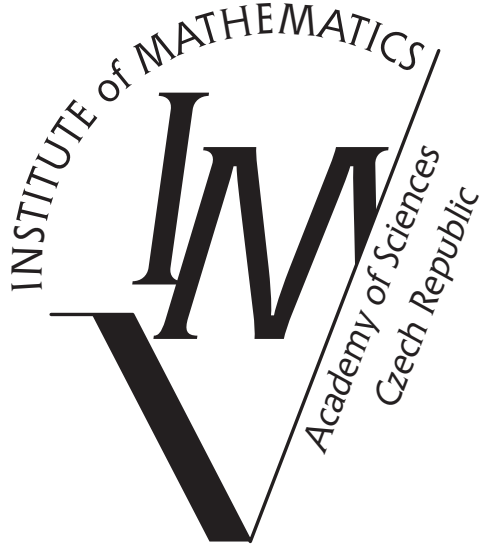
Contrôle supervisé modulaire et hiérarchique des systèmes à événements discrets sous observations partielles

Jan Komenda (travail avec T. Masopust)

23 Novembre 2023
MSR 2023, Toulouse

Institut des Mathématiques, Academie Tchèque des Sciences



Résumé

- Systèmes à événements discrets representés par products résaux de Petri ou produits d'automates et leurs extensions
- Contrôle supervisé guarantie la sûreté et autres propriétés de tels systèmes (manufacturier, réseaux de transports, de communication etc.)
- Problème de l'explosion combinatoire, nécesite des approches efficaces (compositionelles, modulaires) à leurs vérification et contrôle
- Dans le cas des observations partielles, contrôle supervisé souffre de complexité double exponentielle
- Approche modulaire apporte des avantages (calcul) mais auusi inconvénients (superviseurs sont plus réstrictifs et bloquants)
- Deux approches pour guarantir la permissivité maximale
- Champs d'applications divers (systèmes manufacturiers, de transport, imagerie médical: IRM)

Automates – Generateurs

Definition (Generateurs)

A generator est la structure

$$G = (Q, \Sigma, \delta, q_0, F)$$

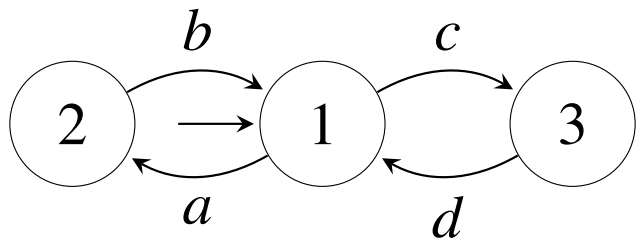
Q est un ensemble (fini) des états

Σ est un alphabet (fini) des événements)

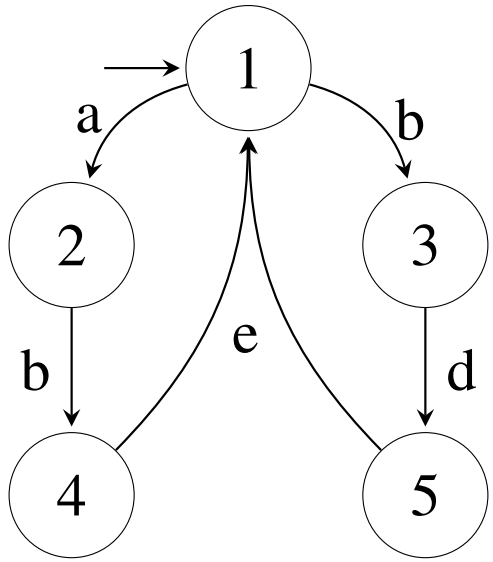
δ : $Q \times \Sigma \to Q$ est fonction (partielle) de transition

$q_0 \in Q$ état initial

$F \subseteq Q$ états finaux (marqés)



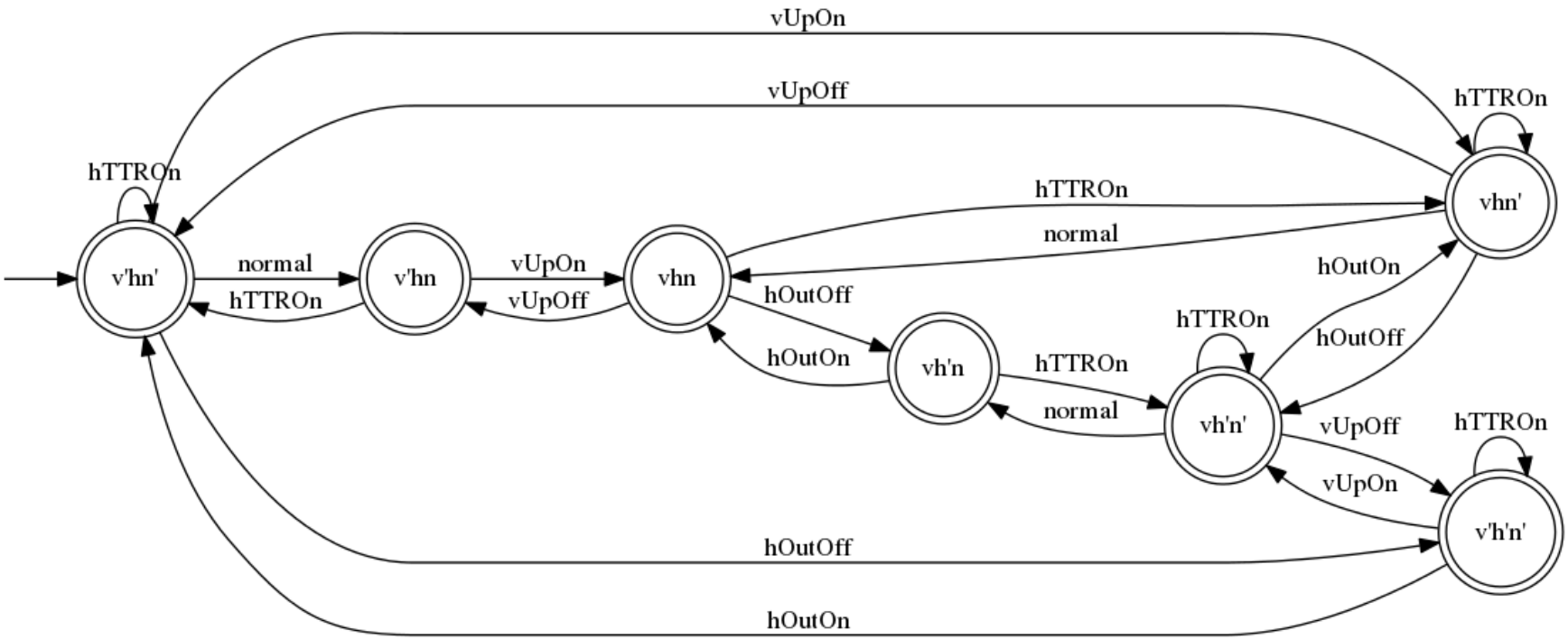
Generateurs – Comportement



Le langage géneré par $G = (Q, \Sigma, \delta, q_0)$ est

$$L(G) = \{w \in \Sigma^* \mid \delta(q_0, w) \in Q\} = \{\varepsilon, b, bd, bde, a, ab, abe, \ldots\}$$

Modèle pour l'imagerie par résonance magnétique



Objectives du Contrôle

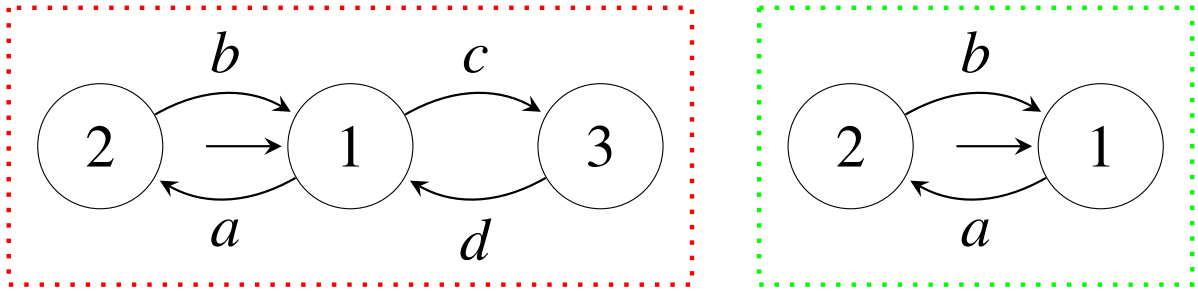
Objectives: propriétés désirées à être imposées par le contrôleur

Parmis ces objectives il y a:

- Sûreté – langage du système contrôlé ne depasse pas la spécification
- Comportement minimal – language du système contrôlé inclut au moins un langage minimal
- nonblocage – absence de "deadlocks" et "livelocks" (pour les langages marqés)

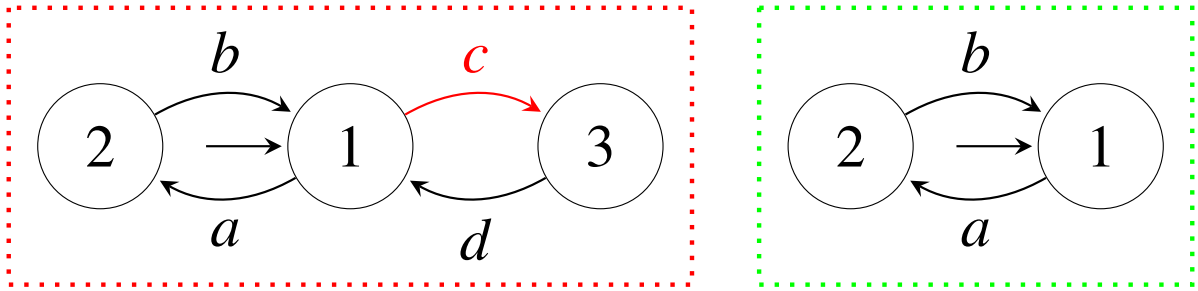
Contrôle supervisé

- Systéme (générateur) G
  - un DFA
- Pour une spécification K
  - a langage ou générateur
- Le but est de construire un superviseur S tel que $S/G \equiv K$
  - superviseur comme une application ou un automate



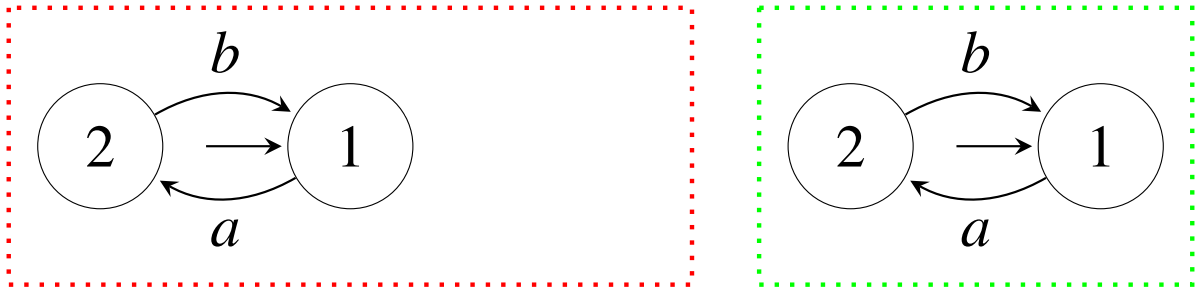
Contrôle supervisé

- Systéme (générateur) G
  - un DFA
- Pour une spécification K
  - a langage ou générateur
- Le but est de construire un superviseur S tel que $S/G \equiv K$
  - superviseur comme une application ou un automate



Contrôle supervisé

- Systéme (générateur) G
  - un DFA
- Pour une spécification K
  - a langage ou générateur
- Le but est de construire un superviseur S tel que $S/G \equiv K$
  - superviseur comme une application ou un automate

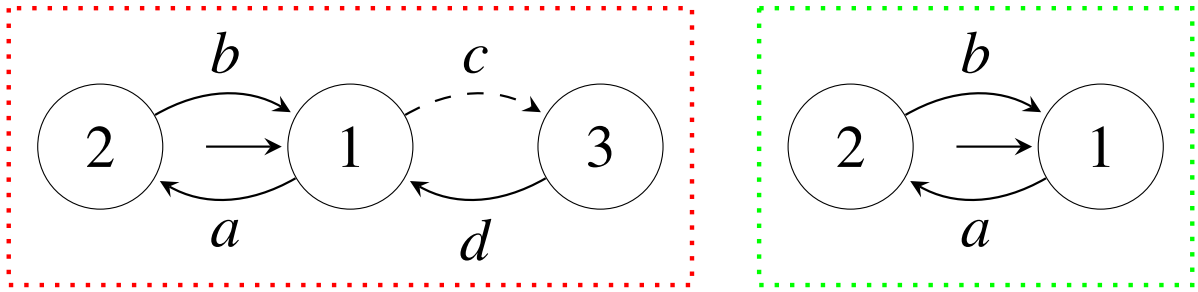


Événements Contrôlables

- Événements Incontrôlables – ne peuvent être interdits

  Raisons:

  - inhérent (fautes);
  - non supprimables due aux limitations du hardware ou des actionneurs;
  - modelisés comme incontrôlable par choix
    (top priorité, should not be disabled, tick de l'horloge etc.)

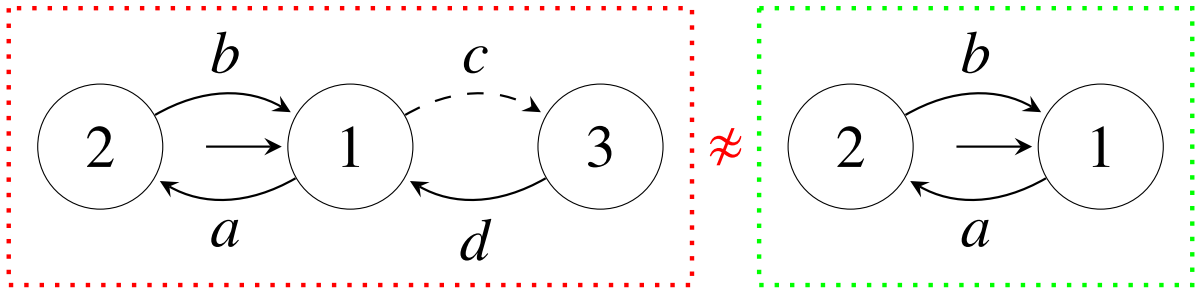


Événements Contrôlables

- Événements Incontrôlables – ne peuvent être interdits

  Raisons:

  - inhérent (fautes);
  - non supprimables due aux limitations du hardware ou des actionneurs;
  - modelisés comme incontrôlable par choix
    (top priorité, should not be disabled, tick de l'horloge etc.)

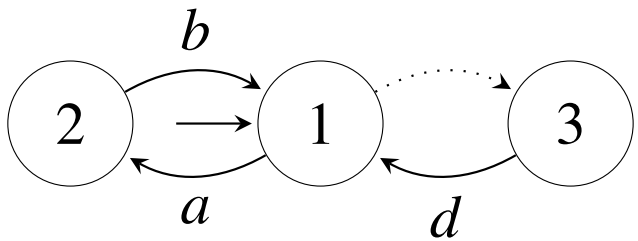


Événements Observables

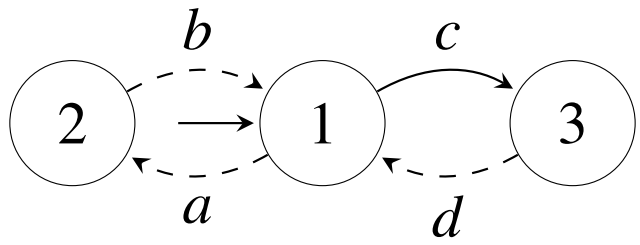
- Événements Inobservables – ne peuvent être tracés par des capteurs

  Raisons:

  - absence de capteurs pour tracer leurs occurrences
  - Événement dans une location lointaine sans la communication de son occurrence – typique pour des systems distribués répartis
  - Événement de faute: toujours inobservables



Générateur Contrôlé

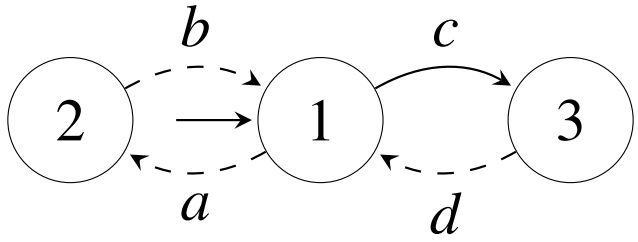


- A générateur contrôlé est (G, Σ_c, Γ), où
  - G est un générateur sur Σ,
  - $\Sigma_c \subseteq \Sigma$ sont événements contrôlables,
  - $\Sigma_u = \Sigma \setminus \Sigma_c$ sont événements incontrôlables,et
  - $\Gamma = \{\gamma \subseteq \Sigma \mid \Sigma_u \subseteq \gamma\}$ sont "control patterns".

Example

- $G = (\{1,2,3\}, \{a,b,c,d\}, \delta, 1, \{1\})$
- $\Sigma = \{a,b,c,d\}, \quad \Sigma_c = \{c\}, \quad \Sigma_u = \{a,b,d\}$
- $\Gamma = \{\{a,b,d\}, \{a,b,c,d\}\}$

Contrôle supervisé



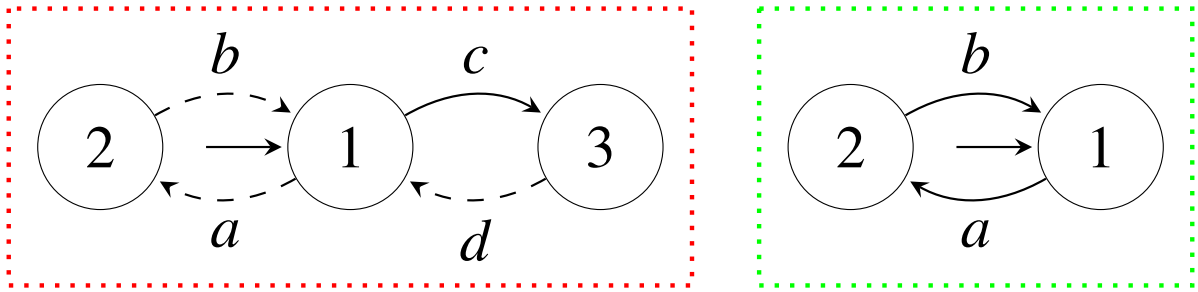
- Ub Un contrôle supervisé pour (G, Σ_c, Γ) est une application

$$S : L(G) \to \Gamma$$

Example

- $\Gamma = \{\{a,b,d\},\{a,b,c,d\}\}$
- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a,b,d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a,b,c,d\}$
- for all other strings w, $S(w) = \{a,b,d\}$

Système en boucle fermée

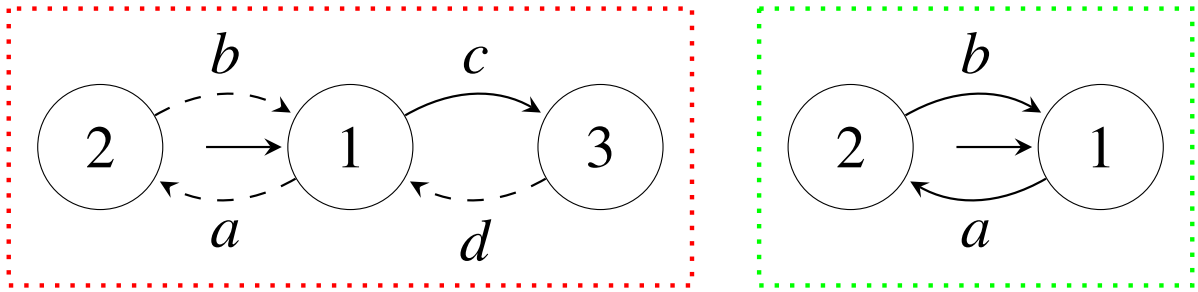


- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, \ldots\}$

Système en boucle fermée

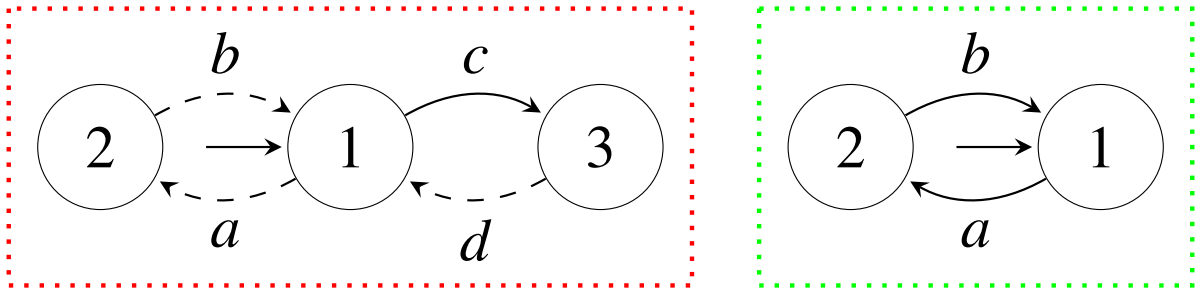


- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, a, \ldots\}$

Système en boucle fermée

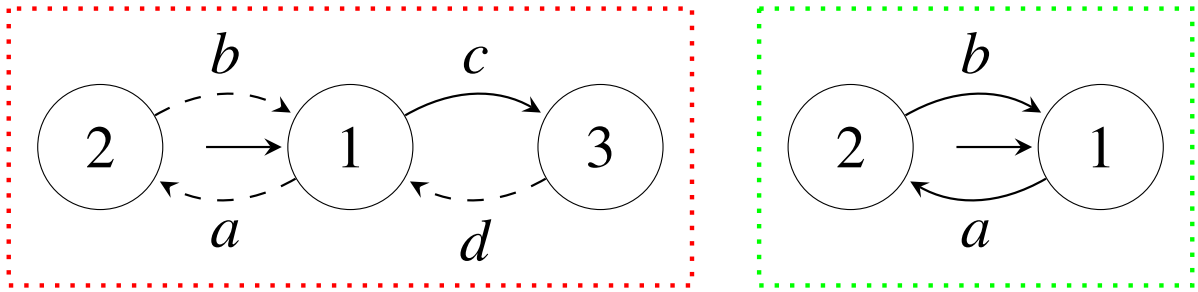


- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, a, ab, \ldots\}$

Système en boucle fermée

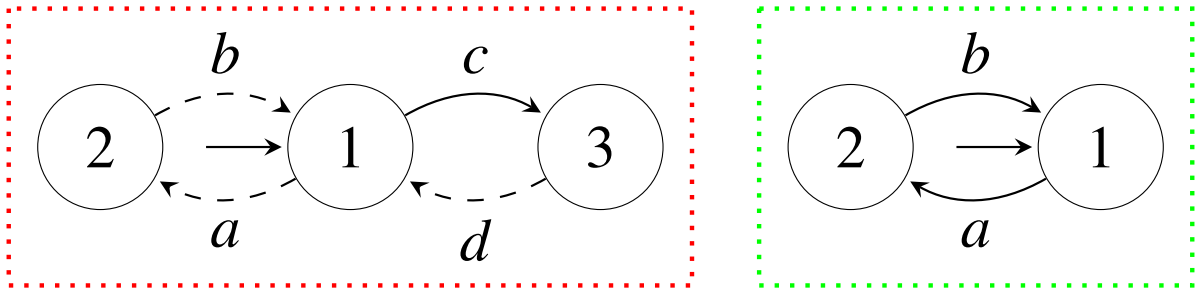


- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, a, ab, aba, \ldots\}$

Système en boucle fermée

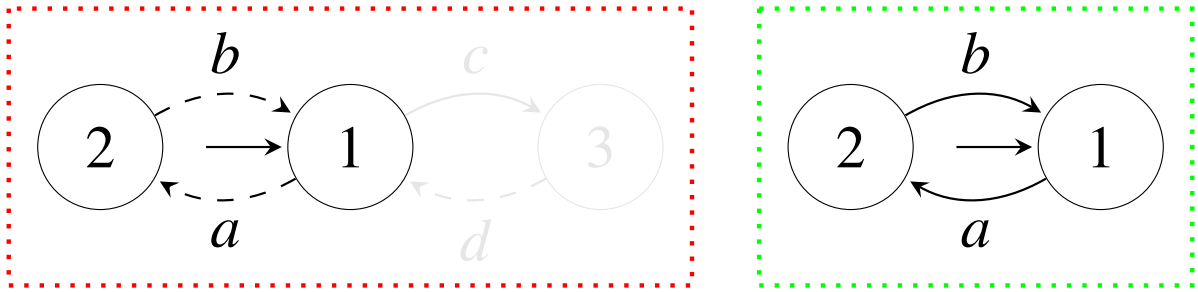


- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, a, ab, aba, abab, \ldots\}$

Système en boucle fermée



- A Système en boucle fermée associé avec (G, Σ_c, Γ) et S est $L(S/G) \subseteq \Sigma^*$ tel que
  - $\varepsilon \in L(S/G)$ et
  - $sa \in L(S/G)$ ssi $s \in L(S/G)$, $a \in S(s)$,et $sa \in L(G)$, .

Example

- $S(\varepsilon) = S(ab) = S((ab)^*) = \{a, b, d\}$
- $S(a) = S(aba) = S(a(ba)^*) = \{a, b, c, d\}$
- for all other strings w, $S(w) = \{a, b, d\}$
- $L(S/G) = \{\varepsilon, a, ab, aba, abab, \ldots\}$

Problème de contrôle supervisé

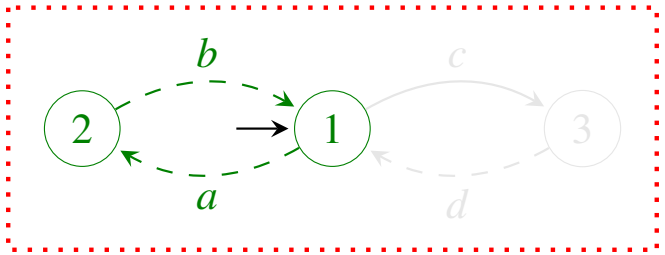
Given

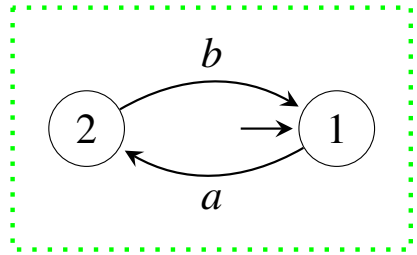
- générateur contrôlé (G, Σ_c, Γ)
- spécification $K \subseteq L(G)$

Y a t-il un superviseur $S : L(G) \to \Gamma$ t.q. $L(S/G) = K$?

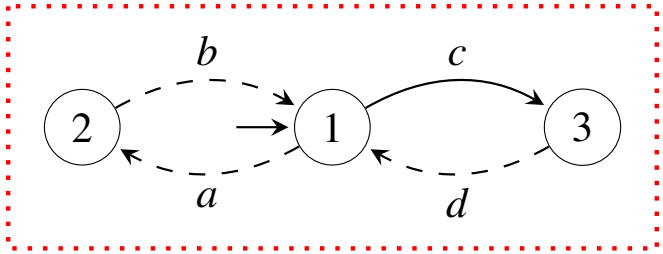
Exemple.

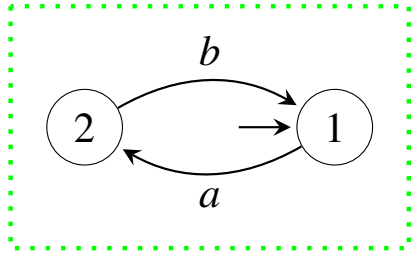
- $K = \{\varepsilon, a, ab, aba, abab, \ldots\}$
- $S((ab)^*) = \{a, b, d\}$, $S(a(ba)^*) = \{a, b, c, d\}$, $S(w) = \{a, b, d\}$
- $L(S/G) = K$





Contrôlabilité





Definition

Given

- générateur contrôlé (G, Σ_c, Γ)
- $K \subseteq L(G)$

Soit $\Sigma_u = \Sigma \setminus \Sigma_c$.
K est contrôlable pour $L(G)$ et Σ_u si

$$K\Sigma_u \cap L(G) \subseteq K$$

Example

- $\{\varepsilon, a, ab, aba, \ldots\}\{a, b, d\} \cap L(G) \subseteq \{\varepsilon, a, ab, aba, \ldots\}$

Théorème de contrôlabilité

Theorem (Ramadge & Wonham, 1987)

Soit

- (G, Σ_c, Γ) *un générateur contrôlé , et*
- $K \subseteq L(G)$ *une spécification.*

Il existe un superviseur $S : L(G) \to \Gamma$ *t.q.*

$$L(S/G) = K$$

ssi

spécification K *est contrôlable*

pour $L(G)$ *et* Σ_u.

Observations partielles : propriétés

$P : \Sigma^* \to \Sigma_o^*$, où $\Sigma_o \subseteq \Sigma$ ... événements observables
Exemple. $P : \{a, b, \ldots, y, z\}^* \to \{e, h, l, o\}^*$

$$P(abhdeadiladilasoaf) = hello$$

- un générateur contrôlé (G, Σ_c, Γ)
- spécification $K \subseteq L(G)$

K est *observable pour* $L(G)$ *et* P *si pour tout* $s \in \text{prefix}(K)$ *et* $a \in \Sigma_c$ *:*

$$sa \in L(G), \quad s'a \in \text{prefix}(K), \quad P(s) = P(s') \Rightarrow sa \in \text{prefix}(K).$$

Superviseur S *sous obs. partielles* $P : \Sigma^* \to \Sigma_o^*$ *est* $S : P(L(G)) \to \Gamma$
Afin qu'il existe un superviseur S *avec* $L(S/G) = K$ *il faut que* K *soit contrôlable et observable.*

Normalité et Observabilité relative

Definition

K est *normal* pour L et $P : \Sigma^* \to \Sigma_o^*$ si $P^{-1}P(K) \cap L \subseteq K$.

Definition

K est *C-observable* pour L et Σ_o si
$\forall w \in K$ et $(\forall w' \in C)$ avec $P(w) = P(w')$ et $(\forall a \in \Sigma)$
$(wa \in K \wedge w'a \in L \Rightarrow w'a \in K)$.

Les sous langages suprêmes normaux et suprêmes relativement observables sublanguages toujours existent.

Langages admissibles

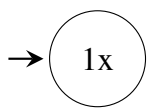
Propriétés des Langages de $L(S/G)$

1. $\sup \mathrm{C}(K, L)$, le sous-langage suprême contrôlable de K,
2. $\sup \mathrm{N}(K, L)$, le sous-langage suprême normal de K,
3. $\sup \mathrm{CN}(K, L)$, le sous-langage suprême contrôlable et normal de K,
4. $\sup \mathrm{RO}(K, L)$, le sous-langage suprême K-observable de K.

Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

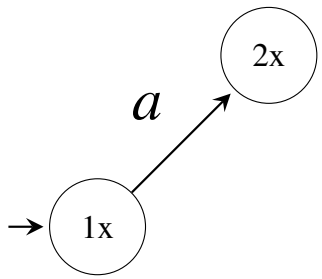
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

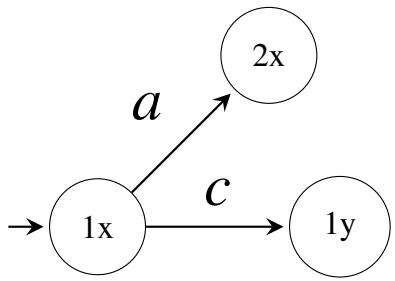
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

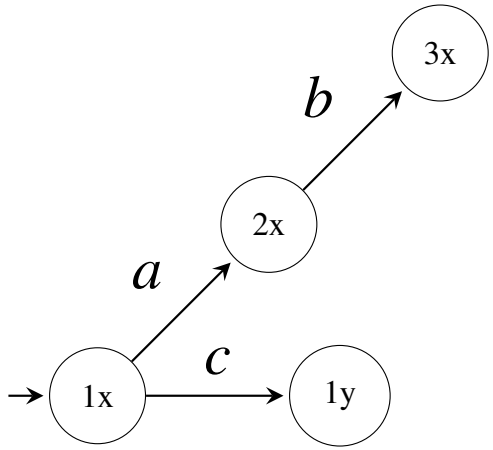
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

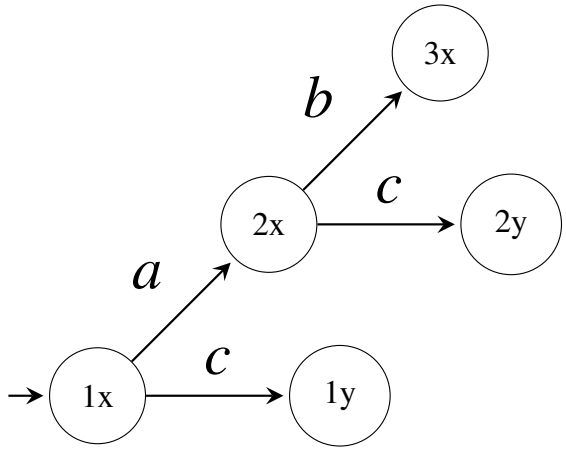
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

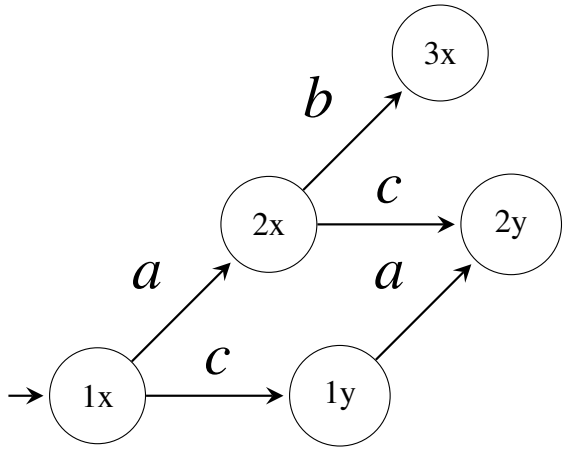
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

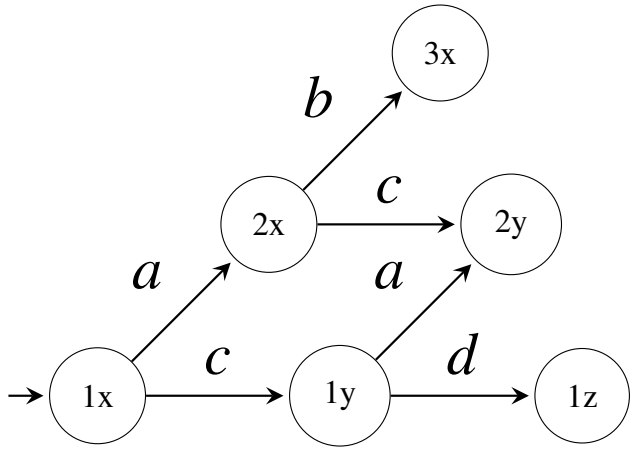
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

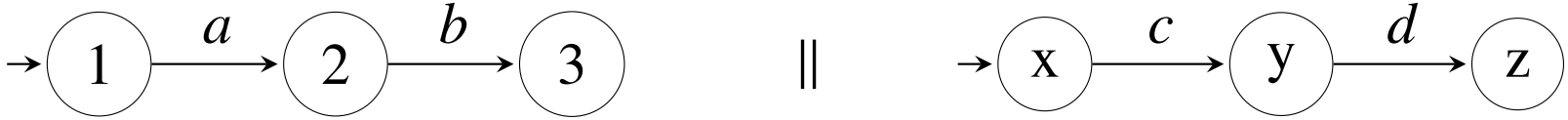
Example

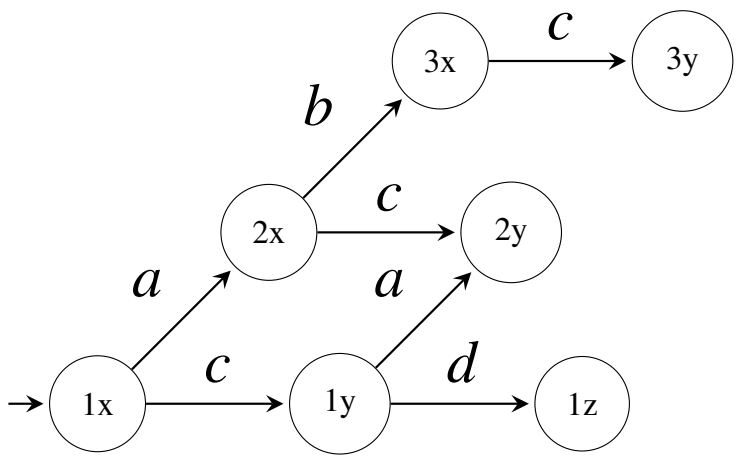


Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

Example

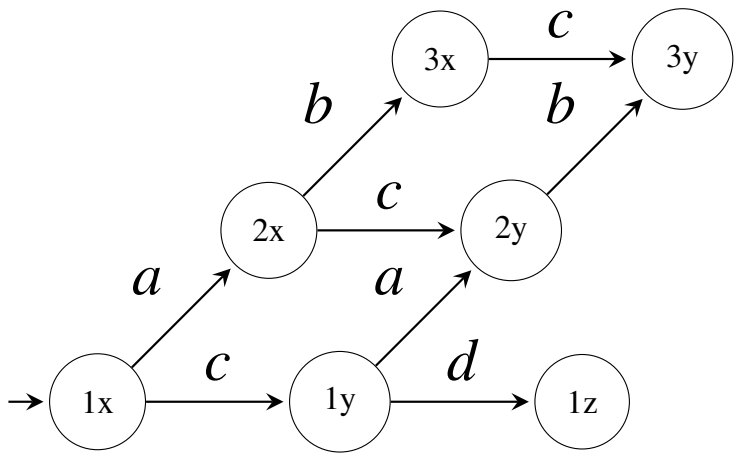




Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

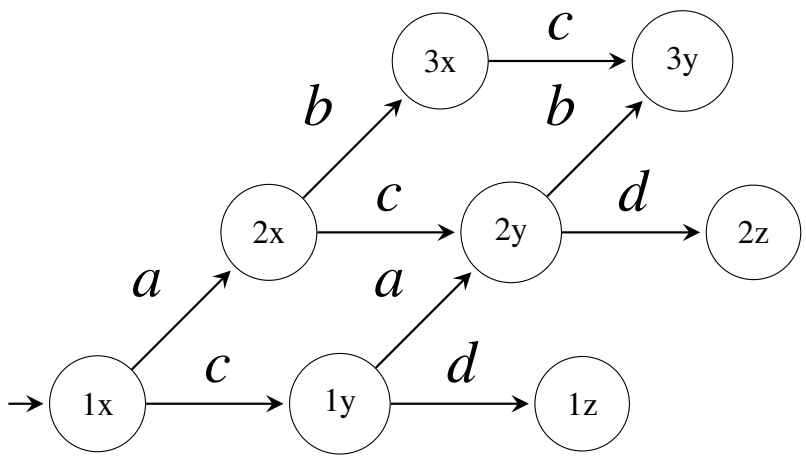
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

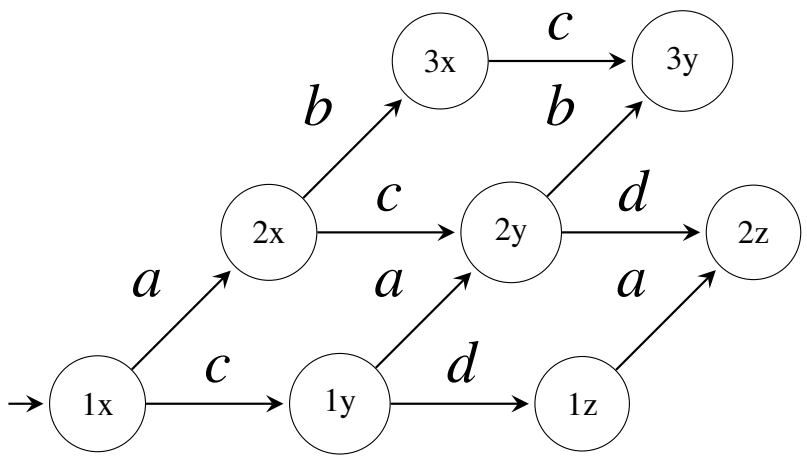
Example



Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

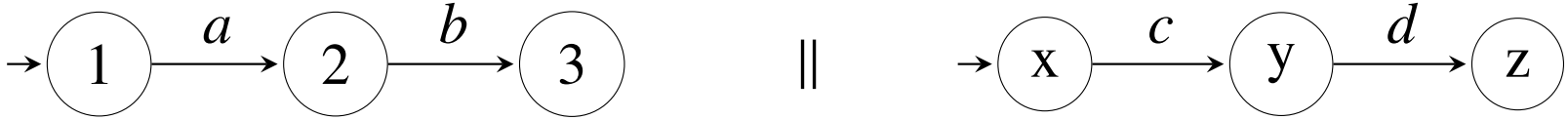
Example

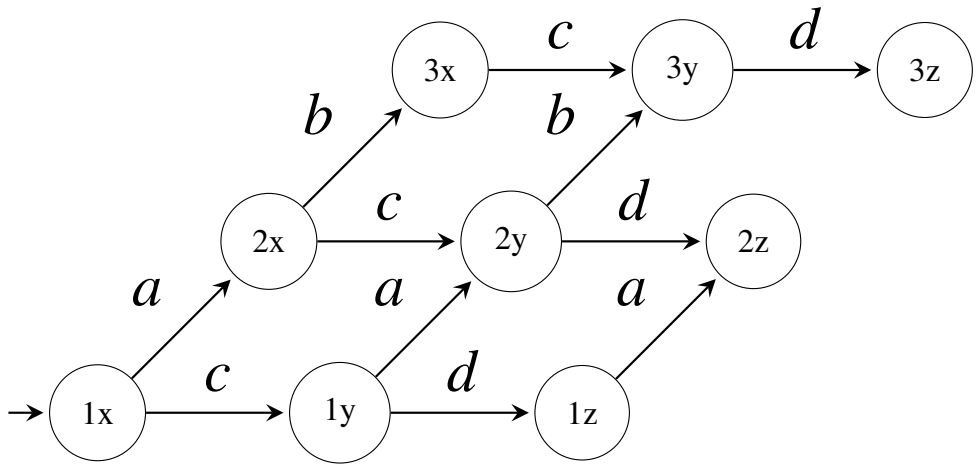


Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

Example

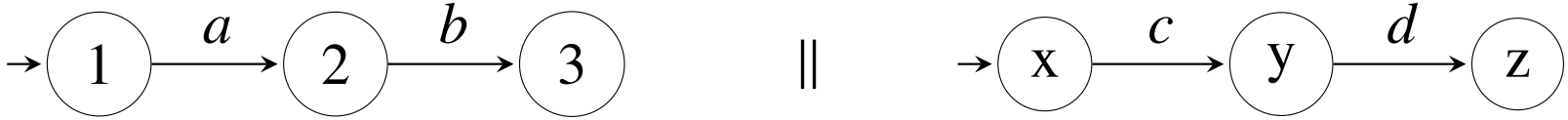


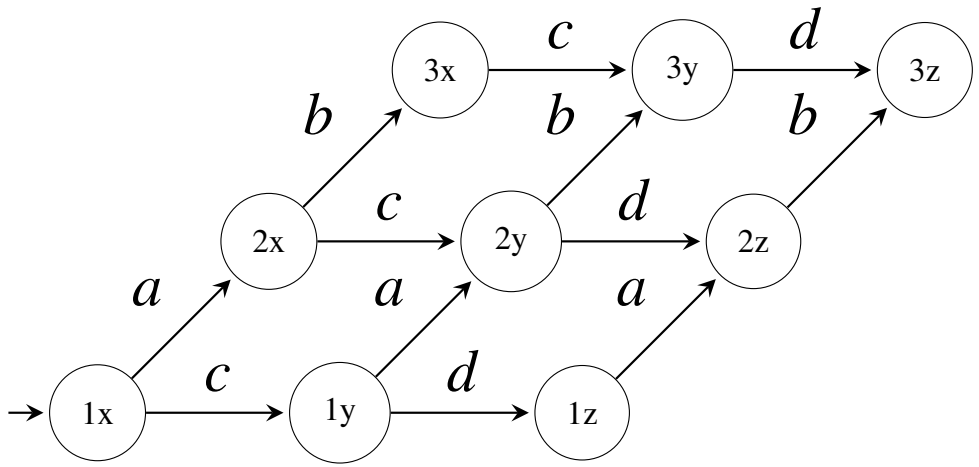


Systèmes de large taille

- Composés d'un grand nombre des (petites) composantes
- Composition parallèle (produit synchrone)

Example





Contrôle Modulaire

$$
\begin{aligned}
\mathbb{Z}_n &= \{1,2,\ldots,n\},\ \textbf{indice}.\\
G &= G_1 \parallel \cdots \parallel G_n,\ \text{produit synchrone},\\
&\quad G\ \textbf{système global}, G_1,\ldots,G_n,\ \textbf{systèmes locaux}.\\
\Sigma_i &\quad \textbf{alphabets locaux}, \text{pas nécessairement disjoints!}\\
\{L_i, i \in \mathbb{Z}_n\} &\quad \textbf{langages locaux}\ , L = \|_{i=1}^{n} L_i,\\
\Sigma_i &= \Sigma_{ic} \cup \Sigma_{iu},\ \text{une partition},\\
\Sigma_{ic} &\quad \textbf{événements localement contrôlables}\ ,\\
\Sigma_{iu} &\quad \textbf{événements localement incontrôlables},\\
E &= \cup_{i=1}^{n} \Sigma_i,\ \text{the } \textbf{alphabets global}, \Sigma_c = \cup_{i=1}^{n} \Sigma_{ic}.\\
P_i : \Sigma^* \to \Sigma_i^* &\quad \text{projections sur les alphabets locaux}, \forall i \in \mathbb{Z}_n.\\
&\quad P^{-1} : \mathrm{Pwr}(\Sigma_i^*) \to \mathrm{Pwr}(\Sigma^*),\ \text{projection inverse}.
\end{aligned}
$$

Littérature 1 sur la supervision modulaire

- P.J. Ramadge et W.M. Wonham (1988).
  nonconflicting languages et $\Sigma_i = \Sigma_j$ for $\forall i,j \in \mathbb{Z}_n$.
- Y. Willner et M. Heymann (1991).
  Tous les événements partagés sont contrôlables.
- K.C. Wong et S.H. Lee (2002). Generalisationà la condition,

$$\Sigma_{iu} \cap \Sigma_j = \Sigma_i \cap \Sigma_{ju}, \quad \forall i,j \in \mathbb{Z}_n.$$

  événements partagés ont le même statut de contrôlabilité
- B. Gaudin, H. Marchand (2004).
  Spécification non decomposables avec les événements partagés tous contrôlables.
- Approche basé sur le contrôle hierarchical (conditions dites "observer" et "OCC" , Lei Feng 2009, Schmidt 2010)

Littérature 2

Notions

Définition. Composantes G_i $i \in \mathbb{Z}_n$

sont d'accord sur contrôlabilité de leurs événements partagés si

$$\Sigma_{iu} \cap \Sigma_j = \Sigma_i \cap \Sigma_{ju}, \ \forall i,j \in \mathbb{Z}_n, \ i \neq j.$$

(K.C. Wong, S.H. Lee (EJC 2002)).

Proposition On a alors $\Sigma_u = \cup_{i=1}^{n} \Sigma_{iu}$.

Définition. Langage de Spécification . Consider $K \subseteq \Sigma^*$.

$$\begin{aligned} K_i &= K \cap P_i^{-1}(L_i), \ \{K_i \subseteq \Sigma^*, i \in \mathbb{Z}_n\}. \\ & \quad K_i \text{ sur-approximation locale de } K \cap L. \text{ On a} \\ K \cap L &= \cap_{i=1}^{n} K_i. \end{aligned}$$

(B. Gaudin, H. Marchand (WODES.2004)).

Notion de contrôlabilité mutuelle.

Def. Contrôlabilité mutuelle globale of $\{L_i \subseteq \Sigma_i^*, i \in \mathbb{Z}_n\}$ if

$$P_j^{-1}(L_j)\Sigma_{ju} \cap P_i^{-1}(L_i) \subseteq P_j^{-1}(L_j),\ \forall i,j \in \mathbb{Z}_n,\ i \neq j.$$

Remarques Interprétation
$P_j^{-1}(L_j)$ est contrôlable par rapport au $P_i^{-1}(L_i)$ et Σ_{ju}.
Modular controllability (MC) of $\{L_i \subseteq \Sigma^*, i \in \mathbb{Z}_n\}$ if

$$L\Sigma_u \cap P_i^{-1}(L_i) \subseteq L,\ \forall i \in \mathbb{Z}_n.$$

Remarks Interpretation L controllable with respect to $(P_i^{-1}(L_i), \Sigma_u)$.
High-computational complexity. Necessary condition.
Proposition For $n = 2$, GMC = MC. For $n \geq 3$, GMC $\Rightarrow$ MC.

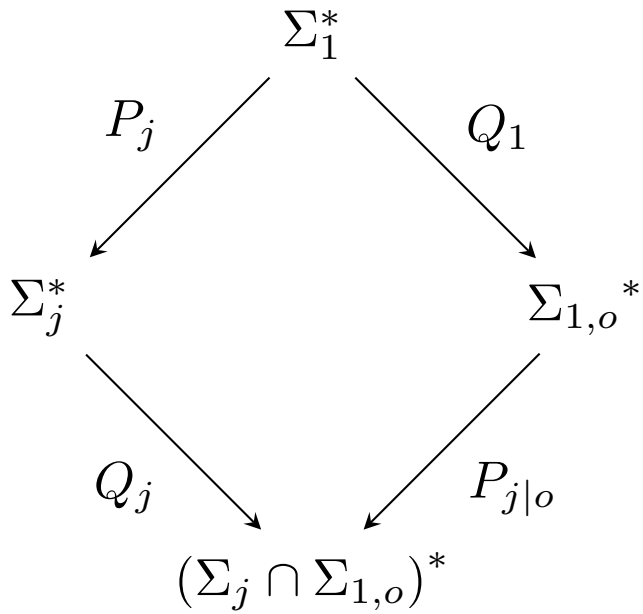
Complexité

Complexité temporelle for the computation of the suprême controllable sublanguage.

Modular		$O(n_m^2 (n^*)^3 n_K^3)$ polynomial,
Global		$O((n^*)^{3n_m} n_K^2)$, exponential ,
	n_m	nombre de modules,
	n_i	taile maximale de modules $i \in \mathbb{Z}_{n_m}$,
	n^*	$= \max_{i \in \mathbb{Z}_{n_m}} n_i$,
	n_K	taile minimale de l'automate de spécification $K \subseteq \Sigma^*$,
Modulaire		$Q(n_m (n^*)^3 n_K^2) + O(n_m^2 (n^*)^3 n_K^3) \leq Q(n_m^2 (n^*)^3 n_K^3)$,
		correspond au contrôle local et vérification de GMC.

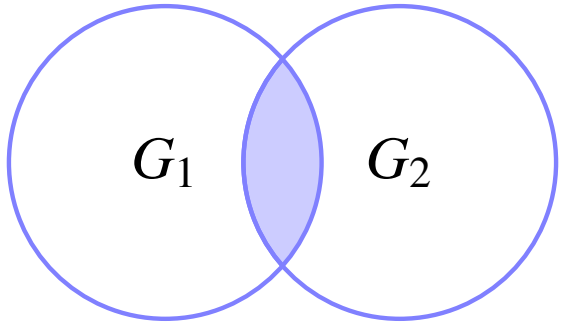
Contrôle modulaire sous observation partielle

Superviseurs locaux S_i contrôlent événement dans $\Sigma_{i,c} = \Sigma_i \cap \Sigma_c$ et observent événement dans $\Sigma_{i,o} = \Sigma_i \cap \Sigma_o$.
Notons $\Sigma_{i,u} = \Sigma_i \setminus \Sigma_{i,c}$.
Local observations are projections $Q_i : \Sigma_i^* \to \Sigma_{i,o}^*$.



Coordinateur

$$G = G_1 \parallel G_2 \parallel \ldots \parallel G_n$$



Composantes partagent une information commune

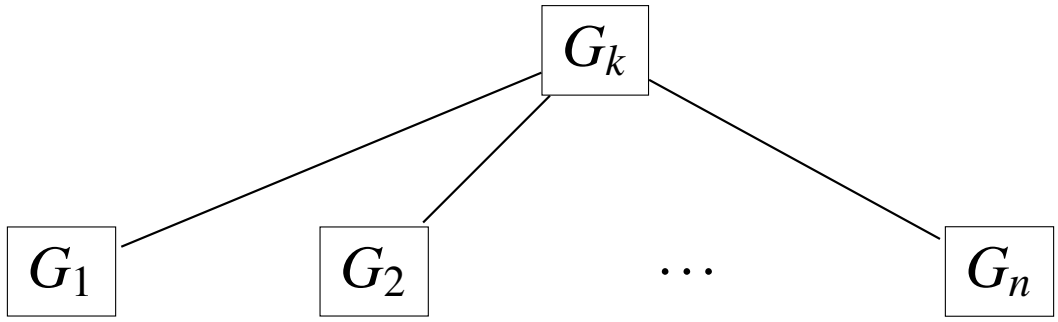
– événements partagés = communication

Abstraire l'information – coordinateur G_k

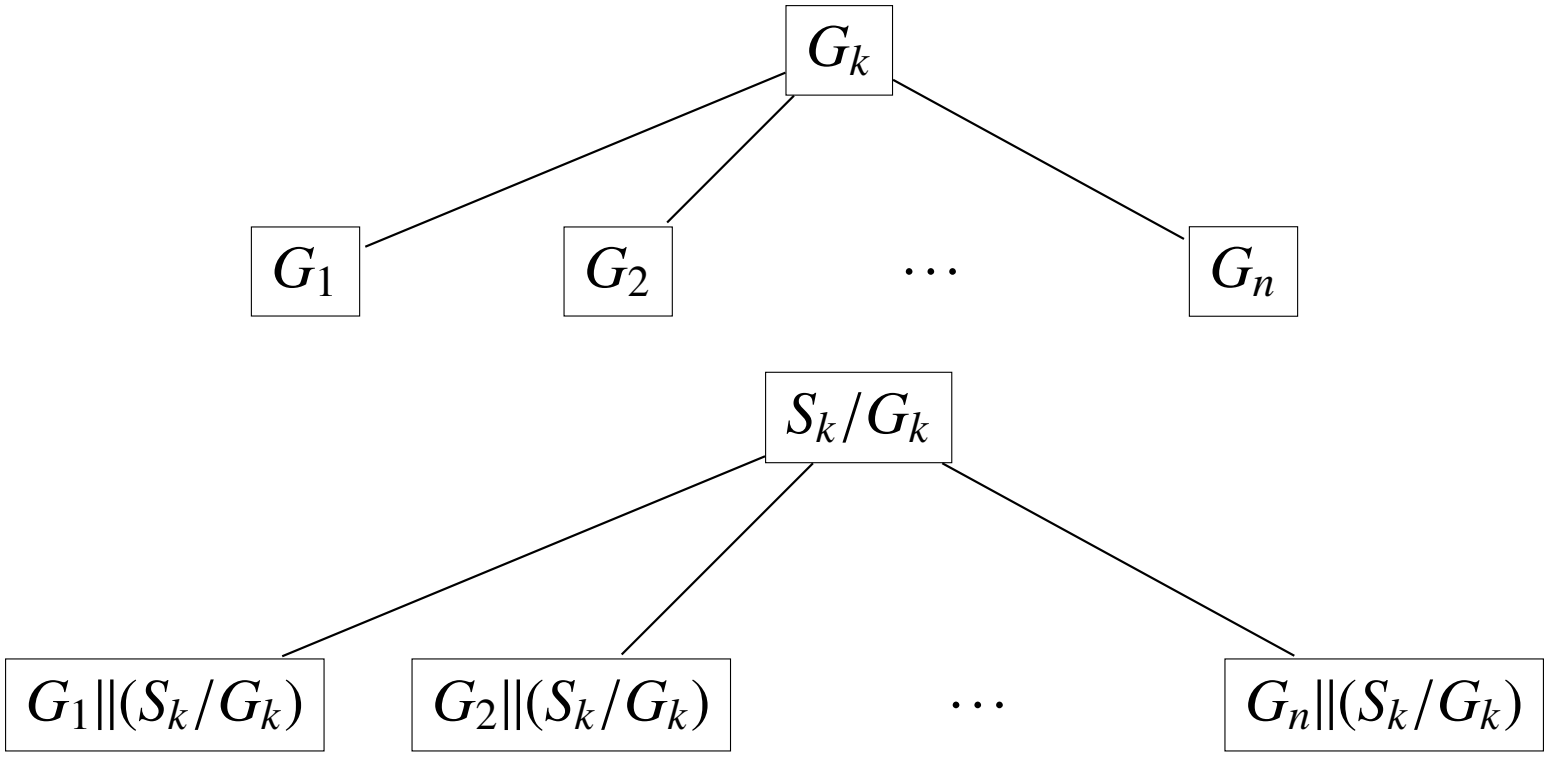
Le coordinateur partage cette information parmis les sous systèmes

$$G_1 \parallel G_2 \parallel \ldots \parallel G_n \parallel G_k$$

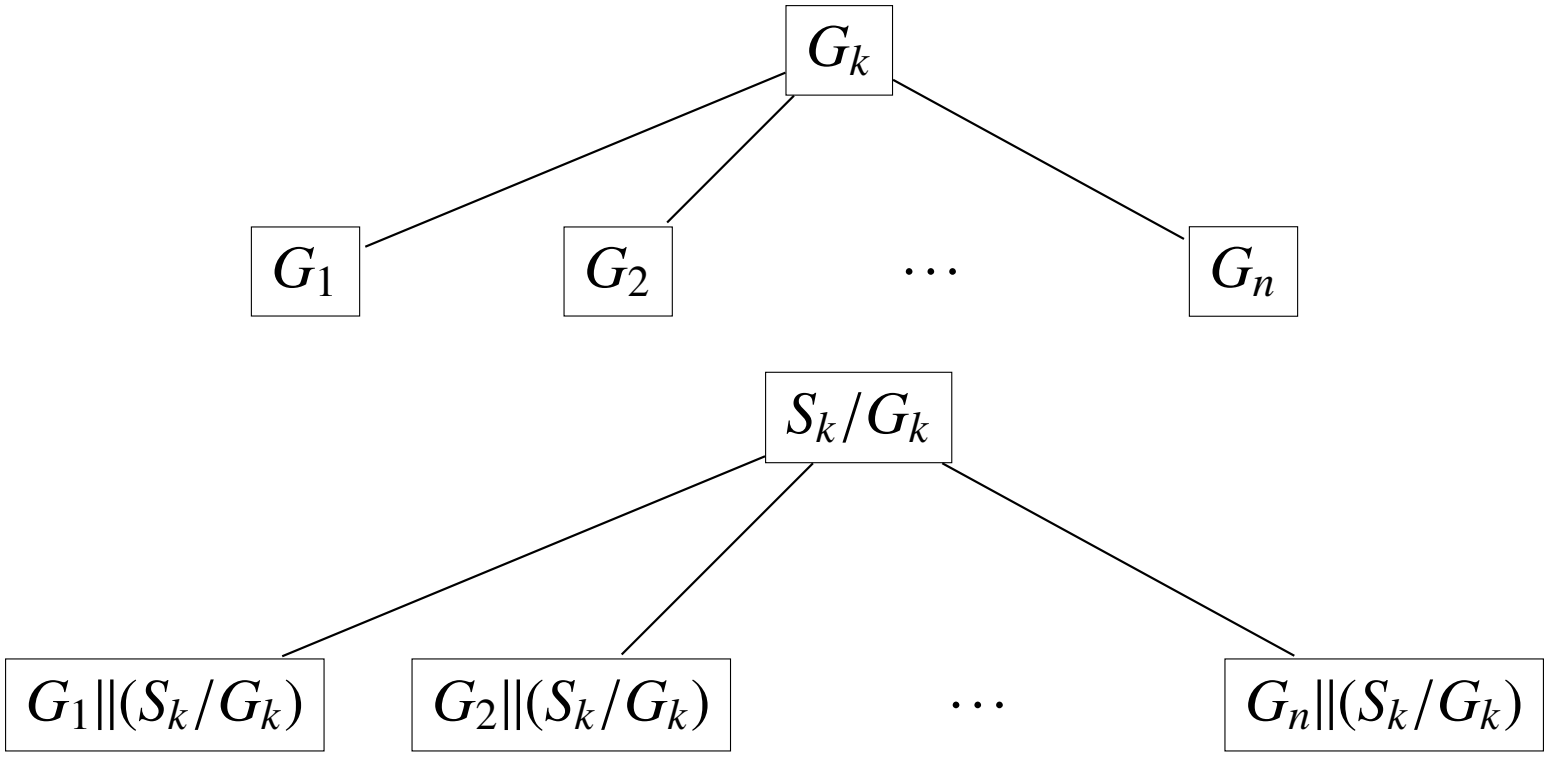
Coordination – idée de base



Coordination – idée de base



Coordination – idée de base



$$S_1/(G_1\|(S_k/G_k)) \parallel S_2/(G_2\|(S_k/G_k)) \parallel \ldots \parallel S_n/(G_n\|(S_k/G_k)) \equiv K$$

Problème de contrôle avec coordination

Problem

Given

- *Composantes G_1 et G_2 sur alphabets Σ_1 et Σ_2, resp.*
- *et une spécification $K \subseteq L(G_1 \parallel G_2)$*

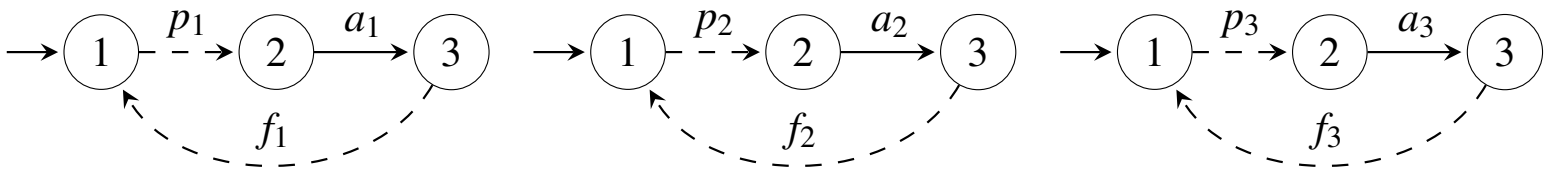
Construire

- *un coordinateur G_k sur Σ_k et*
- *superviseurs S_1, S_2, S_k t.q.*

$$L(S_1/[G_1 \parallel (S_k/G_k)]) \parallel L(S_2/[G_2 \parallel (S_k/G_k)]) = K?$$

Exemple simple

- $\Sigma_c = \{a_1, a_2, a_3\}$
- p_i = préparer
- a_i = accès (à la resource unique)
- f_i = finir



- Spécification: à chaque instant un seul système peut acceder à la résource

Décomposabilité Conditionelle

Definition

Langage $K \subseteq (\Sigma_1 \cup \Sigma_2)^*$ est conditionellement décomposable pour les alphabets $\Sigma_1, \Sigma_2, \Sigma_k \subseteq \Sigma_1 \cup \Sigma_2$ si

$$K = P_{1+k}(K) \parallel P_{2+k}(K)$$

$$P_{1+k} : (\Sigma_1 \cup \Sigma_2)^* \rightarrow (\Sigma_1 \cup \Sigma_k)^*$$

Un tel Σ_k toujours existe!

Construction des coordinateurs

Given

- Composantes G_1 et G_2 sur Σ_1 et Σ_2, resp. et
- une spécification K

Construire un coordinateur G_k sur Σ_k comme suit:

1. Init Σ_k est l'ensemble des événements partagés
2. Étend Σ_k t.g. K devient conditionellement décomposable

Sous langages suprêmes sublanguages – calcul

Theorem (JK, TM, JHvS, 2010)

Given

- G_1, G_2, G_k, *et*
- K *conditionellement décomposable*

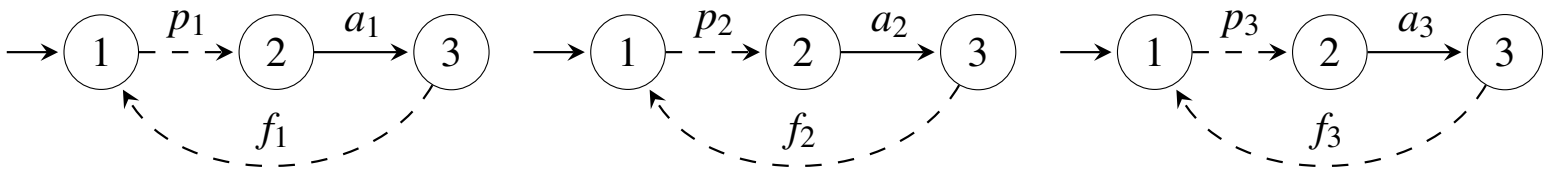
On Défini

$$\sup \mathrm{C}_{1+k} = \sup \mathrm{C}(P_{1+k}(K), L(G_1) \| \sup \mathrm{C}_k)$$
$$\sup \mathrm{C}_{2+k} = \sup \mathrm{C}(P_{2+k}(K), L(G_2) \| \sup \mathrm{C}_k)$$

Si P_k^{i+k} *sont* $(P_i^{i+k})^{-1}(L_i)$*-observeur et OCC pour* $(P_i^{i+k})^{-1}(L_i)$*, alors*

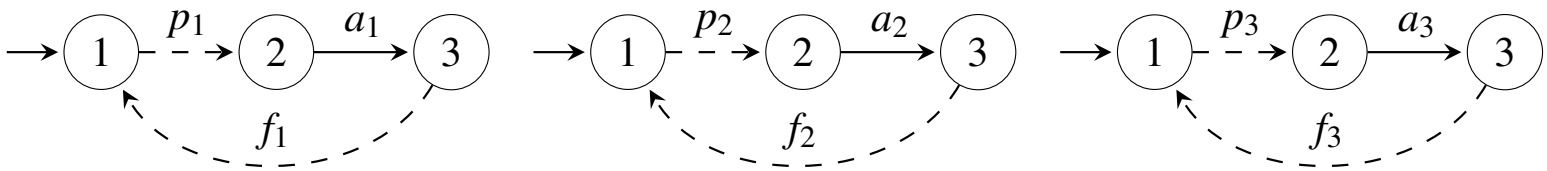
$$\sup \mathrm{C}_{1+k} \| \sup \mathrm{C}_{2+k} = \sup \mathrm{C}(K, L, E)\,.$$

Exemple Simple – coordinateur



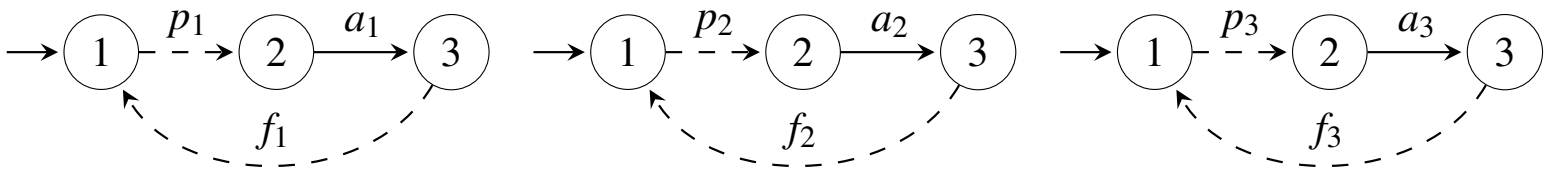
- $\Sigma_k = (\Sigma_1 \cap \Sigma_2) \cup (\Sigma_1 \cap \Sigma_3) \cup (\Sigma_2 \cap \Sigma_3) = \emptyset$

Exemple Simple – coordinateur



- $\Sigma_k = (\Sigma_1 \cap \Sigma_2) \cup (\Sigma_1 \cap \Sigma_3) \cup (\Sigma_2 \cap \Sigma_3) = \emptyset$
- Éxtend $\Sigma_k = \{a_1, a_2, a_3\}$

Exemple Simple – coordinateur



- $\Sigma_k = (\Sigma_1 \cap \Sigma_2) \cup (\Sigma_1 \cap \Sigma_3) \cup (\Sigma_2 \cap \Sigma_3) = \emptyset$
- Éxtend $\Sigma_k = \{a_1, a_2, a_3\}$
- Construire $G_k = P_k(G_1) \parallel P_k(G_2) \parallel P_k(G_3)$

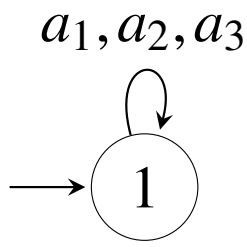


Figure: Coordinateur G_k – communication de l'accès à la if resource

Exemple simple– superviseurs

On construit les superviseurs

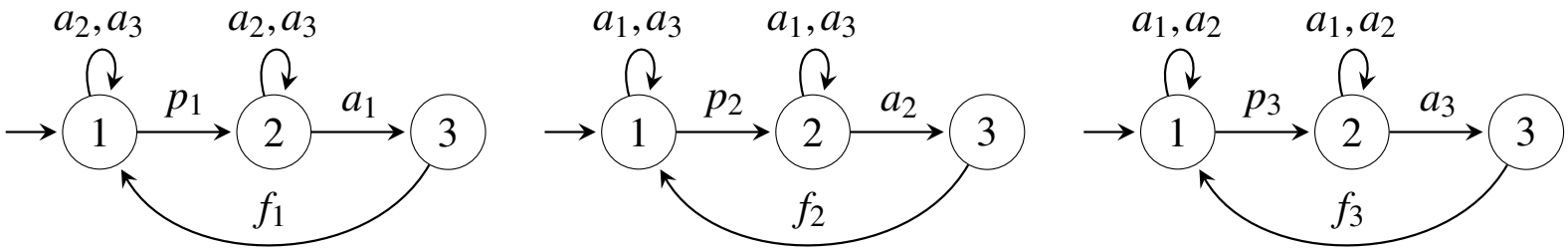


Figure: Superviseurs $\sup \mathrm{C}_{1+k}$, $\sup \mathrm{C}_{2+k}$, et $\sup \mathrm{C}_{3+k}$

- Les superviseurs communiquent via le coordinateur

Exemple simple– superviseurs

On construit les superviseurs

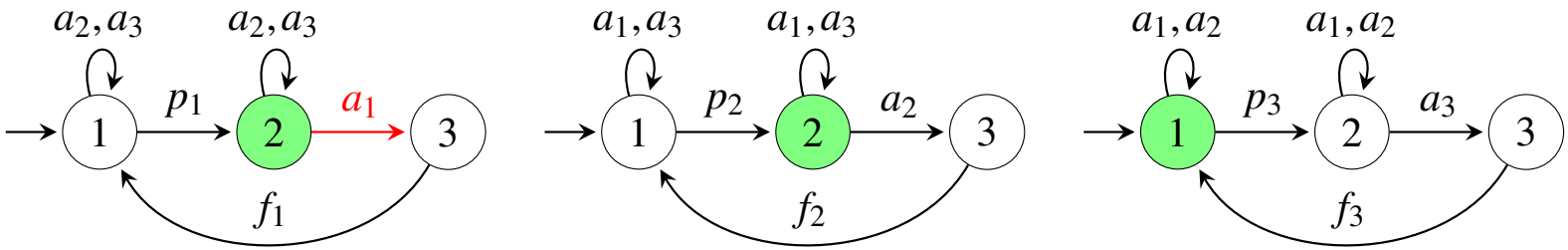


Figure: Superviseurs $\sup \mathrm{C}_{1+k}$, $\sup \mathrm{C}_{2+k}$, et $\sup \mathrm{C}_{3+k}$

- Les superviseurs communiquent via le coordinateur

Exemple simple– superviseurs

On construit les superviseurs

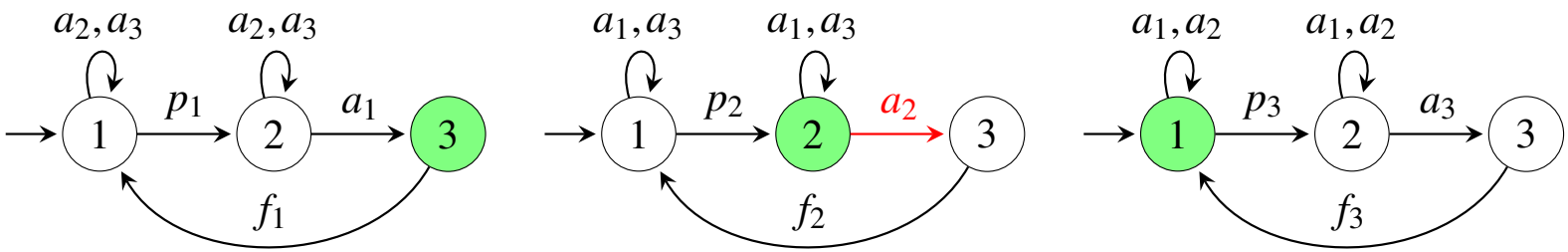


Figure: Superviseurs $\sup \mathrm{C}_{1+k}$, $\sup \mathrm{C}_{2+k}$, et $\sup \mathrm{C}_{3+k}$

- Les superviseurs communiquent via le coordinateur

Analyse de Complexité

- $L(G_i)$ avec m_i états
- $L(G_k)$ avec m_k états
- $P_{i+k}(K)$ avec n_i états
- $P_k(K)$ avec n_k états

Complexit du calcul des superviseurs :

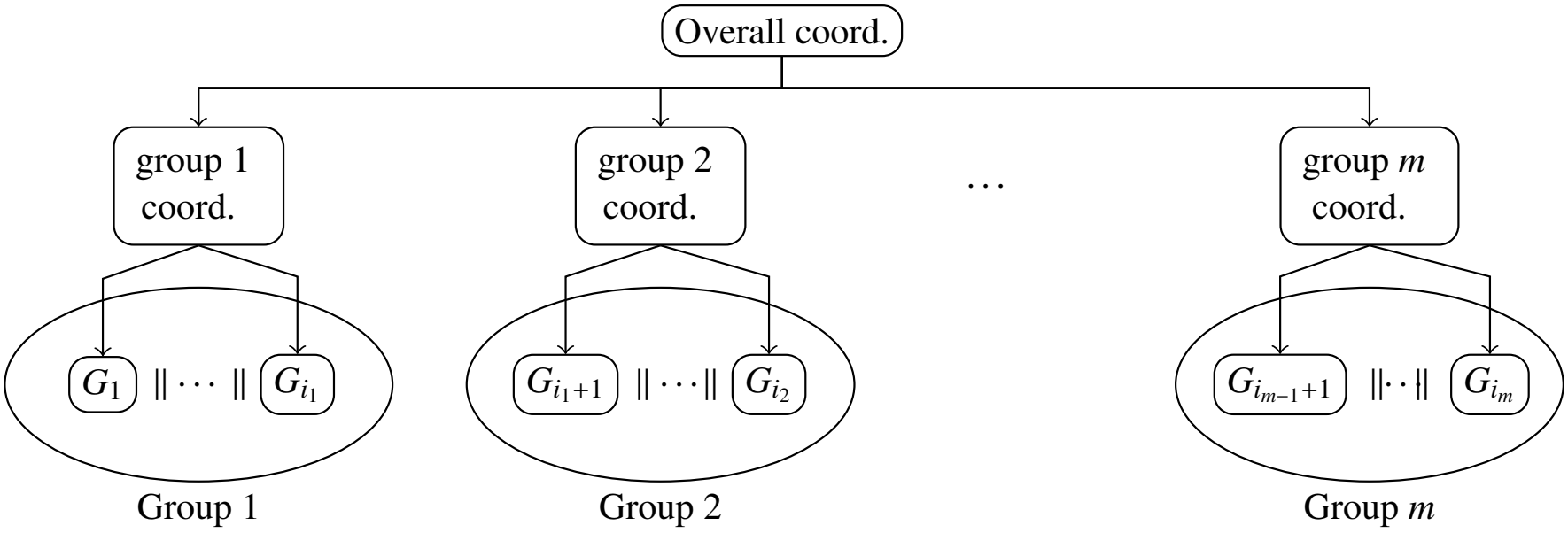
$$Q(m_k n_k + \sum_{i=1}^{n} m_i n_i m_k n_k)$$

vs.

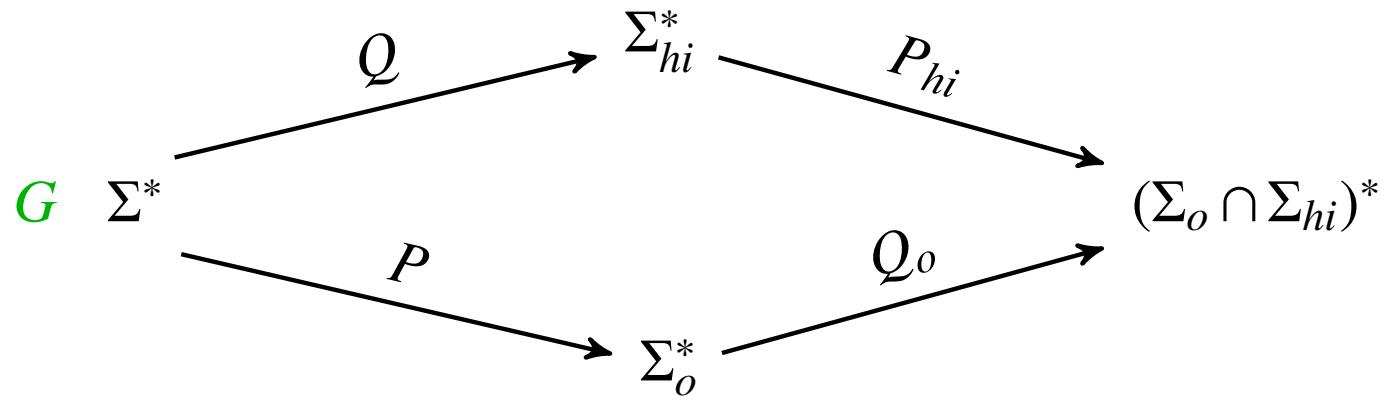
$$Q(m_k n_k \prod_{i=1}^{n} m_i n_i)$$

dans le cas monolithique.

Hierarchie à plusieurs niveaux

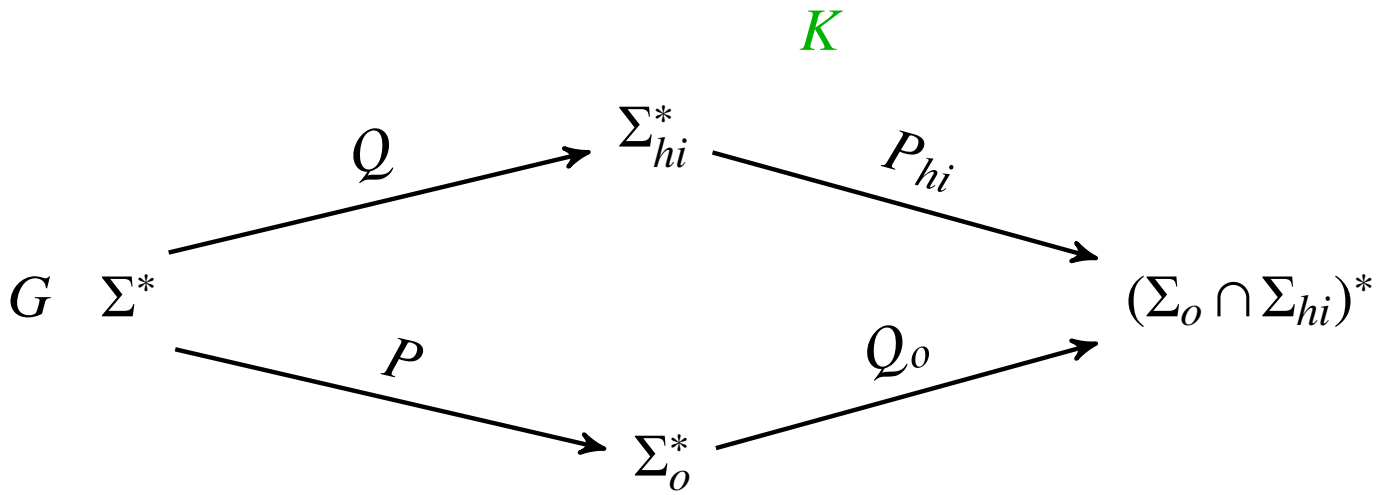


Commande supervisée hiérarchique



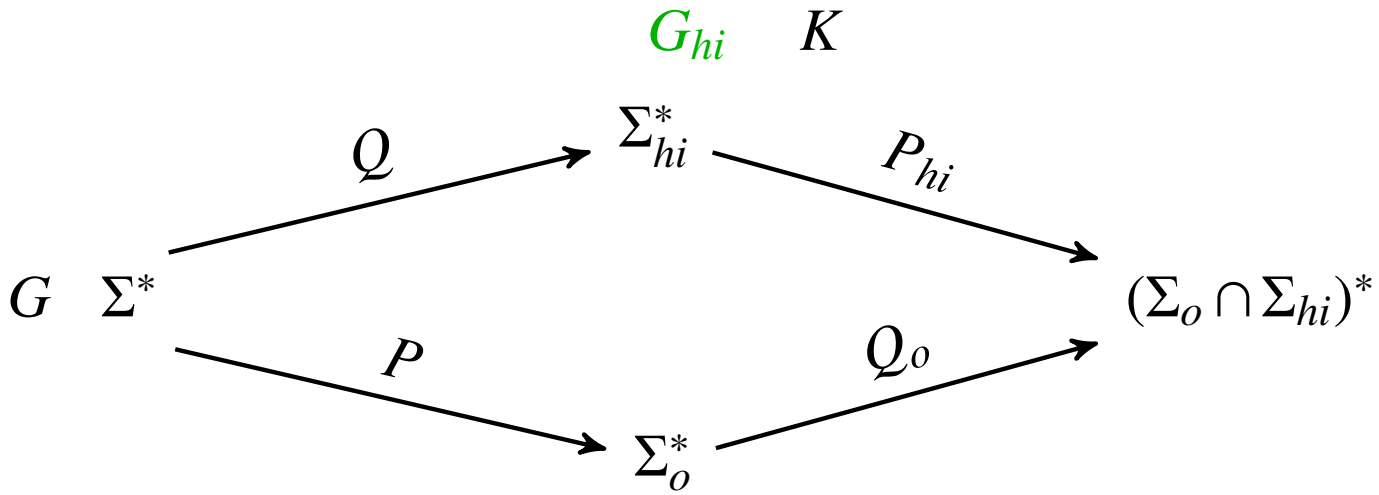
- Un système donnée (niveau bas) G sur Σ

Commande supervisée hiérarchique



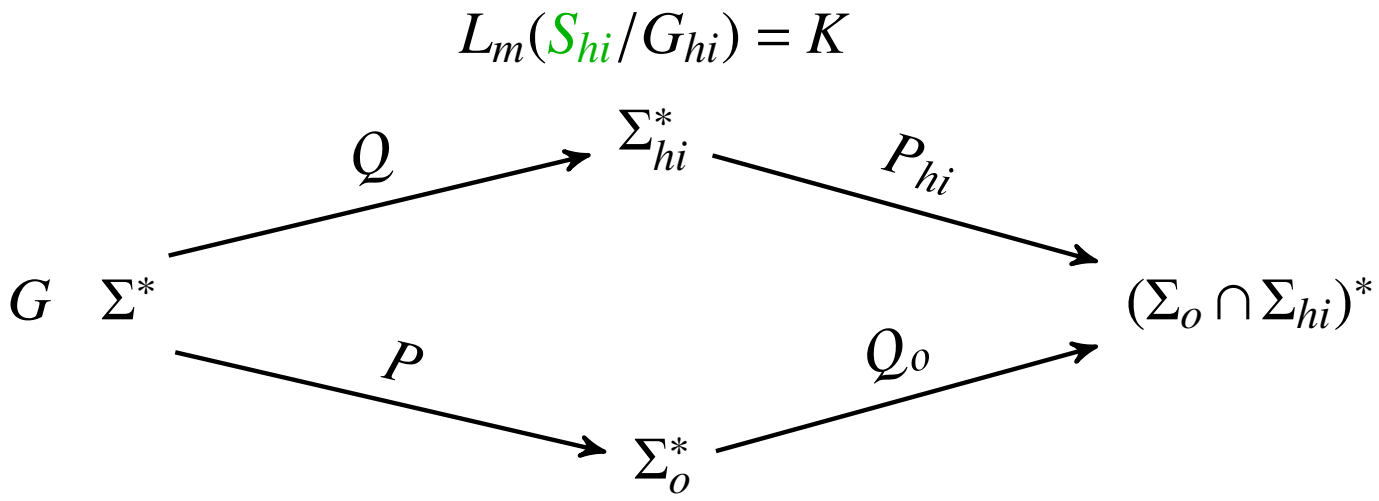
- Un système donnée (niveau bas) G sur Σ
  et une spécification K du haut niveau sur $\Sigma_{hi} \subseteq \Sigma$

Commande supervisée hiérarchique



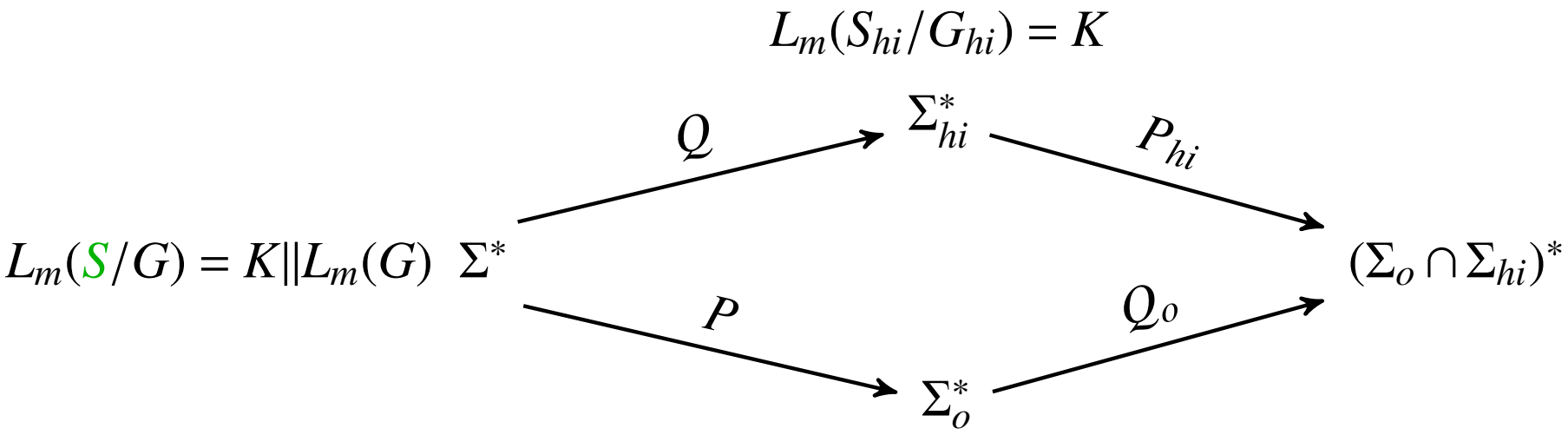
- Un système donnée (niveau bas) G sur Σ
  et une spécification K du haut niveau sur $\Sigma_{hi} \subseteq \Sigma$
- On abstrait G à G_{hi} sur Σ_{hi}

Commande supervisée hiérarchique



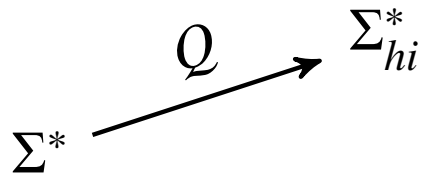
- Un système donnée (niveau bas) G sur Σ
  et une spécification K du haut niveau sur $\Sigma_{hi} \subseteq \Sigma$
- On abstrait G à G_{hi} sur Σ_{hi}
- On calcule un superviseur nonbloquant et le moins restreignant (maximallement permissive) S_{hi}

Commande supervisée hiérarchique



- Un système donnée (niveau bas) G sur Σ
  et une spécification K du haut niveau sur $\Sigma_{hi} \subseteq \Sigma$
- On abstrait G à G_{hi} sur Σ_{hi}
- On calcule un superviseur nonbloquant et le moins restreignant (maximallement permissive) S_{hi}
- On implémente S_{hi} et construit un superviseur nonbloquant et le moins restraignant S pour le système initial

Le cas des observations complètes



Theorem

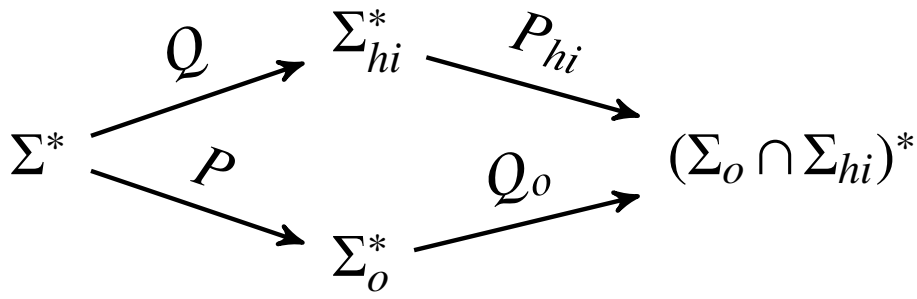
Pour un système G et une spécification $K \subseteq \Sigma^*_{hi}$,

$$\textit{si } Q : \Sigma^* \to \Sigma^*_{hi} \textit{ est } L_m(G)\textit{-observeur et OCC ou LCC}$$

alors K est contrôlable ssi $K \parallel L_m(G) \subseteq \Sigma^*$ est controllable.

- OCC = output control consistent (Zhong et Wonham, 1990)
- LCC = local control consistent (Schmidt et Breindl, 2011)
  - OCC implique LCC

Les notions de consistence observationnelle clées



$L \subseteq \Sigma^*$ est observation consistent (OC) pour Q, P, et P_{hi} si

- pour tous $t, t' \in Q(L)$ t.q. $P_{hi}(t) = P_{hi}(t')$,
- ils existent $s, s' \in L$ t.q.
  - $Q(s) = t$,
  - $Q(s') = t'$, et
  - $P(s) = P(s')$.

$L \subseteq \Sigma^*$ est locally observation consistent (LOC) pour Q et P si

- pour tout $s, s' \in L$ et $e \in \Sigma_c \cap \Sigma_{hi}$ t.q.
  - $Q(s)e \in Q(L)$,
  - $Q(s')e \in Q(L)$, et
  - $P(s) = P(s')$
- ils existent $u, u' \in (\Sigma \setminus \Sigma_{hi})^*$ t.q.
  - $P(u) = P(u')$,
  - $sue \in L$, et $s'u'e \in L$.

Résultats précedents

Theorem (CDC 2011)

Soit $K \subseteq Q(L_m(G))$ une spécification du haut niveau.
(1) Si $L(G)$ est OC pour Q, P, et P_{hi}, et
K et $L_m(G)$ sont nonconflicting, et $L(G)$ est LOC pour Q, P, et Σ_c, alors
K est observable pour $Q(L(G))$, $\Sigma_{hi} \cap \Sigma_o$, et $\Sigma_{hi} \cap \Sigma_c$ ssi $K||L_m(G)$ est observable pour $L(G)$, Σ_o, et Σ_c.

(2) Si Q est un $L_m(G)$-observeur et LCC pour $L(G)$, et
$L(G)$ est OC pour Q, P, et P_{hi}, et LOC pour Q, P, et Σ_c, alors

K est contrôlable pour $Q(L(G))$ et $\Sigma_{uc} \cap \Sigma_{hi}$, et observable pour $Q(L(G))$, $\Sigma_o \cap \Sigma_{hi}$, et $\Sigma_c \cap \Sigma_{hi}$ ssi $K||L_m(G)$ est contrôlable pour $L(G)$ et Σ_{uc}, et observable pour $L(G)$, Σ_o, et Σ_c.

(3) Si $L(G)$ est OC pour Q, P, et P_{hi}, et K et $L_m(G)$ sont nonconflicting, alors
K est normal pour $Q(L(G))$ et P_{hi} ssi
$K||L_m(G)$ est normal pour $L(G)$ et P.

Préservation de permissivité maximale

- OC et LOC suffissent pour préserver observabilité:
  - K est observable ssi $K \parallel L_m(G)$ est observable

- Et si K n'est pas observable?
  - sous langage normal suprémal
  - sous langage relativement observable suprémal
  - Quand est ce que ces langages sont préservés?

Problèmes ouverts

- OC et LOC, sont elles decidables?
- Préservent-elles permissivité maximale?

Problèmes ouverts

- OC et LOC, sont elles decidables?
  – sous certaines conditions elles sont satisfaites par construction pour SED modulaires!

- Préservent-elles permissivité maximale? – Non,
  mais nous proposons la modification de OC (MOC) et MOC preserve sup N

Problème de permissivité maximale du contrôle hiérarchique sous observations partielles

- Y-a t'il un nonblocking supervisor S_{hi} t.q.

$$L_m(S_{hi}/G_{hi}) \subseteq K$$

ssi il y a S t.q.

$$L_m(S/G) \subseteq K \parallel L_m(G)$$

Problème de permissivité maximale du contrôle hiérarchique sous observations partielles

- Y-a t'il un nonblocking supervisor S_{hi} t.q.

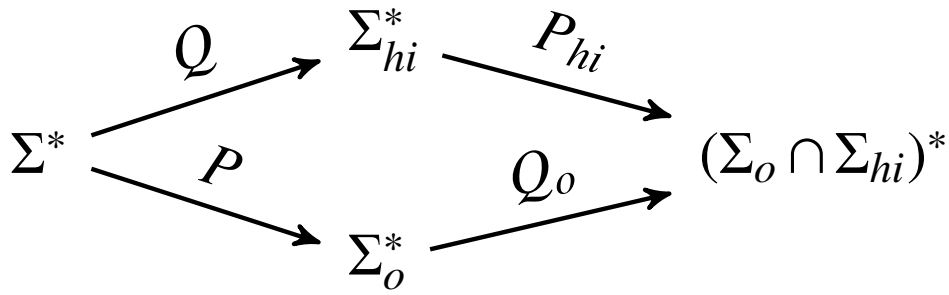
$$L_m(S_{hi}/G_{hi}) \subseteq K$$

ssi il y a S t.q.

$$L_m(S/G) \subseteq K \parallel L_m(G)$$

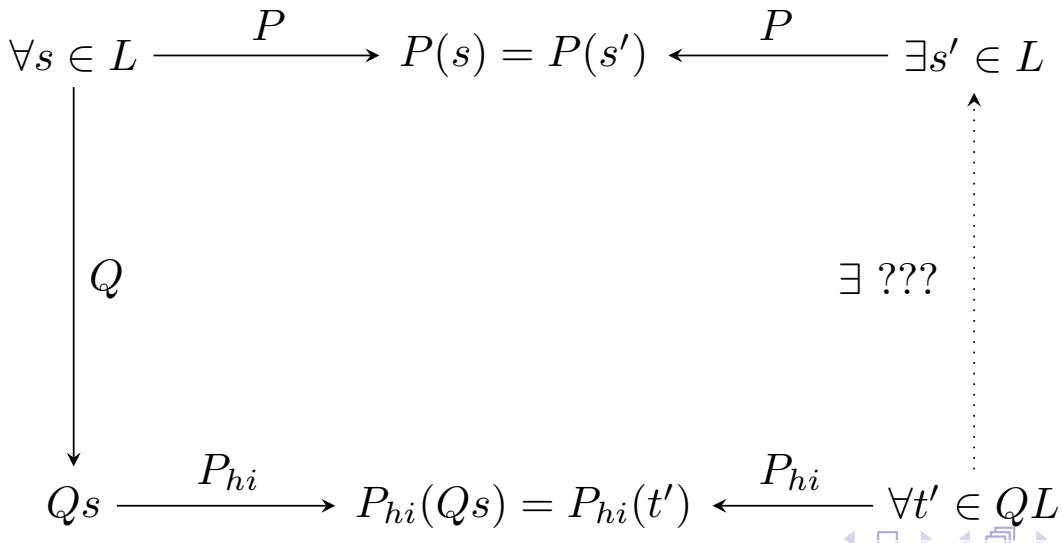
- OC et LOC ne suffisent pas pour préserver la permissivité maximale

Permissivité maximale



Langage $L \subseteq \Sigma^*$ est modified observation consistent (MOC) pour Q, P, et P_{hi} si

- pour tout $s \in L$ et tout $t' \in Q(L)$ t.q. $P_{hi}(Q(s)) = P_{hi}(t')$
- il existe $s' \in L$ avec
  - $P(s) = P(s')$ et $Q(s') = t'$.



Normalité

MOC préserve souslangages suprêmes normaux.

Lemma

(Normalité préservé par abstraction) Soit un DFA G avec $L = L(G)$. Si L est MOC pour Q, P, et P_{hi}, alors normalité de $S \subseteq L$ pour L et P implique la normalité de $Q(S)$ pour $Q(L)$ et P_{hi}.

Theorem

Soit G un DFA et $K \subseteq \Sigma_{hi}^*$ une spécification. Si $L(G)$ est MOC, alors

$$\sup \mathrm{N}\Big(K \parallel L_m(G), L(G)\Big) = \sup \mathrm{N}(K, Q(L(G))) \parallel L_m(G)$$

quand $\sup \mathrm{N}(K, Q(L(G)))$ *et* $L_m(G)$ *sont "nonconflicting".*

L'idée de la preuve. Il suffit d'appliquer le Lemme sur préservation de la normalité , car $\sup \mathrm{N}(K \| L, L) \subseteq Q^{-1}(\sup \mathrm{N}(K, Q(L)))$ ssi $Q(\sup \mathrm{N}(K \| L, L)) \subseteq \sup \mathrm{N}(K, Q(L))$.

Cas spéciaux

Deux cas spéciaux sont consideré dans la littérature:

- $\Sigma_o \subseteq \Sigma_{hi}$
- $\Sigma_{hi} \subseteq \Sigma_o$

Cas spéciaux

Deux cas spéciaux sont consideré dans la littérature:

- $\Sigma_o \subseteq \Sigma_{hi}$
- $\Sigma_{hi} \subseteq \Sigma_o$

Les deux inclusions impliquent MOC, et donc aussi OC.

Observabilité Relative

Theorem

Soit G un Aut. Fini et K une spécification.

- *Si L(G) est MOC et LOC et*
- *K et $L_m(G)$ sont nonconflicting*

alors

$$\sup \mathrm{RO}\Big(K \parallel L_m(G), L(G)\Big) \subseteq \sup \mathrm{RO}(K, Q(L(G))) \parallel L_m(G)\,.$$

Observabilité Relative

Theorem

Soit G un Aut. Fini et K une spécification.

- *Si L(G) est MOC et LOC et*
- *K et $L_m(G)$ sont nonconflicting*

alors

$$\sup \mathrm{RO}\Big(K \parallel L_m(G), L(G)\Big) \subseteq \sup \mathrm{RO}(K, Q(L(G))) \parallel L_m(G)\,.$$

- L'inclusion opposite n'est pas valide en général!
  - La solution du haut-niveau peut être meilleur!
  - Si les langages à droite sont "nonconflicting", alors le langage à droite est observable
    (mais pas forcément relativement observable)

Projections

Under which condition:
$\|_{i=1}^{n} L(S_i/G_i) = L(S/\|_{i=1}^{n} G_i)$?

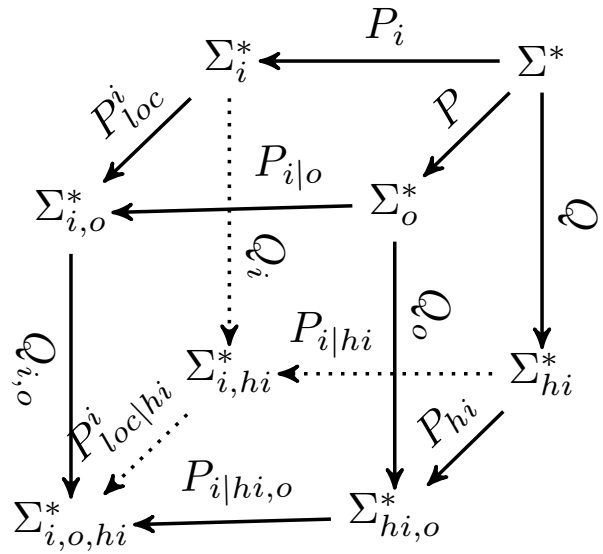


Figure: Notation pour les projections.

Cas $Q = P_i$: projections locales comme abstractions nous aident!

Calcul modulaire des langages $\sup \mathrm{N}$

Le calcul de $\sup \mathrm{N}$ pour les systèmes modulaires est double exponentiel (exp. en nombre des états qui est exp. en nombre des composants)!
$L_1 \parallel \cdots \parallel L_n = L(G_1 \parallel \cdots \parallel G_n)$, et une K donnée soit comme composition des specifications locales $K_i \subseteq L_i$, ou comme une spécification globale $K \subseteq \|_{i=1}^{n} L_i$.

Theorem

Pour $i = 1, \ldots, n$, $K = \|_{i=1}^{n} K_i$, *et* $\sup \mathrm{N}_i = \sup \mathrm{N}(K_i, L_i, P^i_{loc})$.
(1) Si $\sup \mathrm{N}_i$ *sont nonconflicting, alors*

$$\sup \mathrm{N}(K, L, P) \subseteq \|_{i=1}^{n} \sup \mathrm{N}_i .$$

(2)Si $L = \|_{i=1}^{n} L_i$ *est MOC pour* P_i, P, *et* P^i_{loc}, *et* $P_i(L) = L_i$, *alors*

$$\sup \mathrm{N}(K, L, P) = \|_{i=1}^{n} \sup \mathrm{N}_i .$$

Preuve.

$P_i(\sup \mathrm{N}(K,L,P)) \subseteq \sup \mathrm{N}_i$.
Ceci découle de normalité de $P_i(\sup \mathrm{N}(K,L,P))$ pour L_i et P^i_{loc}.
Il suffit d'appliquer le Lemme sur préservation de normalité (grâce à MOC) , où P_i jouent le rôle de l' abstraction Q et P^i_{loc} sont observations locales. □

Rémarque.
1. On a prouvé que si tous les événements partagés sont observables, alors $L = \|_{i=1}^{n} L_i$ est MOC pour P_i, P, et P^i_{loc}
2. $P_i(L) = L_i$ si on remplace L_i par $\tilde{L}_i = L_i \cap \cap_{j \neq i} P_i(L_j)$, et on a toujours $L = \|_{i=1}^{n} \tilde{L}_i$.

$$\begin{array}{ccccc} \forall s \in L & \xrightarrow{P} & P(s) = P(s') & \xleftarrow{P} & \exists s' \in L.\, P_i(s') = t' \\ \downarrow P_i & & & & \downarrow P_i \\ P_i(s) & \xrightarrow{P^i_{i,o}} & P^i_{i,o}(P_i(s)) = P^i_{i,o}(t') & \xleftarrow{P^i_{i,o}} & \forall t' \in P_i(L) \end{array}$$

Condition simple

Important: si $\Sigma_k \subseteq \Sigma_{hi}$ alors MOC est toujours satisfaite

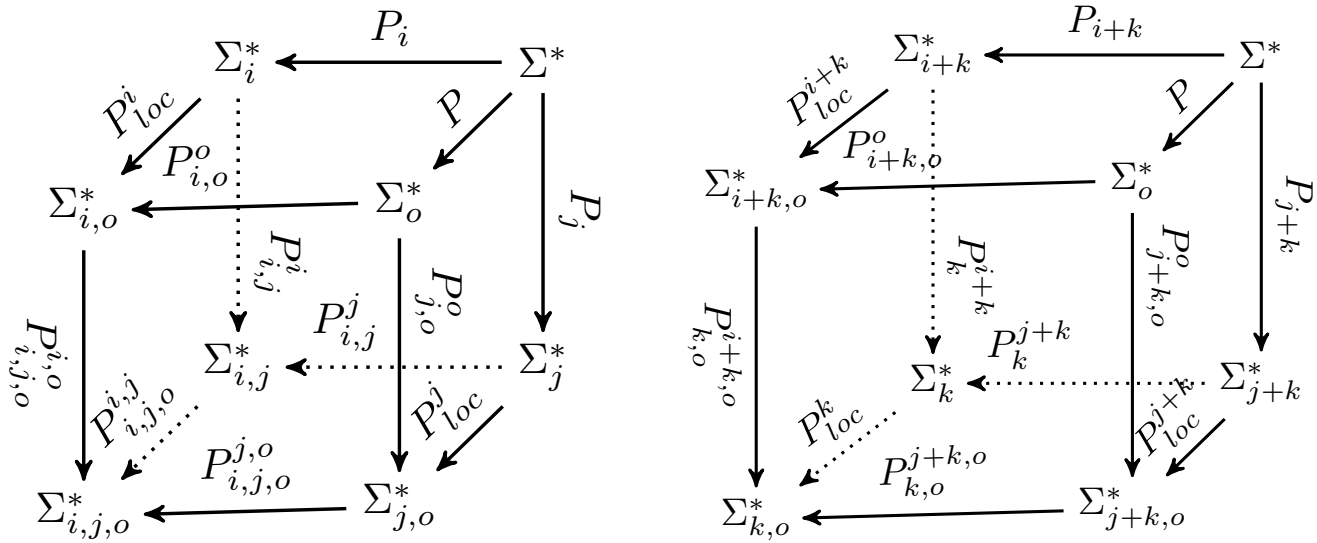


Figure: Projections avec coordination (à droite).

Calcul modulaire des superviseurs

Theorem

Soient L_i sur Σ_i, $i = 1, \ldots, n$, et $L = \|_{i=1}^{n} L_i$ avec $K \subseteq L$ une spécification. [(1)] *Calculons l'alphabet Σ_k qui contient les événements partagés t.q. K est décomposable conditionellement.* [(2)]. *Si $\Sigma_k \subseteq \Sigma_o$, et L_{i+k} et $\sup \mathrm{N}(K_{i+k}, L_{i+k}, P_{loc}^{i+k})$ sont nonconflicting, alors*

$$\|_{i=1}^{n} \sup \mathrm{N}(K_{i+k}, L_{i+k}, P_{loc}^{i+k}) = \sup \mathrm{N}(K, L, P).$$

Preuve. $\Sigma_k \subseteq \Sigma_o$ implique que $L = \|_{i=1}^{n} L_i$ est MOC pour les projections P_k^{i+k}, P_{loc}^{i+k}, et P_{loc}^{k}, for $i = 1, \ldots, n$.

---

(1) Le cas $K = \|_{i=1}^{n} K_i$ y est inclut.

(2) $P_i^{i+k} : \Sigma_{i+k}^* \to \Sigma_i^*$

Contrôle des IRM

L'imagerie par résonance magnétique (IRM):
Modéle de scanner dans la thèse de R. Theunissen (TU Einhoven)

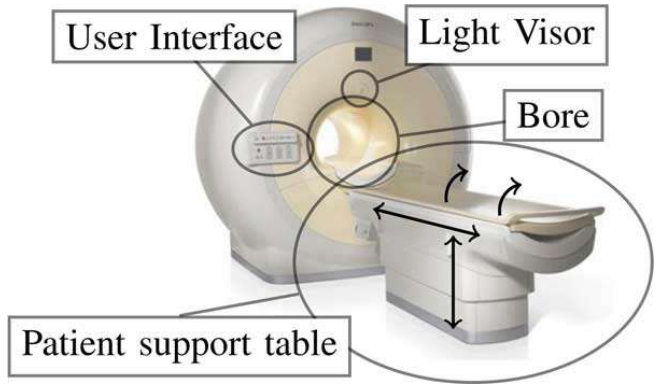
R. J. M. Theunissen, M. Petreczky, R. R. H. Schiffelers, D. A. van Beek and J. E. Rooda, "Application of Supervisory Control Synthesis to a Patient Support Table of a Magnetic Resonance Imaging Scanner," in IEEE Transactions on Automation Science and Engineering, vol. 11, no. 1, pp. 20-32, Jan. 2014

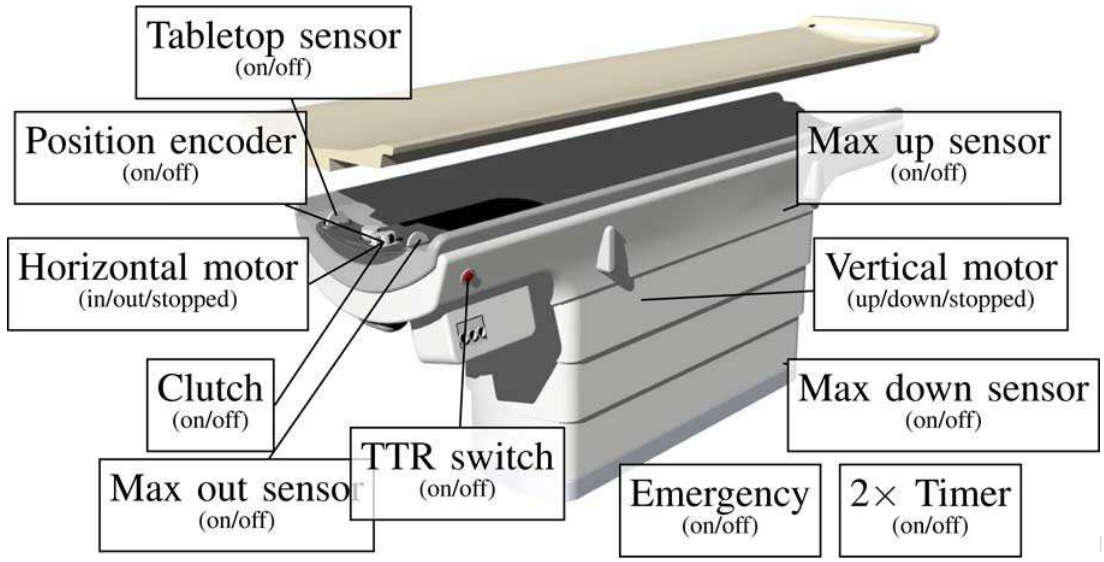
Modèle: VAxis || HAxis || UI où
VAxis=VASensors || VActuators || VARelations

Dans Theunissen et al. les événements sont tous observables.
Nous: les Spécifications sont événements qui apparaissent dans stous les autres inobservables.

MOC est guarantie (tous les événements partagés sont observables).

Résonance magnétique





Modèle pour l'imagerie par résonance magnétique

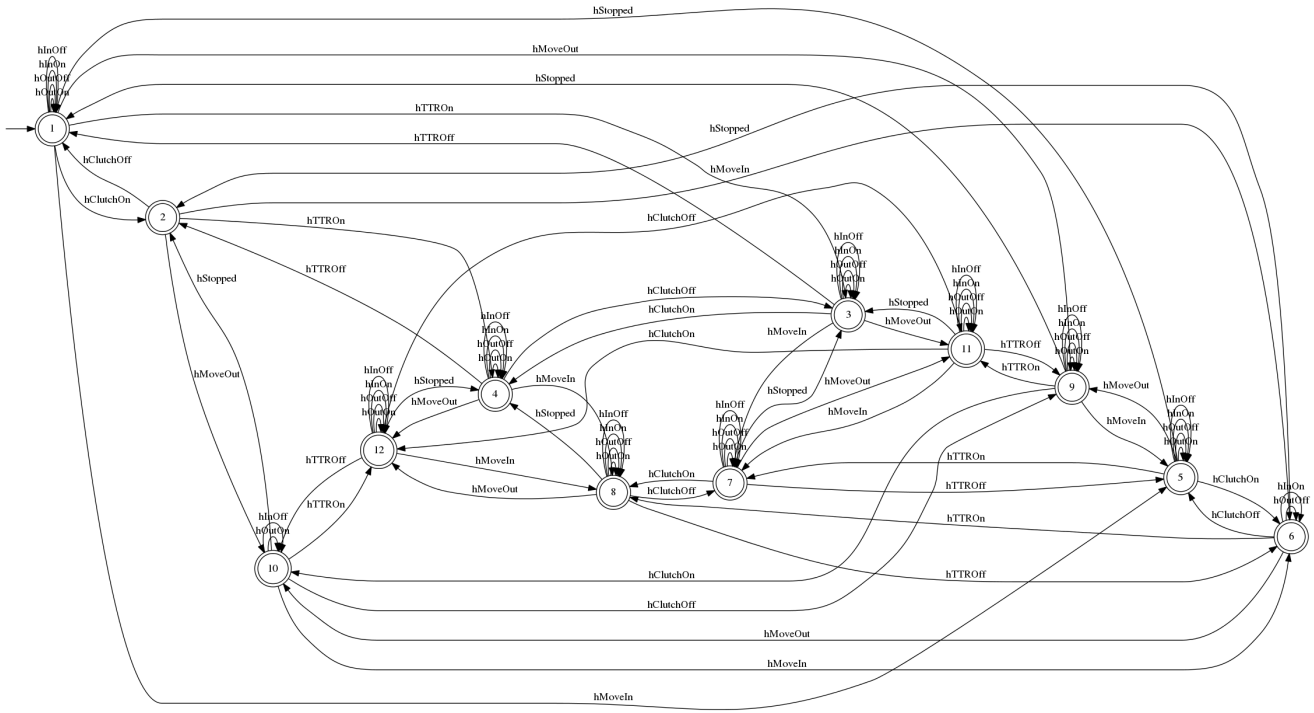


Figure: Module horizontale d'un IRM.

Movement verticale

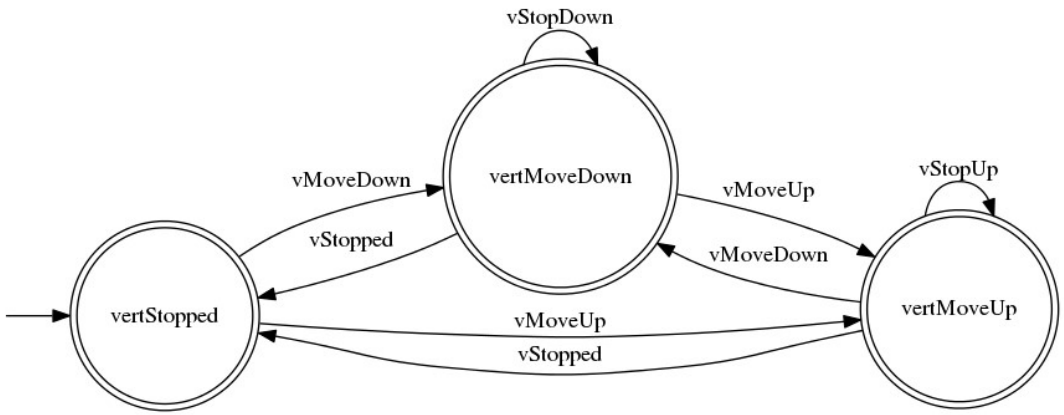


Figure: une partie du modèle vertical.

Spécification HVreq

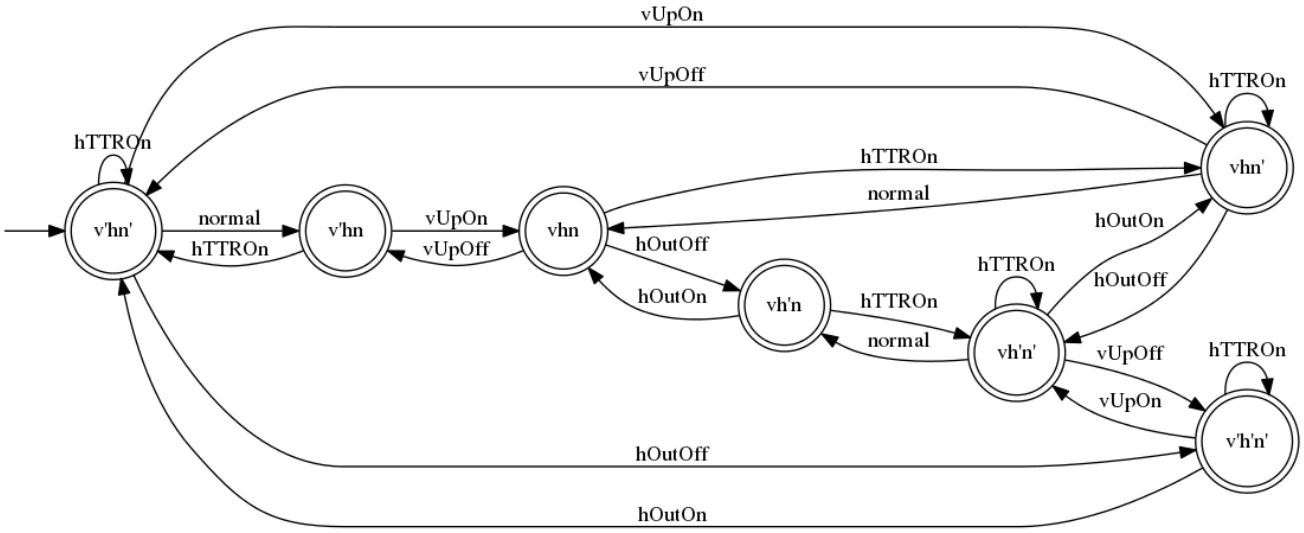


Figure: Spécification horizontale.

Exemple de Spécification

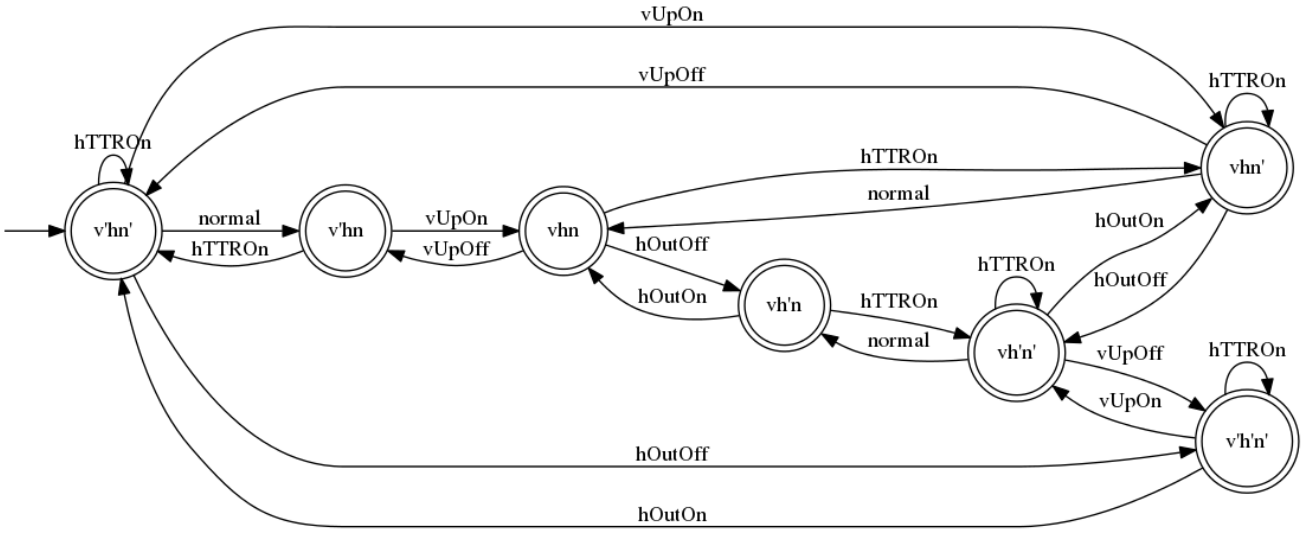


Figure: Spécification horizontale/verticale.

Champs d'application de SCT

- Systèmes de production
- AGV (véhicules/chariots autoguidés), véhicules aériens, véhicules sous-marins.
- réseaux de communication
- Systèmes logiciels (bases de données)
- Contrôle des feux de circulation
- Contrôle de scanner IRM

Table: Temps de calcul (seconds).

	VReq	HReq	HVReq (global)	UIReq (global)	Mono
Time	0.01	0.03	2.66 (9.37)	2.72 (oom)	4006

Résultats

Table: Taille des superviseurs.

	$9 \times$ local	Monolithic	$4 \times$ global	$4 \times$ high
States	3334	68672	213359	3768
Trans.	26763	616000	2772156	31486
Time	5.42	4006	hors mémoire	ca. 11

Conclusions et Perspectives

- contrôle hierarchique appliqué au contrôle modulaire

connu observation consistency (OC) et local OC (LOC) sont suffisantes pour preserver observabilité

- Mais: OC et LOC ne preservent pas la permissivité maximale
  - Modified OC (MOC) preserve la permissivité maximale
- Pour les SED modulaires MOC est satisfait par construction si les événements partagés sont tous observables
  - MOC est versatile (applicable e.g. pour contrôle hierarchique et modulaire des systèmes multi-agents avec morphisme "remplacant")
  - Problème ouvert: decidabilité de OC et MOC
  - Extension aux problèmes de cyber-sécurité: capteurs et/ou actionneurs attaqués

Merci

Temps pour vos questions.