

# Contrôle à la Ramadge/Wonham

## Supervision

- Le système  $S$  à superviser est discret, avec ensemble d'évènements (actions)  $\Sigma$  divisé en  $\Sigma_c$ , l'ensemble des actions contrôlables et  $\Sigma_u$ , l'ensemble des actions non-contrôlables.

Exemple : pour une imprimante  $S$ ,  $\Sigma_u$  représente l'ensemble des actions des utilisateurs (print, cancel, switch off, etc). L'ensemble  $\Sigma_c$  représente les actions que l'on peut autoriser/interdire, par exemple on peut interdire une impression d'un utilisateur si elle dépasse une taille donnée.

- On s'intéresse à des langages  $L$  **préfixiels** : si  $v \in L$  et  $u$  préfixe de  $v$  ( $u \leq v$ ), alors  $u \in L$ .
- Un **superviseur** pour un langage  $L$  est une fonction

$$s : L \rightarrow \mathcal{P}(\Sigma)$$

tel que  $\Sigma_u \subseteq s(w)$  pour tout  $w \in L$ . Le superviseur indique donc l'ensemble  $s(w)$  d'actions possibles après l'historique d'actions  $w$ , et n'interdit aucune action non-contrôlable.

# Supervision de systèmes finis

## Système fini

On modélise le système à superviser en tant qu'automate fini (déterministe)  $\mathcal{A} = (S, \Sigma, \delta, s_0, F)$  :  $S$  ensemble d'états,  $s_0 \in S$  état initial,  $\delta : S \times \Sigma \rightarrow S$  fonction (partielle) de transition,  $F \subseteq S$  états acceptants. On note  $\mathcal{L}(\mathcal{A})$  le langage de  $\mathcal{A}$ , ainsi que  $\text{Ch}(\mathcal{A})$  l'ensemble des mots étiquetant un chemin dans  $\mathcal{A}$  qui commence par l'état initial.

## Système supervisé

Etant donné un automate  $\mathcal{A}$  et un superviseur  $s$ , le **comportement supervisé** de  $\mathcal{A}$  est le langage préfixiel  $\text{Ch}_s(\mathcal{A})$  défini de manière inductive par :

- $\epsilon \in \text{Ch}_s(\mathcal{A})$
- Pour tout  $w \in \text{Ch}_s(\mathcal{A})$  et  $a \in \Sigma$  :  $wa \in \text{Ch}_s(\mathcal{A})$  ssi  $wa \in \text{Ch}(\mathcal{A})$  et  $a \in s(w)$ .

Le **langage supervisé** de  $\mathcal{A}$  :  $\mathcal{L}_s(\mathcal{A}) = \mathcal{L}(\mathcal{A}) \cap \text{Ch}_s(\mathcal{A})$ .

Le système supervisé est **non-bloquant**, si tout chemin supervisé peut atteindre un état acceptant : pour tout  $u \in \text{Ch}_s(\mathcal{A})$  il existe  $v$  t.q.  $uv \in \mathcal{L}_s(\mathcal{A})$ .

# Supervision de systèmes finis

## Rem.

La propriété d'être non-bloquant est équivalente à l'égalité entre  $\text{Ch}_s(\mathcal{A})$  et la clôture prefixielle de  $\mathcal{L}_s(\mathcal{A})$ , notée  $\text{Pref}(\mathcal{L}_s(\mathcal{A}))$ .

## Contrôlabilité

Un langage  $\text{Spec} \subseteq \Sigma^*$  est **contrôlable** (par rapport à un système  $\mathcal{A}$ ), si

$$\text{Pref}(\text{Spec})\Sigma_u \cap \text{Ch}(\mathcal{A}) \subseteq \text{Pref}(\text{Spec})$$

Autrement dit : toute action non-contrôlable qui est possible après une exécution de  $\text{Spec}$  dans le système  $\mathcal{A}$ , est possible aussi dans  $\text{Spec}$ .

## Rem.

L'ensemble vide  $\emptyset$  et l'ensemble  $\Sigma^*$  sont contrôlables (quelque soit  $\mathcal{A}$  avec alphabet  $\Sigma$ ).

# Supervision de systèmes finis

## Prop.

Soit  $\mathcal{A}$  un automate fini et  $\text{Spec} \subseteq \text{Ch}(\mathcal{A})$  un langage préfixiel régulier et non-vidé. Alors  $\text{Spec}$  est contrôlable ssi il existe un superviseur  $s$  t.q.  $\text{Ch}_s(\mathcal{A}) = \text{Spec}$ . De plus, si un tel  $s$  existe, alors il existe un à mémoire finie.

## Prop.

Soit  $\mathcal{A}$  un automate fini où tout état est co-accessible, et  $\text{Spec} \subseteq \mathcal{L}(\mathcal{A})$  un langage régulier non-vidé. Les deux assertions suivantes sont équivalentes :

- 1 il existe un superviseur  $s$  t.q.  $\mathcal{A}$  supervisé par  $s$  est non-bloquant et  $\mathcal{L}_s(\mathcal{A}) = \text{Spec}$ .
- 2  $\text{Spec}$  est contrôlable et  $\text{Pref}(\text{Spec}) \cap \mathcal{L}(\mathcal{A}) = \text{Spec}$ .

De plus, si un tel  $s$  existe, alors il existe un à mémoire finie.

# Exemples

## Spec préfixiel contrôlable

$\mathcal{L}(\mathcal{A}) = (a^+b)^*a^*$ , Spec “au plus deux  $a$  consécutifs” :

$$\text{Spec} = (ab + aab)^*(\epsilon + a + aa)$$

Spec est contrôlable ssi  $a \in \Sigma_c$ . Si  $a \in \Sigma_c$ , alors superviser  $\mathcal{A}$  revient à compter le nombre de  $a$  vus depuis le dernier  $b$ .

## Spec préfixiel non-contrôlable

$\mathcal{L}(\mathcal{A}) = (a^+b)^*a^*$ , Spec “au plus deux  $a$  consécutifs avant le dernier  $b$ ” :

$$\text{Spec} = (ab + aab)^*a^*$$

Si  $b \in \Sigma_u$  alors  $aaa \in \text{Spec}$ , mais  $aaab \in \text{Ch}(\mathcal{A}) \setminus \text{Spec}$ .

## Spec contrôlable (non-préfixiel)

$\mathcal{L}(\mathcal{A}) = (a^+b)^*$ , Spec =  $a^+b$ .

Si  $\Sigma_c = \{a\}$  alors Spec est contrôlable. On a aussi  $\text{Pref}(\text{Spec}) \cap \mathcal{L}(\mathcal{A}) \subseteq \text{Spec}$ .

# Supervision de systèmes finis

## Sous-langage supervisé maximal

Soit  $\text{Spec} \subseteq \text{Ch}(\mathcal{A})$ . Le sous-langage contrôlable maximal  $\text{Spec}_{\max} \subseteq \text{Spec}$  de  $\text{Spec}$  est défini par

$$\text{Spec}_{\max} = \max \mathcal{C}(\text{Spec})$$

où  $\mathcal{C}(\text{Spec})$  est l'ensemble des parties contrôlables  $K$  de  $\text{Spec}$  (Rem. : l'union de 2 parties contrôlables est contrôlable aussi, donc le max est bien défini.).

## $\text{Spec}_{\max}$ et point fixe

On peut décrire (et calculer)  $\text{Spec}_{\max}$  par un (plus grand) point fixe. On définit une fonction  $F : \mathcal{P}(\Sigma^*) \rightarrow \mathcal{P}(\Sigma^*)$  par

$$F(K) = \max\{J \subseteq K \mid \text{Pref}(J)\Sigma_u \cap \text{Ch}(\mathcal{A}) \subseteq \text{Pref}(K)\}$$

## Prop.

$\text{Spec}_{\max}$  est le plus grand point fixe de  $F$  : (1)  $\text{Spec}_{\max} = F(\text{Spec}_{\max})$ , et (2) pour tout  $K \subseteq \text{Spec}$  tel que  $F(K) = K$ , on a  $K \subseteq \text{Spec}_{\max}$ .

# Sous-langage supervisé maximal

## Calcul de $\text{Spec}_{\max}$

On définit une suite décroissante de langages par :

$$T_0 = \text{Spec}, \quad T_{n+1} = F(T_n), \quad n \geq 0$$

ainsi que  $T_\infty = \bigcap_{n \geq 0} T_n$ . On note aussi que  $\text{Spec}_{\max} \subseteq T_\infty$ .

## Exemple

Dans l'exemple  $\mathcal{L}(\mathcal{A}) = (a^+b)^*a^*$ ,  $\text{Spec} = (ab + aab)^*a^*$ ,  $\Sigma_u = \{b\}$  on calcule

$$T_1 = F(\text{Spec}) = (ab + aab)^*(\epsilon + a + aa), \quad F(T_1) = T_1 = T_\infty$$

Ici,  $\text{Spec}_{\max} = T_\infty$ .

## Prop.

Soit  $\mathcal{A}$  un automate fini où tous les états finis sont acceptants, et  $\text{Spec} \subseteq \text{Ch}(\mathcal{A})$  régulier. Alors  $\text{Spec}_{\max} = T_\infty$  et il existe  $n \geq 0$  tel que  $T_\infty = T_n$ .