Coalgèbre et systèmes

Jan Komenda

Institute of Mathematics, Czech Academy of Sciences, Brno, Czech Republic

Séminaire du LISA

Angers, France, 18 juin 2012

Outline

Outline

1. Introduction
2. Coalgèbre Universelle
3. Automates á multiplicités déterministes
4. Application aux systèmes linéaires
5. Automates à multiplicités
6. automates (max,+)
   - Automates (max,+) comme coalgébres

Réferences

- J.J.M.M. Rutten. Universal coalgebra: a theory of systems. Theoretical Computer Science 249(1), 2000, pp. 3-80.
- J.J.M.M. Rutten Behavioural differential equations: a coinductive calculus of streams, automata, and power series. Theoretical Computer Science Volume 308(1–3), pp. 1–53, 2003.
- J.J.M.M. Rutten. Algebraic specification and coalgebraic synthesis of Mealy automata. Proceedings FACS 2005, ENTCS Vol. 160, Elsevier, 2006, pp. 305-319.
- J.J.M.M. Rutten. Coalgebraic foundations of linear systems (an exercise in stream calculus). Proceedings CALCO 2007, LNCS 4624, Springer, 2007, pp. 425-446.
- J. Komenda. Coinduction in Concurrent Timed Systems. Information and Computation Vol. 211, pp. 77–105, 2012.
- F. Bonchi, M.M. Bonsangue, M. Boreale, J.J.M.M. Rutten and A. Silva. A coalgebraic perspective on linear weighted automata. Information and Computation Vol. 211, pp. 77–105, 2012.

Coalgèbre et systèmes

- Coalgèbre est duale à l'algèbre $F(X) \to X$
- Systèmes de transition etiquettés y compris temporisés et probabiliste sont coalgèbres
- Différents types d'automates sont coalgèbres des foncteurs dans la Categorie **Set**
- Automates à multiplicités sont aussi coalgèbres
- Les automates deterministes admettent coalgèbres finales: langages, langages partiels, series formelles (Moore)
- Systèmes linèaires et automates à multiplicités sont coalgèbres dans la Categorie **Vec**

Outline

Definition de Coalgèbre

- Categorie: objets et morphismes entre objets (y compris identié autour d'un objet)(Set, Vec, Groupes, etc.)
- Foncteurs: entre categories ou Endofoncteurs: **Cat** $\rightarrow$ **Cat**
- Foncteurs sont définis sur les objets et sur les morphismes
- Ils respectent les identités et morphismes
- Coalgèbre sur la catégorie **Set** correspondant au foncteur $F : Set \rightarrow Set$ (F-Coalgèbre) est la structure (S, α), où S est un ensemble et $\alpha_S : S \rightarrow F(S)$ donne la structure (dynamique d'un système)
- Exemples: $F(S) = S \times A, F(S) = S^A, F(S) = Pwr(S)^A, F(S) = Pwr(S \times A), F(S) = (\{\emptyset\} + S)^A$, $F(S) = B \times S^A$, $F(S) = (B \times S)^A, \ldots$

Eléments base de la théorie des coalgèbres

- **F-Homomorphisme** de $S = (S, \alpha_S)$ à $T = (T, \alpha_T)$ est la function $f: S \to T$ telle que le diagramme commute:

$$\begin{tikzcd} F(S) \arrow[d, "F(f)"] & S \arrow[l, "\alpha_S"'] \arrow[d, "f"'] \\ F(T) & T \arrow[l, "\alpha_T"'] \end{tikzcd}$$

- En bref, $f: S \to T$ est un F-Homomorphisme si'il respecte la structure donné par F: (préserve et refléte les transitions)
- Exemples: homomorphisms des automates, des automates partiels, des automates á multiplicité, des systèmes linèaires

Bisimulation

F-Bisimulation Relation $R \subseteq S \times T$ est une bisimulation s'il existe une structure de F-coalgèbre sur R telle ques les projections π_S et π_T de R à S et T sont des F-homomorphismes.

$$\begin{tikzcd}
S \arrow[d, "\alpha_S"'] & R \arrow[l, "\pi_S"'] \arrow[r, "\pi_T"] \arrow[d, "\alpha_R"'] & T \arrow[d, "\alpha_T"'] \\
F(S) & F(R) \arrow[l, "F(\pi_S)"'] \arrow[r, "F(\pi_T)"] & F(T)
\end{tikzcd}$$

- Notation: $s \sim t$ for $\langle s, t \rangle \in R$
- Exemples: bisimulation des automates, automates partiels, des systèmes linèaires

Coalgèbre finale

- (B, α_B) est coalgèbre finale de F si pour toute F-coalgèbre (S, α_S) il existe un unique $F - homomorphisme\ l : S \to B$ appelé comportemental
- Pour $s \in S$, $l(s)$ est le comportement de l'état $s \in S$ (comme si s était initial).
- Pour $s, s' \in S$, $s \sim s'$ iff $l(s) = l(s')$ (en particulier dans la coalgèbre finale $s \sim s'$ iff $s = s'$
- Existence de coalgèbre finale depend du F: conditions suffisantes e.g.: foncteurs continus (e.g. polynomiaux) ou bornés admettent coalgèbres finales
- $F = Pwr$ n'admet pas de coalgèbre finale, mais si l'on se restreint aux sous ensembles finis ou dénombrables, alors coalgèbre finale existe
- Il existe des techniques pour construire les coalgèbre finales: comme limites des diagrammes

Coïnduction: définitions et preuves

- Preuve par coïnduction: bisimilation implique égalité sur les coalgèbres finales
- Example: pour $F = S^A$ (automates), $B = Pwr(A^*)$ (langage), sur les langages bisimilarité coïncide avec égalité Preuve par coïnduction: par propriété de morphisme il suffit de donner la structure sur la fonction que l'on veux définir
- Exemple: pour $F = S \times A$ (automates) on a $B = A^\omega$, avec $\langle tail, head \rangle$, alors le zip des deux séquence $s = (a_1, a_2, \dots)$ et $t = (b_1, b_2, \dots)$ donné par:

  $zip(s,t) = (a_1, b_1, a_2, b_2, \dots)$ est bien défini par

  $tail(zip(s,t)) = zip(t, tail(s))$ et $head(zip(s,t)) = head(s)$

Propriétés des séquences infinies

Notation.(Séquences élémentaires $A = K$)

- Pour $k \in K$, $[k] = (k, 0, 0, \dots) = k$
- $X = (0, 1, 0, 0, \dots)$
- Polynomiaux: support fini
- Pour $\sigma, \tau \in K^{\omega}$:
  $(\sigma + \tau)(n) = \sigma(n) + \tau(n) \quad (\sigma \times \tau)(n) = \sum_{i=0}^{n} \sigma(i) + \tau(n-i)$
- Si $\sigma(0) \neq 0$, alors σ^{-1} t.q.
  $\sigma^{-1} \times \sigma = 1$ existe.
- Définition coalgébrique correspondantes: ($head(\sigma) = \sigma(0)$ et $tail(\sigma) = \sigma'$
  $(\sigma + \tau)' = \sigma' + \tau'$ avec $(\sigma + \tau)(0) = \sigma(0) + \tau(0)$
  $(\sigma \times \tau)' = \sigma' \times \tau + \sigma(0) \times \tau'$ avec $(\sigma \times \tau)(0) = \sigma(0).\tau(0)$
  $(\sigma^{-1})' = -[\sigma(0)]^{-1} \times \sigma' \times (\sigma^{-1})$ avec $(\sigma^{-1})(0) = (\sigma(0))^{-1}$

Solution des équations différentielles linéaires

- **Théorème fondamental.** Pour tout $\sigma \in A^{\omega}$ il y a
  $\sigma = \sigma(0) + (X \times \sigma)'$
- Pour $(\sigma,\ \tau)' = (\sigma,\ \tau) \times M$, $(\sigma,\ \tau)(0) = N$, où
  M est une matrice 2×2 sur K et N un vecteur
- $(\sigma,\ \tau) = (\sigma,\ \tau)(0) + X \times (\sigma,\ \tau)' = N + X \times M \times (\sigma,\ \tau)$
- Alors,
  $(\sigma, \tau) = N \times (I - (X \times M))^{-1}$ est la solution!
- **Rémarque.** Pour tout $s \in (A^*)^K$ (série formelle) il y a
  $s = s(\varepsilon) + \sum_{a \in A}(X_a \times s_a)$, oú $(X_a)(\varepsilon) = 0$, $(X_a)_a = 1$ et $(X_a)_a = 1$.
  Rappel: $s_a(w) = s(aw)$

Exemple: équation différentielle linéaire

Prenons système:
$\sigma' = \tau$ et $\tau' = -\sigma + 2\tau$ avec
$(\sigma(0) = 1,\ \tau(0) = 2$.
Alors,
$(\sigma,\ \tau)' = (\sigma,\ \tau) \times M$ et $(\sigma,\ \tau)(0) = N$, avec $M = \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix}$ et
$N = [1,\ 2]$.
On obtient:

$$(\sigma, \tau) = N \times (I - (X \times M))^{-1} = (1,\ 2) \times \begin{pmatrix} 1 & X \\ -X & 1 - 2X \end{pmatrix}^{-1} =$$

$$= (1,\ 2) \times \begin{pmatrix} \frac{1-2X}{(1-X)^2} & \frac{-X}{(1-X)^2)} \\ \frac{X}{(1-X)^2)} & \frac{1}{(1-X)^2)} \end{pmatrix} = \left(\frac{1}{(1-X)^2} \quad \frac{2-X}{(1-X)^2})\right).$$

Solution ci-dessus est rationnelle (toujours le cas)!
Rémarque. Pour les séries formelles on aurait
$S = N \times (I - \sum_{a \in A}(X_a \times M_a))^{-1}$ comme solution de
$S_a = S \times M_a$ et $S(\varepsilon) = N$.

Automates partiels

- Automates partiels sont coalgèbres: $F(S) = \mathscr{B} \times (\{\emptyset\} + S)^A$ avec $\mathscr{B} = \{0,1\}$, i.e. $F = \langle o, t \rangle$: $o : S \to \mathscr{B}$ et $t : S \to (\{\emptyset\} + S)^A$
- L'automate final de comportement de tous les états de tous les automates partiels est formé de langages partiels $\mathscr{L} = \{(L_m, L) : L_m \subseteq L = \bar{L}\}$.
- Transitions dans $\mathscr{L}$ sont des quotients (dérivées par a): $t(L_m, L)(a) = ((L_m)_a, L_a)$ ssi $(L_m)_a = \{w \in A^* : aw \in L_m\}$ existe (dans ce cas L_a existe aussi)
- $o(L_m, L) = 1$ ssi $\varepsilon \in L_m$
  Il satisfait le principe des preuves par coïnduction: bisimilarité implique équalité.
- Definition de operations sur langages partiels par coinduction: (produit synchrone, langages supremales contrôlables etc.)

Exemples de définitions par coïnduction

Soient $K \subseteq A_1$ et $L \subseteq A_2$ deux langages partiels.
Définition. Produit Synchrone.

$$(K \parallel L)_a = \begin{cases} K_a \parallel L_a, & \text{si } a \in A_1 \cap A_2, \\ K_a \parallel L, & \text{si } a \in A_1 \setminus A_2, \\ K \parallel L_a, & \text{si } a \in A_2 \setminus A_1, \end{cases}$$

et $o(K \parallel L) = 1 \quad \text{ssi} \quad o(K) = 1$ et $o(L) = 1$.
Definition. Produit supervisé. Soient $A_1 = A_2 = A$.

$$(K/_{A_u}L)_a = \begin{cases} K_a/_{A_u}L_a, & \text{si } K \xrightarrow{a} \text{ et } L \xrightarrow{a}, \\ 0/_{A_u}L_e, & \text{si } K \stackrel{a}{\not\to} \text{ et } L \xrightarrow{a} \text{ et } a \in A_u, \\ \emptyset, & \text{sinon}, \end{cases}$$

et $o(K/_{A_u}L) = 1$ ssi $o(L) = 1$.

Outline

Automates á multiplicités déterministes comme coalgèbres

- Automates de Mealy: avec les entrées dans A et les sortie dans K sont coalgèbres
  (S, t), avec S l'ensemble des états et la fonction de transition:
  $t : S \to (K \times S)^A$.
- Un automate partiel de Mealy est (S, t), où
  $t : S \to (\{\emptyset\} + (K \times S))^A$.
- Afin de considérer les états finaux, posons $(S, (o, t))$, où
  $o : S \to K$ et $t : S \to (\{\emptyset\} + (K \times S))^A$.
- Examples de multiplicités :
  $K = \mathbb{R}_{min} = (\mathbb{R} \cup \{\infty\}, min, +, \infty, 0)$ automates (min,+) (coût)
  $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ automates (max,+) (temps)
  $K = \mathscr{I}^{max}_{max} = (\mathbb{R}_{\max} \times \mathbb{R}_{\max} \cup (-\infty, -\infty), \oplus, \otimes, (-\infty, -\infty), (0, 0))$
  automates d'intervales
  $([0, 1], +, \times, 0, 1)$ automates probabilistes

Automate Final de Mealy

- Comportements des automates de Mealy sont fonctions causales entre les séquences infinies d'entrées et de sorties: $f : A^\omega \to K^\omega$. $f :\ A^\omega \to K^\omega$ est *causale* si $\forall n \in N$, $\sigma, \tau \in A^\infty$: $\forall i :\ i \leq n$: $\sigma(i) = \tau(i)$ then $f(\sigma)(n) = f(\tau)(n)$.
- Dérivées des suites: $\omega = (\omega_0, \omega_1, \dots) \in K^\omega \to \omega' = (\omega_1, \dots)$.
- Structure d'automate de la coalgèbre finale: $t(f) = \langle f[a], f_a \rangle$ $f[a] = f(a : \sigma)(0)$ et $f_a(\sigma) = f(a : \sigma)'$
- $A^\infty = A^\omega \cup A^+$, où $A^+ = A^* \setminus \{\lambda\}$ f est coherent si $\sigma \in A^\omega$: $f(\sigma)(k) = \emptyset$ alors $f(\sigma)(n) = \emptyset$ pour tout $n > k$.
- $\mathscr{F} = (\mathscr{F}, t_{\mathscr{F}})$ est la coalgébre finale des automates partiels de Mealy : $\mathscr{F} = \{f :\ A^\omega \to (1+K)^\omega \mid f \text{ est causale et coherent}\}$.

$$t_{\mathscr{F}}(f)(a) = \begin{cases} \langle f[a], f_a \rangle & \text{si } f[a] \neq \emptyset \in 1, \\ \emptyset & \text{sinon,} \end{cases}$$

Presentation équivalente de comportements

- $s_0 \stackrel{\sigma(0)|k_0}{\to} s_1 \stackrel{\sigma(1)|k_1}{\to} s_2 \cdots \stackrel{\sigma(n)|k_n}{\to} s_{n+1}$.
  Posons $l(s_0)(\sigma)(n) = k_n$.
- $\mathscr{F}$ est isomorphe aux fonctions entre séquences finies ou infinies des entrés et les séquences finies ou infinies des sortie!

  $\mathscr{F}_\infty = \{f : A^\infty \to K^\infty | f$ préserve la longueur,causale et$dom(f)$ fermés$\}$.
- $f[a] = f(a)(0)$ si f est défie pour $a \in A$.
- $f_a : \; A^\infty \to (1+K)^\infty$ donnée par $f_a(s) = f(a:s)'$
- $$t_{\mathscr{F}_\infty}(f)(a) = \begin{cases} \langle f[a], f_a\rangle & \text{si } f[a] \neq \emptyset \\ \text{pas defini} & \text{sinon,} \end{cases}$$

Théorème fondamental des fonctions sur des séquences

Théorème. Pour tout $f \in \mathscr{F}$ et $\sigma = (\sigma(0), \sigma(1), \ldots, \sigma(k), \ldots) \in A^\omega$ nous avons:

$$f(\sigma) = f(\sigma)(0) \oplus X f_{\sigma(0)}(\sigma')(0) \oplus \ldots X^k f_{\sigma(0) \ldots, \sigma(k-1)}(\omega^{(k)})(0) \oplus \ldots$$

ou bien (equivalently),

$$f(\sigma) = f[\sigma(0)] \oplus X f_{\sigma(0)}[\sigma(1)] \oplus \ldots X^k f_{\sigma(0) \ldots, \sigma(k-1)}[\sigma(k)] \oplus \ldots$$

Proposition. Pour tout $f \in \mathscr{F}_\infty$, $\omega \in A^\infty$, et $a \in A$: $f(a) : f_a(\omega) = f(a\omega)$. Plus généralement, pour tout $u \in A^+$ et $\omega \in A^\infty$: $f(u) : f_u(\omega) = f(u\omega)$.

Proprietés des séquences infinies

Notation.(Séquences élémentaires)
Pour $k \in K$, $\sigma = (\sigma(0) : \sigma') \in A^\infty$, et $a \in A$ nous définissons:

$$(f^\infty[a] \odot g)(\sigma(0) : \sigma') = \begin{cases} f(\sigma(0)) : g(\sigma') & \text{si } a = \sigma(0) \in dom(f), \\ \text{pas défi} & \text{sinon,} \end{cases}$$

Théorème. Pour tout $f \in \mathscr{F}_\infty$ nous avons: $f = \bigoplus_{a \in A} f^\infty[a] \odot f_a$.

Théorème. Pour tout $f \in \mathscr{F}_\infty$ et $a \in A$: $(f^\infty[a] \odot f)_a = f$

Outline

Systèmes linéaires comme automates de Mealy

- Notation: $F : V \to V$ dynamique interne ($x' = Ax$), $G : I \to V$ input and $H : V \to O$ output
- Systèmes linéaires sont Machines de Mealy: (V, Φ) où $\Phi : V \to (O \times V)^I$ donné par $\Phi(v)(i) = \langle H(v), F(v) + G(i) \rangle$
- Systéme final des comportements est formé par $\{\Gamma : \ I^\omega \to O^\omega\}$ (transfert)

Outline

Automates à multiplicités algébriquement

$G = (S, A, \alpha, t, \beta)$ sur K: avec
S ensemble des états, $\alpha : Q \to K$, $t : Q \times A \times Q \to K$, and $\beta : Q \to K$, appelés délais initial, fonction de transition et transition, délais final.

- **Codage:** la valeur $t(q, a, q') \in K \setminus 0_K$ correspond á l'existence d'une $a-$transition de q 'à q' et $t(q, a, q') = 0_K$ s'il n'y pas de transition from de q 'à q' étiquettée par a.
  On code t par des matrices de morphisme $\mu(a) \in K^{\|Q\| \times \|Q\|}$: pout tout $a \in A$: $[\mu(a)]_{q,q'} = t(q, a, q')$
- Un automate à multiplicité reconnais la série formelle définie comme suit:
  $I(G)(a_1 \dots a_n) = \sum\{\alpha(x_0).k_1 \dots k_n.\beta(x_n) \,\|\exists x_0 \overset{a_1\|k_1}{\rightarrow} x_1 \cdots \overset{a_n\|k_n}{\rightarrow} x_n\}$
- Algébriquement: $I_G(w) = \alpha \times \mu(w) \times \beta$

Bisimulation des automates à multiplicités (nondeterministes)

Bisimulation sur $S = (S, t)$ est une relation $R \subseteq S \times R$ t.q. $\forall s \in S$ et $\forall s' \in S$: if $\langle s, s' \rangle \in R$ then

(i) $o(s) = o(t)$

(ii) $\forall a \in A$ and $\forall s'' \in S$

$$\sum_{s'' :\ \langle s'', s' \rangle \in R} t(s)(a)(s'') = \sum_{s'' :\ \langle s'', s' \rangle \in R} t(s')(a)(s'')$$

Remark. Pour les automates à multiplicités partiels, il faut ajouter la condition classique sur l'existence des paires des états qui sont liés par R après chaque transition!
(iii) $\forall a \in A, \exists q :\ s \xrightarrow{a} q \ \Rightarrow \exists q' s' \xrightarrow{a} q'$ t.q. $\langle q, q' \rangle \in R$ et inversement

Automates à multiplicités (nondeterministes) coalgébriquement

- Peuvent être considerés soit comme W-coalgèbre dans la Catégorie *Set* $S, W = \langle o, t \rangle$ avec $o : S \to K$ et $t : S \to (K^S)^A$ avec K^S de support fini
- ou bien comme automates deterministes dans la Catégorie *Vec* des espaces vectoriels et applications lináires (morphismes)
- Les deux traitements conduisent aux différentes notions de bisimulation:
- Bisimulation des automates á multiplicités (le premier cas)
- Bisimulation des automates á multiplicités lináires coïncide avec égalité des séries formelles D'où l'intérêt de linéarisation des automates á multiplicités (déterminisation!)

Existence de coalgébre finale des automates à multiplicités

Proposition. Le foncteur W des automates à multiplicités admet une coalgébre finale.
L'existence de coalgébre finale découle de la bornétude de W.

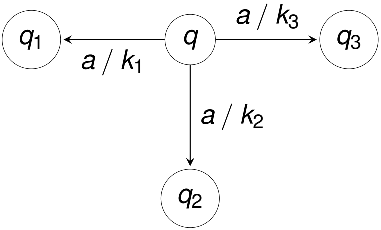
Bornétude du foncteur W signifie que pour tout $W-$ coalgébre $(S, \langle o, t \rangle)$ et pour tout $s \in S$ le nombre de transitionspossibles est borné par un cardinal.

Ici il y seulement un nombre fini de transitions possible, i.e. bornétude par ω.
Proposition. W-Bisimulation (diagramme commutative) coincides with weighted bisimulation above.

Automates à multiplicités linéaires

Définition. Automates à multiplicités linéaires (dans K)sont les coagébres de type automate de Moore, mais sur des spaces vectoriels (dans catégorie **Vec**), soit
$\mathscr{L} = \langle \tilde{o}, \tilde{t} \rangle$, où $\langle \tilde{o}, \tilde{t} \rangle : V \to (K \times V)^A$
Notation: $v_1 \stackrel{a}{\to} v_2$ pour $t(v_1)(a) = v_2$
Comportement: série formelles reconnue par le vecteur $v \in V$ est donnée par: $l(v)(a_1 \dots a_n) = o(v_n)$, où $v \stackrel{a_1}{\to} \cdots \stackrel{a_n}{\to} v_n$.
Linéarisation des automates à multiplicité Pour $G = (S, \langle \tilde{o}, \tilde{t} \rangle)$ nous avons $\tilde{G} = K^S_\omega, \langle \tilde{o}, \tilde{t} \rangle$, où
L'ensemble des états est l'espace vectoriel libre engendré par S, i.e. K^S_ω, $\tilde{o}$ et $\tilde{t}$ sont des linéarisations de o, t, resp.
Plus précisement:
pour $v = k_1 s_1 + \cdots + k_n s_n \in K^S$: $o(v) = k_1.o(s_1) + \cdots + k_n.o(s_n) \in K$ et $\tilde{t}(v)(a) = k_1.t(x_1)(a) + \cdots + k_n.t(x_n)(a) \in K^S_\omega$.

Idée derrière linéarisation



Ces trois transitions sont équivalentes á une seule transition dans $\tilde{G}$:
$q \xrightarrow{a} k_1 q_1 + k_1 q_2 + k_3 q_3,$

où $q \in S$ est un des vecteurs de l'espace libre K_ω^S.

Automate finale à multiplicités linéaires

Automate finale des $\mathscr{L}$-coagèbres est celui des séries formelles K^{A^*},
Pour $d \in K^{A^*}$: $o(d) = d(\varepsilon)$ et $t(d)(a) = d_a$ donnée par:
$d_a(w) = d(aw)$ pour tout $w \in A^*$.
De plus, sa structure est lináire, i.e. o et t le sont!

Proposition. Soient $G = (S, \langle \tilde{o}, \tilde{t} \rangle)$ et $\tilde{G} = K^S, \langle \tilde{o}, \tilde{t} \rangle$ son automate linéarisé. Pour tout $v = k_1 x i_1 + \cdots + k_n x i_n$:
$l(v) = k_1 l_G(x_{i_1}) + \cdots + k_n l_G(x_{i_n})$.
De plus, l est la linéarisation de $l(G) : S \to K^{A^*}$.

$$\begin{tikzcd}
K^S_\omega \arrow[dr, "l"] & \\
S \arrow[u, "v_S"] \arrow[r, "l(G)"] & K^{A^*}
\end{tikzcd}$$

Bisimulation linéaire

R est une relation linéaire si elle est détermiée par son noyau $ker(R) = \{v_1 - v_2 \| \, v_1 R v_2\}$. Si R ne l'est pas, alors sa linéarisation $R^l : \; vRv'$ ssi $v - v' \in span(ker(R))$ l'est.

Définition. Une relation linéaire $R \subseteq V \times V$ est une bisimulation linéaire weighted lin. bisim. si pour tout $\langle v, v' \rangle \in R$:

(i) $o(v) = o(v')$

(ii) $\forall a \in A$, $\langle t(v)(a), t(v')(a) \rangle \in R$

Charactérisation: Bisimulations linéaires sont exactement des noyaux des homomorphismes des automates à multiplicités linéaires!

Proposition. Soit $(V, \langle \tilde{o}, \tilde{t} \rangle)$ un automate linéaire à multiplicités et $l : V \to K^{A^*}$ son unique morphisme final. Alors, $ker(l)$est la plus grande bisimulation linéaire on V.

Corrolaire. L'égalités des séries formelles est la plus grande bisimulation linéaire.

Calcul de bisimulation linéaire

Trois méthodes de calcul effectif:

1. rafinnement itératif de $ker(o)$ par des transitions
2. inverse: rafinnement de complement de $ker(o)$ pour calculer le complement de
3. Téchnique basée sur la construction catégoriale d'une coagèbre finale

Seule la troisième méthode utilisable pour les semi-anneaux (mais elle est la plus abstraite!)

Théorème. Soit $(V, \langle \tilde{o}, \tilde{t} \rangle)$ un automate linéaire à multiplicités et $l : V \to K^{A^*}$ son unique morphisme final. Alors, $l(v) = v \times (I - (\sum_{a \in A})X_a \times \mu(a))^{-1} \times \beta$.
La formule s'obtient en utilisant la théries des "streams" (et solutions d'équations différentielles sur streams)

Outline

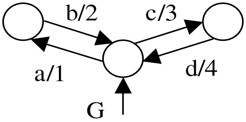
automates (max,+) algébriquement

- (automates (max,+) sont $G = (Q, \alpha, t, \beta)$, avec Q états, $\alpha : Q \to \mathbb{R}_{max}$, $t : Q \times A \times Q \to \mathbb{R}_{max}$, and $\beta : Q \to \mathbb{R}_{max}$, délais initiaux, de transition, et finaux

  Meaning: $t(q, a, q') \in \mathbb{R}_{max}$ correspond à $a-$transition de q versq' et

  $t(q, a, q') = \varepsilon$ si pas de transition de q vers q' avec label a.

Exemple: un automate (max,+)



Automates (max,+) Nonambigus et déterministes

- Automates (max,+) semble être un modele simple, mais leur pouvoir de modélisation est grande, au moins les réseaux de Petri saufs (1-bornés)!
- Automate (max,+) est dit nonambigu, si pour tout mot $w \in A^*$, il y a au maximum un chemin qui a du succès (qui compte pour la s'erie formelle).
- Une série (max,+)-rationelle est dite nonambigue ,s'il existe un automate nonambigu qui la reconnaît.
- Lombardy and Sacharovitch: séries nonambigues sont ecatement des séries à la fois (max,+) et (min,+)-rationel
- Pour les séries rationnelle mais ambigues équalité et inéquality est indécidable!
- Les séries rationnelle mais ambigues peuvent ne pas être déterminisable!
- L'approche coalgébrique de déterminisation basée sur linéarisation peut ne pas terminer!

Automates (max,+) déterministes

$S = (S, t),\ t : S \to (1 + (\mathbb{R}_{\max} \times S))^A$

Un **homomorphisme** entre $S = (S, t)$ et $S' = (S', t')$ est $f:\ S \to S'$ telle que $\forall s \in S$ and $\forall a \in A$: if $s \overset{a|b}{\to} s'$ then $f(s) \overset{a|b}{\to} f(s')$, or equivalently:

$$\begin{tikzcd}
(1 + (\mathbb{R}_{\max} \times S))^A \arrow[d, "F(f)"] & S \arrow[l, "t"'] \arrow[d, "f"'] \\
(1 + (\mathbb{R}_{\max} \times S'))^A & S' \arrow[l, "t'"']
\end{tikzcd}$$

Bisimulation entre $S = (S, t)$ et $S' = (S', t')$ est $R \subseteq S \times S'$ tel que $\forall s \in S$ et $\forall s' \in S'$: si $\langle s, s' \rangle \in R$ alors

(i) $\forall a \in A$: $s \overset{a}{\not\to}$ ssi $s' \overset{a}{\not\to}$

(ii) $\forall a \in A:\ s \overset{a|b}{\to} q\ \Rightarrow\ s' \overset{a|b'}{\to} q'$ t. q.$\langle q, q' \rangle \in R$, et $b = b'$, et

(iii) $\forall a \in A:\ s' \overset{a|b'}{\to} q'\ \Rightarrow\ s \overset{a|b}{\to} q$ t.q. $\langle q, q' \rangle \in R$, et $b = b'$.

Automates (max,+) générales

$S, \langle o, t\rangle$ avec $o : S \to \mathbb{R}_{\max}$ and $t : (\mathbb{R}_{\max}^S)^A$
Notion de bisimulation est différent du précedent tout comme sur autre semi-anneau K:

A **bisimulation** sur $S = (S, t)$ est une relation $R \subseteq S \times R$ t.q. $\forall s \in S$ et $\forall s' \in S$: if $\langle s, s'\rangle \in R$ then

(i) $o(s) = o(t)$

(ii) $\forall a \in A$ and $\forall s'' \in S$

$$\sum_{s'' :\ \langle s'', s'\rangle \in R} t(s)(a)(s'') = \sum_{s'' :\ \langle s'', s'\rangle \in R} t(s')(a)(s'')$$

(iii) $\forall a \in A, \exists q :\ s \overset{a}{\to} q \ \Rightarrow \exists q' s' \overset{a}{\to} q'$ t.q. $\langle q, q'\rangle \in R$ et inversement

Automates (max,+) linéarisation

- Liéairisation répose sur la notion de dimension
- Or, les moduloïdes il n'y a pas la même théorie que pour les modules!
- De plus, la formule pour $I(v)$ ne s'applique pas (pas d'inversion á $\oplus$)
- Il est possible d'utiliser la théorie des séries rationnelles, mais
- Il faut utiliser la thérie des moduloïdes, cf. Butkovic and Cunningham-Green ou travaux de E. Wagneur

Conclusion

- Coalgèbre a connu un progrès enorme
- Coalgèbre est utile pour la théorie des systèmes (automatique théorique)
- Pas seulement les automates sont coalgèbres (systèmes linéaires , graphes, etc.)
- On a des notions canonique de homorphisme et bisimulation
- Automates nondeterministe à multiplicités, leur linéarisation
- Pour appliquer aux séries formelles (max,+) il faut remplacer les notions des anneaux par celles des semi-anneaux