Problème d'(A, B)-invariance sur les langages

Ines Klimann*
ines.klimann@liafa.jussieu.fr

Résumé

On se propose de résoudre, dans le cadre de la théorie des langages rationnels, l'équivalent d'un problème extrêmement classique en automatique : la recherche d'un sous-espace (A, B)-invariant maximal.

1. Une justification concrète du problème

En automatique, on est souvent confronté à un problème du type suivant : on a un signal d'entrée et un signal de sortie qui en dépend, mais certaines perturbations viennent s'ajouter. Pour tenter d'atténuer ces perturbations, on a recours à la rétroaction. C'est une méthode classique (notament en électronique) qui consiste à réinjecter une partie du signal de sortie sur l'entrée. Le nouveau système ainsi obtenu est un *système à retard.*

Example 0.1 — On considère le circuit électronique de la figure 1. Il réalise $V_s = V_e$.

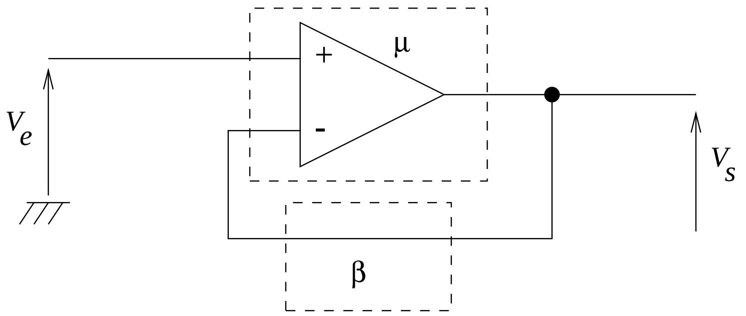


FIG. 1: *Montage suiveur.*

De façon plus générale, le comportement d'un système à retard est régi par des équations du type

$$\begin{cases} \dot{x} = Ax + Bu + Ew \\ y = Cx \end{cases}$$

*Ines Klimann – LIAFA – Université Denis Diderot Paris VII – Case 7014
2, place Jussieu – F-75251 Paris Cedex 05

où x, y, u et w appartiennent à certains espaces vectoriels et A, B, C et E sont des opérateurs linéaires fixés ($\dot{x}$ désigne la dérivée de x par rapport au temps). Le terme x représente l'entrée, le terme y la sortie, le terme u un paramètre que l'utilisateur peut régler et le terme w des perturbations dont on ne connaît rien.

Le problème est le suivant : trouver le plus grand ensemble de conditions initiales tel que, quelles que soient les perturbations, la sortie est identiquement nulle. On réduit assez simplement ce problème à la recherche du sous-espace (A, B)-*invariant* maximal de $\ker C$, c'est-à-dire le plus grand sous-espace V de $\ker C$ qui vérifie $AV \subseteq V + \operatorname{Im} B$.

Dans le cadre d'espaces vectoriels de dimension finie, cet espace est très simple à trouver par restrictions successives à des sous-espaces vectoriels. Mais on ne sait pas encore résoudre ce problème dans le cadre de modules car le sous-module d'un module de type fini n'est pas nécessairement de type fini. On se propose ici de résoudre le problème formellement analogue dans le cadre des langages rationnels : soit A, B et K, trois langages rationnels fixés, comment trouver $\sup\{X \mid AX \subseteq X \cup B \text{ et } X \subseteq K\}$?

2. Les outils de résolution

On travaille sur un alphabet fini Σ. Si E est un ensemble, $\mathcal{P}(E)$ désigne l'ensemble des parties de E.

La *coupe à gauche* d'un langage X par un langage A est le langage

$$A \backslash X = \{w \in \Sigma^* \mid Aw \subseteq X\}.$$

On voit facilement que $A \backslash X = \bigcap_{v \in A} v \backslash X$. Si X est reconnu par un automate complet déterministe $(\Sigma, Q, i, \cdot, F)$, le langage $v \backslash X$ est alors reconnu par l'automate $(\Sigma, Q, i \cdot v, \cdot, F)$. Cette remarque amène naturellement à la définition suivante.

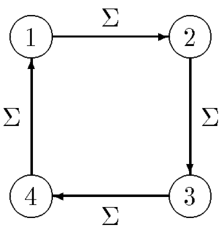
Definition 0.1 Un *multi-automate* est une structure $\mathcal{A} = (\mathbf{B}, \Phi)$, où

- $\mathbf{B}$ est un *système de transitions*, c'est-à-dire un triplet $(\Sigma, Q, \cdot)$ tel que $(q, x) \mapsto q \cdot x$ est une fonction de $Q \times \Sigma$ dans Q ; $\mathbf{B}$ est la *base* de A.
- Φ est une formule booléenne positive (c'est-à-dire sans négation) sur $Q \times \mathcal{P}(Q)$ appelée *formule d'acceptation*.

On note $L^{\mathbf{B}}(j, F)$ le langage reconnu par l'automate $(Q, j, \cdot, F)$. Le langage $L^{\mathbf{B}}(\Phi)$ *reconnu* par le multi-automate $(\mathbf{B}, \Phi)$ est l'image de la formule booléenne Φ par le morphisme du treillis distributif libre sur $Q \times \mathcal{P}(Q)$ dans le treillis distributif $\mathcal{P}(\Sigma^*)$ qui à (j, F) associe $L^{\mathbf{B}}(j, F)$.

Ce nouveau type d'automate permet de simuler les opérations d'union et d'intersection sur des langages reconnus par des automates (ou des multi-automates) de même base.

La coupe à gauche est liée aux multi-automates par la formule suivante : si A et X sont deux langages, X étant reconnu par le multi-automate $(\mathbf{B}, \Phi)$, alors $A \backslash X$



$$\Phi = (1, \{2,3,4\})$$
$$\Psi = (1, \{2,3,4\}) \wedge (3, \{2,3,4\})$$

FIG. 2: *$X = \Sigma^* - (\Sigma^4)^*$ est reconnu avec Φ et $A \backslash X$ avec Ψ.*

est reconnu par le multi-automate $\left(\mathbf{B}, \bigwedge_{v \in A} s_v(\Phi)\right)$, où s_v est l'endomorphisme du treillis distributif libre sur $Q \times \mathcal{P}(Q)$ qui à (j, F) associe $(j \cdot v, F)$.

Example 0.2 — Soit A le langage constitué des mots de longueur paire et X le langage des mots de longueur non divisible par 4. La figure 2 représente un multi-automate qui reconnaît le langage X. On applique la formule précédente : $1 \cdot A = \{1, 3\}$, donc le multi-automate de base donnée par la figure 2 et de formule d'acceptation $(1, \{2,3,4\}) \wedge (3, \{2,3,4\})$ reconnaît le langage $A \backslash X$. On peut en conclure que

$$\begin{aligned} A \backslash X &= \{w \in \Sigma^* \mid |w| \not\equiv 0 \ [4]\} \cap \{w \in \Sigma^* \mid |w| \not\equiv 2 \ [4]\} \\ &= \{w \in \Sigma^* \mid |w| \text{ est impaire}\}. \end{aligned}$$

3. Une idée de résolution

On applique alors une méthode de type point fixe pour résoudre le problème. On construit la suite de langages $(K_n)_{n \geq 0}$ par récurrence de la façon suivante :

$$\begin{gathered} K_0 = K \\ \text{pour } n \geq 0, \ K_{n+1} = (A \backslash (B \cup K_n)) \cap K_n. \end{gathered}$$

La suite $(K_n)_{n \geq 0}$ décroît, de limite $K_\infty = \bigcap_{n \geq 0} K_n$. On montre que le langage K_∞ est solution de notre problème et on construit un multi-automate qui reconnaît K_∞.

La complexité temporelle est en

$$O\left(n_A (n_B + n_K)^2 + n_K \cdot \frac{2^{n_B}}{\sqrt{n_B}}\right),$$

où n_A (respectivement n_B et n_K) est la taille de l'automate minimal de A (respectivement de B et de K).

A noter que le problème a aussi été résolu dans le cadre de séries à coefficients dans un semi-anneau idempotent complet.