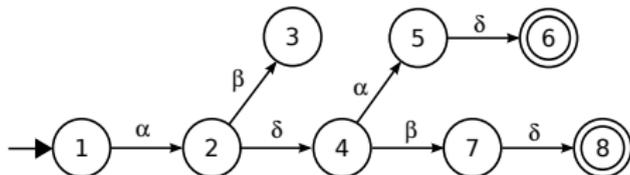
Soit $K \subseteq L(G)$ avec $K \neq \emptyset$. Il existe un superviseur S pour G tel que $L(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

Rappel : \overline{K} est le préfixe-clos de K .

Exemple : Si $K = \{abc, de\}$ alors $\overline{K} = \{\varepsilon, a, ab, abc, d, de\}$.

Blocage

Le procédé G



Procédé bloquant : à partir de l'état 3 impossible d'atteindre 6 ou 8.

Exigence de fin de tâche

Est-il possible d'imposer $K = \{\alpha.\delta.\beta.\delta\}$?

Existence d'un superviseur non bloquant

Soit $K \subseteq L_m(G)$ avec $K \neq \emptyset$. Il existe un superviseur non bloquant S pour G tel que $L_m(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

- 1 Fondements de la théorie du contrôle par supervision
- 2 Problèmes traités dans l'approche centralisée
 - Problème 1 : existe-t-il un procédé sous contrôle ?
 - Exemple
 - Problème 2 : ce procédé sous contrôle existe-t-il ?

Problème classique 1

On donne :

- Le procédé non contrôlé
- Des exigences

On cherche :

- A vérifier que les exigences peuvent être respectées
- A déterminer le procédé sous contrôle
- A déterminer le superviseur

Problème 1 : démarche d'étude détaillée (1/2)

On donne :

- Les modèles des composants : automates G_i
- Des exigences : aussi sous forme d'automates E_j

On construit :

- Le modèle G du procédé global par composition des G_i
- Le modèle $G \parallel E$ du procédé respectant les exigences par composition des G_i et E_j

Problème 1 : démarche d'étude détaillée (2/2)

On donne :

- Les modèles des composants : automates G_i
- Des exigences : aussi sous forme d'automates E_j

On cherche :

- A vérifier que $L_m(G \parallel E)$ est contrôlable par rapport à G et Σ_u :
 - Si oui il existe un superviseur
 - Sinon il faut trouver le plus grand sous-langage de $L_m(G \parallel E)$ contrôlable (**le suprême contrôlable**). Un algorithme de calcul à partir de $G \parallel E$ permet de trouver le résultat :
 - Si le résultat n'est pas \emptyset , les spécifications peuvent être respectées
 - Sinon... (mais cela n'arrive jamais ! ;-)

Modélisation des exigences

Un seul point de vue possible : interdire des événements

Pourtant la littérature présente 4 types d'exigence :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Modélisation des exigences

Un seul point de vue possible : interdire des événements

Pourtant la littérature présente 4 types d'exigence :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Modélisation des exigences

Un seul point de vue possible : interdire des événements

Pourtant la littérature présente 4 types d'exigence :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Modélisation des exigences

Un seul point de vue possible : interdire des événements

Pourtant la littérature présente 4 types d'exigence :

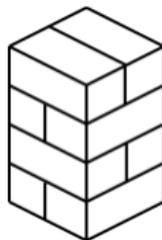
- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

- 1 Fondements de la théorie du contrôle par supervision
- 2 Problèmes traités dans l'approche centralisée
 - Problème 1 : existe-t-il un procédé sous contrôle ?
 - **Exemple**
 - Problème 2 : ce procédé sous contrôle existe-t-il ?

Le cahier des charges



(a) Machine 1



(b) Stock intermédiaire



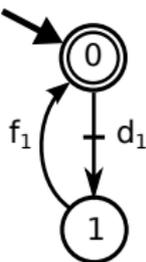
(c) Machine 2

Exigences

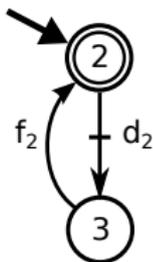
- Le stock intermédiaire ne peut contenir qu'une seule pièce
- La machine 2 ne peut produire que s'il y a une pièce dans le stock

Le procédé non contrôlé

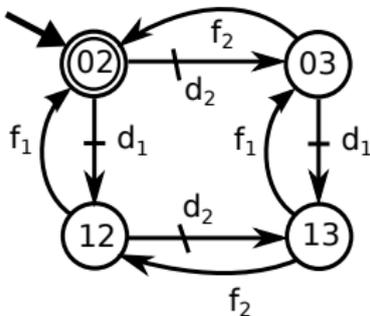
Machine 1



Machine 2



Le procédé G

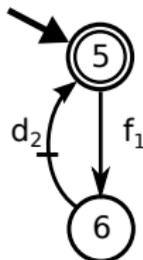


La modélisation des exigences

Exigences

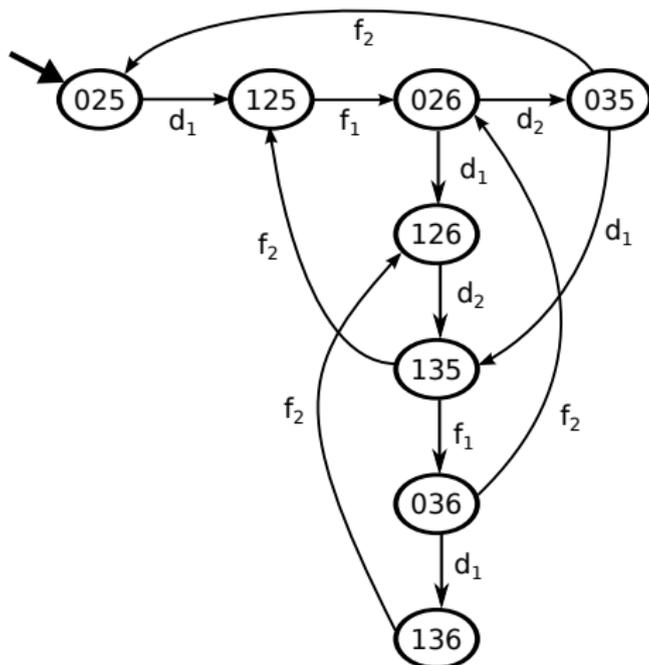
- Le stock intermédiaire ne peut contenir qu'une seule pièce
- La machine 2 ne peut produire que s'il y a une pièce dans le stock

Le modèle E



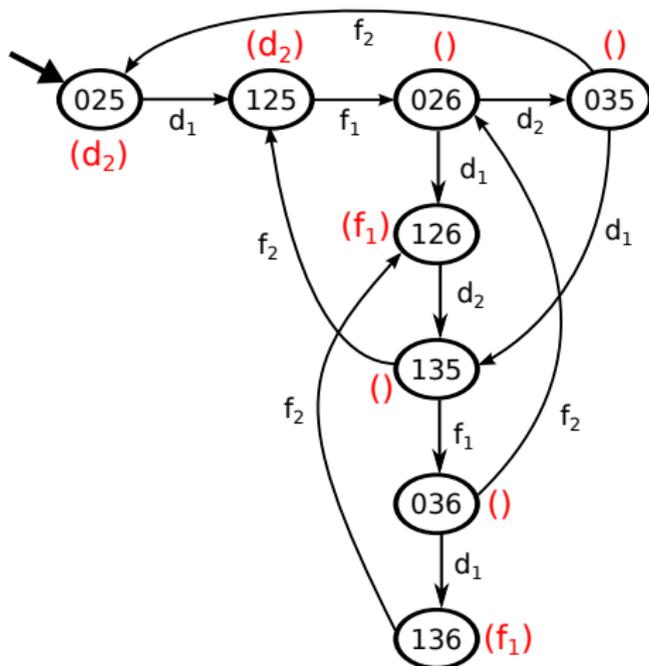
Le procédé respectant les exigences

Le modèle $G \parallel E$



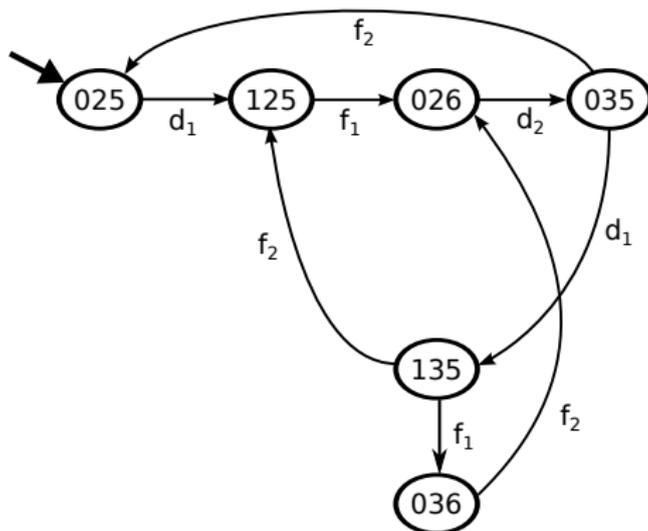
Les événements interdits par les exigences

Le modèle $G \parallel E$ (annoté)



Le procédé sous contrôle

Le modèle S/G



Le superviseur

La fonction

$$S(\varepsilon) = \Sigma - \{d_2\}$$

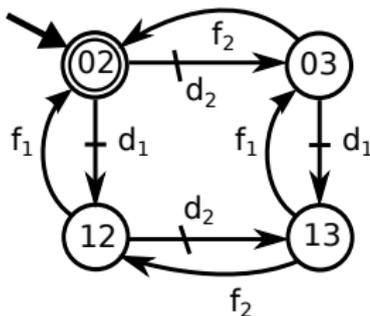
$$S(d_1) = \Sigma - \{d_2\}$$

$$S(d_1.f_1) = \Sigma - \{d_1\}$$

$$S(d_1.f_1.d_2) = \Sigma$$

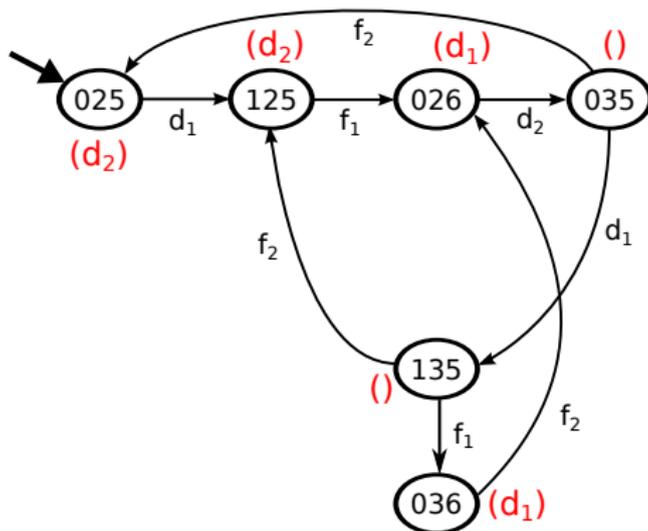
...

Le procédé G



Une représentation du superviseur

Un automate (étiqueté)



- 1 Fondements de la théorie du contrôle par supervision
- 2 Problèmes traités dans l'approche centralisée
 - Problème 1 : existe-t-il un procédé sous contrôle ?
 - Exemple
 - Problème 2 : ce procédé sous contrôle existe-t-il ?

Problème classique 2

On donne :

- Le procédé sous contrôle
- Le procédé non contrôlé

On cherche :

- A vérifier que le procédé sous contrôle est contrôlable
- A déterminer le superviseur

Problème 2 : démarche d'étude

On a

- Le modèle du procédé G
- Le modèle du procédé sous contrôle H tel que $L(H) \subseteq L(G)$ et $L_m(H) \subseteq L_m(G)$

On cherche

- A vérifier que $\mathcal{L}_m(H)$ est contrôlable par rapport à G et Σ_u :
 - Si oui il existe un superviseur
 - Sinon il faut trouver le plus grand sous-langage de $\mathcal{L}_m(H)$ contrôlable (**le suprême contrôlable**). Un algorithme de calcul à partir de H permet de trouver le résultat :
 - Si le résultat n'est pas \emptyset , les spécifications peuvent être respectées
 - Sinon...