Logique avancée

2 - La logique monadique du second ordre

Master Info Nice Sophia Antipolis
E. Lozes

16. avril 2018

Contenu de la séance

1. Les mots comme structure

2. Logique du premier ordre

3. Logique monadique du second ordre

4. Théorèmes de Trakhtenbrot, Büchi et Rabin

Structure

Un vocabulaire est un ensemble $\tau = \{R_i : a_i\}_{i \in I}$ de symboles de relation R_i avec une arité $a_i \geq 0$.

Une τ-structure est un tuple $\mathcal{U} = (U, \{R^{\mathcal{U}_i}\}_{i \in I})$ où U est un ensemble appelé univers de $\mathcal{U}$, et pour tout i $R_i^{\mathcal{U}} \subseteq U^{a_i}$ est une relation entre a_i éléments de U

Exemples

- $\tau = \{P : 1\}$: un ensemble U et une 2-Partition de U
- $\tau = \{R : 2\}$: un graphe orienté
- $\tau = \{p : 0\}$: un ensemble et un bit

En théorie des bases de données, on peut aussi identifier une base de données relationnelle à une structure : chaque table correspond à une relation dont l'arité est le nombre de colonnes de la table, et l'univers est l'ensemble des valeurs apparaissant dans toutes les cases de toutes les tables.

Formules de la logique du premier ordre

Soit $\mathcal{V}_1 = \{x, y, z, \dots\}$ un ensemble de variables du premier ordre. Les formules de la logique du premier ordre sont définies par récurrence :

- $\top$ (vrai), $\bot$ (faux) sont des formules
- $\varphi \vee \psi$ (φ ou ψ) est une formule
- $\varphi \wedge \psi$ (φ et ψ) est une formule
- $\neg\varphi$ (non φ) est une formule
- $\varphi \Rightarrow \psi$ (φ implique ψ) est une formule
- $\varphi \Leftrightarrow \psi$ (φ ssi ψ) est une formule

- $x = y$ est une formule
- $R(x_1, \dots, x_n)$ est une formule
- $\forall x.\phi$ (pour tout x ϕ) est une formule
- $\exists x.\phi$ (il existe x tel que ϕ) est une formule

où φ, ψ sont déjà des formules

Variables libres, formules closes

Variables libres

Une variable x qui apparait dans une formule sans avoir été introduite par un quantificateur $(\forall, \exists)$ (dans la même sous-formule) est dite *libre*.

exemples : x est libre dans les formules ci-dessous

$$x\,R\,x \qquad \forall y.(x\,R\,y) \qquad \big(\forall x.\forall y.(x\,R\,y)\big) \to x\,R\,x$$

On note $fv(\varphi)$ l'ensemble des variables libres de φ (*free variables*).

Une formule close est une formule sans variables libres

exemples :

$$\forall x.(x\,R\,x) \qquad \forall x, y.(x\,R\,y) \qquad \forall x.\big(\forall y.(x\,R\,y)\big) \to x\,R\,x$$

Propriété de structure

Une formule close exprime une propriété de la structure exemple : la formule

$$\forall x . \exists y . E(x, y)$$

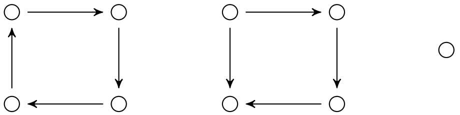
exprime la propriété *"pour tout sommet x du graphe orienté il existe un successeur y de x"*

Une structure $\mathcal{U}$ satisfait (ou pas) une formule close. On dit aussi que $\mathcal{U}$ est un modèle de la formule (ou un contre-modèle).

On note $\mathcal{M} \models \varphi$, et $\models$ est la relation de satisfaction

Exemple

$$\varphi \quad \triangleq \quad \forall x.\exists y.(x \, R \, y)$$



G_1 G_2 G_3 G_4

On a $G_1 \models \varphi$, $G_2 \not\models \varphi$, $G_3 \not\models \varphi$, et $G_4 \models \varphi$.

La formule ci-dessous exprime que le graphe contient au moins 3 sommets distincts.

$$\exists x_1, x_2, x_3. \big(x_1 \neq x_2 \wedge x_2 \neq x_3 \wedge x_3 \neq x_1\big)$$

Formule valide

Une formule close φ est valide $\models \varphi$, lorsque $\mathcal{M} \models \varphi$ pour toute structure $\mathcal{M}$.
Autrement dit φ est valide ssi $\neg\varphi$ n'est pas satisfaisable

Une formule close ψ est une conséquence d'un ensemble $Ax = \{\varphi_i\}_{i \in I}$ de formules closes, $Ax \models \psi$, si pour toute structure $\mathcal{U}$,

$$\text{si} \quad \mathcal{M} \models \varphi_I \quad \text{pour tout } i \in I, \quad \text{alors} \quad \mathcal{M} \models \psi.$$

Deux formules closes φ, ψ sont équivalentes si $\varphi \models \psi$ et $\psi \models \varphi$

	validité	conséquence
exemples	$\models \forall x. x = x$	$\exists x. x = x \ , \ \forall x.(x \, R \, x) \models \exists x. x \, R \, x$

Interprétation

Une interpretation de la formule φ dans la structure $\mathcal{U}$ est une fonction

$$\mathcal{I} \quad : \quad fv(\varphi) \ \rightarrow \ U$$

qui à chaque variable associe un élément de l'univers de la structure.

<u>exemple</u> : $\varphi \quad \triangleq E(x, y)$, et

$$\begin{tikzcd}
x \circ \arrow[r] & \circ\ y \arrow[d] \\
\circ \arrow[u] & \circ \arrow[l]
\end{tikzcd}$$

$$\begin{aligned}
V &= \{v_0, v_1, v_2, v_3\} \\
E &= \{(v_0, v_1), (v_1, v_2), (v_2, v_3), (v_3, v_0)\} \\
\mathcal{I}(x) &= v_0 \\
\mathcal{I}(y) &= v_1
\end{aligned}$$

Second ordre

En programmation fonctionnelle, une fonction est dite d'ordre 2 lorsqu'elle prend en paramètre une fonction d'ordre 1.

par exemple, in Python, la fonction `filter(f,L)` prend en premier paramètre une fonction `f` d'ordre 1, c'est donc une fonction d'ordre 2.

En logique du second ordre, on va pouvoir quantifier sur des formules du premier ordre.

Intuitivement on peut écrire des formules comme

$$\forall\varphi(.,.).\exists x.\varphi(x,x) \quad \text{oder} \quad \forall\varphi : M \to M \to \text{Prop}.\exists x : M.\varphi(x,x)$$

sauf que l'on réserve φ aux ≪ vraies ≫ formules, et on utilise des lettres majuscules pour les *variables du second ordre*. On écrira donc plutôt $\forall X.\exists x.X(x,x)$.

Que veut dire « monadique » ?

un variable du second ordre est monadique si elle est d'arité 1.

monadique

$\forall x . \exists X . \forall y . X(y) \Leftrightarrow x = y$

pas monadique

$\forall x . \exists X . \forall y . X(x, y) \Leftrightarrow x = y$

<u>abréviations</u> : on écrira MSO pour la logique monadique du second ordre et FO pour la logique du premier ordre

Théories

Une théorie est un ensemble T de formules closes tel que pour toute formule close φ, si $T \models \varphi$, alors $\varphi \in T$

Une théorie est décidable si le problème de l'appartenance à T est décidable.

Une théorie T est complète si pour toute formule close φ on a $\varphi \in T$ ou $\neg\varphi \in T$.

La théorie d'une classe $\mathcal{C}$ de structures est l'ensemble des formules closes qui sont valides sur toutes les structures de la classe $\mathcal{C}$

Exemples

La théorie du premier ordre des ordres denses est décidable (on peut éliminer les quantificateurs)

La théorie des graphes est indécidable

Théorème (Büchi) : la théorie monadique du second ordre de $(\mathbb{N}, <)$ est décidable

Théorème (Rabin) : la théorie monadique du second ordre des arbres est décidable

Théorie monadique du second ordre **faible**

Le premier pas vers le théorème de Rabin est le théorème de Büchi-Elgot-Trakhtenbrot.

Théorème : la théorie monadique du second ordre **faible** de $(\mathbb{N}, <)$ est décidable

Autrement dit, étant donnée une formule de MSO sur le vocabulaire $<$, on sait décider si elle est vraie sur toute structure qui est une partie **finie** de $\mathbb{N}$

abréviation : on écrit wMSO pour « weak » MSO

Les mots vus comme des structures

On peut voir le mot $a_1, \ldots, a_n$ comme une structure sur l'univers $\{1, \ldots, n\}$ avec le vocabulaire $\{<: 2\} \cup \{P_a : 1\}_{a \in \Sigma}$, où $<$ est interprété comme l'ordre sur $\{1, \ldots, n\}$ et P_a est l'ensemble des positions où apparait la lettre a

$\Sigma = \boxed{\{a, b, c\}}$ $\qquad$ $w = \boxed{\text{ababc}}$

$\tau = \boxed{\{<: 2, P_a : 1, P_b : 1, P_c : 1\}}$

$$\begin{array}{ccccccccc} \circ & \longrightarrow & \circ & \longrightarrow & \circ & \longrightarrow & \circ & \longrightarrow & \circ \\ P_a & & P_b & & P_a & & P_b & & P_c \end{array}$$

wMSO sur les mots

$\varphi, \psi ::=$	$\mid$	$x < y$	x précède y
	$\mid$	$P_a(x)$	a apparait à la position x
	$\mid$	$X(x)$	X contient la position x
	$\mid$	$\phi \vee \psi$	ϕ ou ψ
	$\mid$	$\neg\phi$	non ϕ
	$\mid$	$\exists x.\phi$	il existe une position x telle que ϕ
	$\mid$	$\exists X.\phi$	il existe un ens. X de positions tel que ϕ

Exemples

1. $ab \models \exists x. P_a(x)$

Exemples

1. $ab \models \exists x.P_a(x)$
2. $ab \not\models \exists x.P_c(x)$

Exemples

1. $ab \models \exists x.P_a(x)$
2. $ab \not\models \exists x.P_c(x)$
3. $ab \not\models \exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$

Exemples

1. $ab \models \exists x. P_a(x)$
2. $ab \not\models \exists x. P_c(x)$
3. $ab \not\models \exists x_1, x_2, x_3. (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
4. $ab \not\models \exists x, y. P_a(x) \wedge P_b(y) \wedge (y < x)$

Exemples

1. $ab \models \exists x.P_a(x)$
2. $ab \not\models \exists x.P_c(x)$
3. $ab \not\models \exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
4. $ab \not\models \exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
5. $ab \models \exists X.\exists x, y.X(x) \wedge \neg X(y)$

Exemples

1. $ab \models \exists x. P_a(x)$
2. $ab \not\models \exists x. P_c(x)$
3. $ab \not\models \exists x_1, x_2, x_3. (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
4. $ab \not\models \exists x, y. P_a(x) \wedge P_b(y) \wedge (y < x)$
5. $ab \models \exists X. \exists x, y. X(x) \wedge \neg X(y)$
6. $ab \not\models \exists X. \forall x. \big(X(x) \Rightarrow P_b(x)\big) \wedge \ \exists x_0. X(x_0) \wedge \neg \exists y. (y < x_0)$

Exemples

1. $ab \models \exists x. P_a(x)$
2. $ab \not\models \exists x. P_c(x)$
3. $ab \not\models \exists x_1, x_2, x_3. (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
4. $ab \not\models \exists x, y. P_a(x) \wedge P_b(y) \wedge (y < x)$
5. $ab \models \exists X. \exists x, y. X(x) \wedge \neg X(y)$
6. $ab \not\models \exists X. \forall x. \big(X(x) \Rightarrow P_b(x)\big) \wedge \underbrace{\exists x_0. X(x_0) \wedge \neg \exists y. (y < x_0)}_{=X(0)}$

Langage d'une formule

Definition

Soit φ une formule de wMSO sur le vocabulaire $\tau := \{P_a\}_{a \in \Sigma}$. Le langage de φ est

$$L(\varphi) \triangleq \{w : w \models \varphi\}.$$

exemples : le langage de $\exists x.P_a(x)$ est $\Sigma^* a \Sigma^*$.

1. $\exists x.P_a(x)$
2. $\exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
3. $\exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
4. $\exists X.\exists x, y.X(x) \wedge \neg X(y)$
5. $\exists X.\forall x.(X(x) \Rightarrow P_b(x)) \wedge X(0)$

1. $\Sigma^* a \Sigma^*$

Langage d'une formule

Definition

Soit φ une formule de wMSO sur le vocabulaire $\tau := \{P_a\}_{a \in \Sigma}$. Le langage de φ est

$$L(\varphi) \triangleq \{w : w \models \varphi\}.$$

exemples : le langage de $\exists x.P_a(x)$ est $\Sigma^* a \Sigma^*$.

1. $\exists x.P_a(x)$
2. $\exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
3. $\exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
4. $\exists X.\exists x, y.X(x) \wedge \neg X(y)$
5. $\exists X.\forall x.(X(x) \Rightarrow P_b(x)) \wedge X(0)$

1. $\Sigma^* a \Sigma^*$
2. $\Sigma\Sigma\Sigma\Sigma^*$

Langage d'une formule

Definition

Soit φ une formule de wMSO sur le vocabulaire $\tau := \{P_a\}_{a\in\Sigma}$. Le langage de φ est
$$L(\varphi) \triangleq \{w : w \models \varphi\}.$$

exemples : le langage de $\exists x.P_a(x)$ est $\Sigma^* a \Sigma^*$.

1. $\exists x.P_a(x)$
2. $\exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
3. $\exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
4. $\exists X.\exists x, y.X(x) \wedge \neg X(y)$
5. $\exists X.\forall x.\big(X(x) \Rightarrow P_b(x)\big) \wedge X(0)$

1. $\Sigma^* a \Sigma^*$
2. $\Sigma\Sigma\Sigma\Sigma^*$
3. $\Sigma^* b \Sigma^* a \Sigma^*$

Langage d'une formule

Definition

Soit φ une formule de wMSO sur le vocabulaire $\tau := \{P_a\}_{a\in\Sigma}$. Le langage de φ est

$$L(\varphi) \triangleq \{w : w \models \varphi\}.$$

exemples : le langage de $\exists x.P_a(x)$ est $\Sigma^* a \Sigma^*$.

1. $\exists x.P_a(x)$
2. $\exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
3. $\exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
4. $\exists X.\exists x, y.X(x) \wedge \neg X(y)$
5. $\exists X.\forall x.\big(X(x) \Rightarrow P_b(x)\big) \wedge X(0)$

1. $\Sigma^* a \Sigma^*$
2. $\Sigma\Sigma\Sigma\Sigma^*$
3. $\Sigma^* b \Sigma^* a \Sigma^*$
4. $\Sigma\Sigma\Sigma^*$

Langage d'une formule

Definition

Soit φ une formule de wMSO sur le vocabulaire $\tau := \{P_a\}_{a\in\Sigma}$. Le langage de φ est

$$L(\varphi) \triangleq \{w : w \models \varphi\}.$$

exemples : le langage de $\exists x.P_a(x)$ est $\Sigma^* a \Sigma^*$.

1. $\exists x.P_a(x)$
2. $\exists x_1, x_2, x_3.(x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_3 \neq x_1)$
3. $\exists x, y.P_a(x) \wedge P_b(y) \wedge (y < x)$
4. $\exists X.\exists x, y.X(x) \wedge \neg X(y)$
5. $\exists X.\forall x.(X(x) \Rightarrow P_b(x)) \wedge X(0)$

1. $\Sigma^* a \Sigma^*$
2. $\Sigma\Sigma\Sigma\Sigma^*$
3. $\Sigma^* b \Sigma^* a \Sigma^*$
4. $\Sigma\Sigma\Sigma^*$
5. $b\Sigma^*$

Théorème de Büchi-Elgot-Trakhtenbrot (2)

théorème : pour tout langage $L \subseteq \Sigma^*$, les trois conditions suivantes sont équivalentes

1. L est régulier
2. L est reconnaissable
3. il existe une formule φ de wMSO telle que $L = L(\varphi)$.

preuve :

- (1)⇔(2) déjà vu
- (2)⇒(3) : à partir d'un automate $\mathcal{A}$ nous devons définir une formule $\varphi_{\mathcal{A}}$ telle que

$$L(\mathcal{A}) = L(\varphi_{\mathcal{A}})$$

- (3)⇒(2) : ...

Théorème de Büchi-Elgot-Trakhtenbrot (2)

théorème : pour tout langage $L \subseteq \Sigma^*$, les trois conditions suivantes sont équivalentes

1. L est régulier
2. L est reconnaissable
3. il existe une formule φ de wMSO telle que $L = L(\varphi)$.

preuve :

- (1)$\Leftrightarrow$(2) déjà vu
- (2)$\Rightarrow$(3) : à partir d'un automate $\mathcal{A}$ nous devons définir une formule $\varphi_{\mathcal{A}}$ telle que

$$L(\mathcal{A}) = L(\varphi_{\mathcal{A}})$$

- (3)$\Rightarrow$(2) : ...

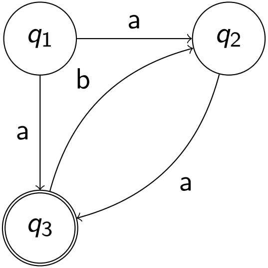
Principe de la construction de $\varphi_{\mathcal{A}}$

$$
\begin{array}{ll}
 & w = a_0 \dots a_{m-1} \in L(\mathcal{A}) \\
\Leftrightarrow & \text{il existe une exécution acceptante} \\
 & q[-1] \xrightarrow{a_0} q[0] \xrightarrow{a_1} \cdots \xrightarrow{a_{m-1}} q[m-1] \\
\Leftrightarrow & \text{pour chaque état } q \in Q \text{ il existe un ensemble } X_q \text{ de positions} \\
 & \text{tel que l'exécution codée par tous les } X_q \text{ est acceptante}
\end{array}
$$

principe du codage :

$$X_q(i) \quad \Leftrightarrow \quad \text{le ième état de l'exécution est} q$$

Exemple



une exécution acceptante du mot aaba

-1	0	1	2	3
	a	a	b	a
q_1	q_2	q_3	q_2	q_3

$$
\begin{aligned}
X_1 &= \emptyset \\
X_2 &= \{0, 2\} \\
X_3 &= \{1, 3\}
\end{aligned}
$$

Définition de $\varphi_{\mathcal{A}}$

Soit $\mathcal{A} = (Q, \Sigma, \Delta, q_I, F)$ un AFN et soit $Q = \{q_1, \ldots, q_n\}$ ses états. Soit $\varphi_{\mathcal{A}}$ la formule :

$$\varphi_{\mathcal{A}} \triangleq \exists X_1, \ldots, X_n. \wedge \varphi_{part} \wedge \varphi_{init} \wedge \varphi_{acc} \wedge \varphi_{\Delta}$$

où

- φ_{part} exprime qu'on a exactement un des $X_1, \ldots, X_n$ à chaque position x

$$\varphi_{part} \triangleq \forall x. \quad \bigvee_{q_i \in Q} X_i(x) \quad \wedge \quad \bigwedge_{q_i, q_j \in Q, q_i \neq q_j} \neg(X_i(x) \wedge X_j)$$

- φ_{init} exprime $q_I \xrightarrow{a_0} q[0]$

$$\varphi_{init} \triangleq \bigwedge_{a \in \Sigma} (P_a(0) \Rightarrow \bigvee_{(q_I, a, q_k) \in \Delta} X_k(0))$$

- φ_{acc} exprime que $q[m-1]$ est un état acceptant

$$\varphi_{acc} \triangleq \bigvee_{q_k \in F} X_k(end)$$

- φ_{Δ} exprime la condition $q[i] \xrightarrow{a_{i+1}} q[i+1]$ pour tout $i = 0, \ldots, m-2$:

$$\varphi_{\Delta} \triangleq \forall x, y.(x + 1 = y) \Rightarrow \bigvee_{(q_i, a, q_j) \in \Delta} X_i(x) \wedge P_a(x) \wedge X_j(y)$$

Théorème de Büchi-Elgot-Trakhtenbrot (2)

théorème : pour tout langage $L \subseteq \Sigma^*$, les trois conditions suivantes sont équivalentes

1. L est régulier
2. L est reconnaissable
3. il existe une formule φ de wMSO telle que $L = L(\varphi)$.

preuve :

- (1)$\Leftrightarrow$(2) déjà vu
- (2)$\Rightarrow$(3) : prouvé !
- (3)$\Rightarrow$(2) : ...

Théorème de Büchi-Elgot-Trakhtenbrot (2)

théorème : pour tout langage $L \subseteq \Sigma^*$, les trois conditions suivantes sont équivalentes

1. L est régulier
2. L est reconnaissable
3. il existe une formule φ de wMSO telle que $L = L(\varphi)$.

preuve :

- (1)$\Leftrightarrow$(2) déjà vu
- (2)$\Rightarrow$(3) : prouvé !
- (3)$\Rightarrow$(2) : ...

Codage d'une interprétation comme mot

Étant donnée une interprétation $\mathcal{I}$ de n variables monadiques $X_1, \ldots, X_n$ et de m variables du premier ordre $x_1, \ldots, x_m$ dans une structure $\mathcal{U}$ d'univers $\{0, \ldots, \ell - 1\}$, ainsi qu'un ordre $\prec$ entre ces variables, on définit le mot $w(\prec, \mathcal{U}, \mathcal{I})$ comme le mot sur l'alphabet $\{0, 1\}^{n+m}$ tel que

- $|w| = \ell$: le mot a la même longueur que l'univers
- à chaque position i, la lettre $w[i]$ de w est un vecteur formé de 0 et de 1 dont la $k$ième valeur $w[k][i]$ vaut 1 ssi la k-ième variable (au sens de $\prec$) est « vraie » à la position i

Exemple

$$\mathcal{M},\mathcal{I} \qquad \underset{X}{\circ} \to \circ \longrightarrow \underset{X,\,Y}{\circ} \longrightarrow \underset{Y}{\circ} \longrightarrow \underset{\text{x,y}}{\circ} \to \circ$$

	0	1	2	3	4	5
X	1	0	1	0	0	0
Y	0	0	1	1	0	0
x	0	0	0	0	1	0
y	0	0	0	0	1	0

$$w(\prec,\mathcal{M},\mathcal{I}) \triangleq \boxed{\begin{matrix}1\\0\\0\\0\end{matrix}}\boxed{\begin{matrix}0\\0\\0\\0\end{matrix}}\boxed{\begin{matrix}1\\1\\0\\0\end{matrix}}\boxed{\begin{matrix}0\\1\\0\\0\end{matrix}}\boxed{\begin{matrix}0\\0\\1\\1\end{matrix}}\boxed{\begin{matrix}0\\0\\0\\0\end{matrix}} \in \Sigma^*, \text{ où } \Sigma = \{0,1\}^4$$

Ordre $\prec \; = X \prec Y \prec x \prec y$

Le $\prec$-langage d'une formule

Definition

Soit φ une formule et $\prec$ un ordre total sur $fv(\varphi)$.

$$L(\prec,\varphi) \triangleq \{w(\prec,\mathcal{M},\mathcal{I}) : \mathcal{M},\mathcal{I} \models \varphi\}$$

exemples :

- φ est la formule $x < y$, et $\prec$ est tel que $x \prec y$.
  Alors $L(\prec,\varphi) = \binom{0}{0}^*.\binom{1}{0}.\binom{0}{0}^*.\binom{0}{1}.\binom{0}{0}^*$
- φ est la formule $X(x)$, et $\prec$ est tel que $X \prec x$.
  Alors $L(\prec,\varphi) = (\binom{0}{0}+\binom{1}{0})^*.\binom{1}{1}(\binom{0}{0}+\binom{1}{0})^*$

Preuve du théorème

Lemme 1

Soit φ une formule et $\prec$ un ordre total sur $fv(\varphi) \cup \tau$.
Alors $L(\prec, \varphi)$ est un langage régulier.

Lemme 2

Soit φ une formule close.
Alors $L(\varphi) = h(L(\prec, \varphi))$, où $h : \{0,1\}^{|\Sigma|} \to \Sigma$ est un homomorphisme

Preuve du théorème

Lemme 1

Soit φ une formule et $\prec$ un ordre total sur $fv(\varphi) \cup \tau$.
Alors $L(\prec, \varphi)$ est un langage régulier.

Lemme 2

Soit φ une formule close.
Alors $L(\varphi) = h(L(\prec, \varphi))$, où $h : \{0,1\}^{|\Sigma|} \to \Sigma$ est un homomorphisme

Les langages réguliers étant clos par intersection et homomorphisme, ces deux lemmes complètent la preuve du théorème de Büchi-Elgot-Trakhtenbrot.

Preuve du Lemme 1

Lemme 1

Soit φ une formule et $\prec$ un ordre total sur $fv(\varphi)$. Alors $L(\prec, \varphi)$ est un langage régulier

La preuve se fait par induction sur φ :

- si φ est la formule $x < y$ ou $P_a(x)$ ou $X(x)$, nous avons déjà vu que $L(\prec, \varphi)$ est régulier
- si φ est la formule $\varphi_1 \vee \varphi_2$: alors $L(\prec, \varphi)$ est l'union $L(\prec, \varphi_1) \cup L(\prec, \varphi_2)$
- si $\varphi = \neg\varphi'$, alors $L(\prec, \varphi) = (\Sigma^* \backslash L(\prec, \varphi')) \cap L_0$, où L_0 est le langage des mots de la forme $w(\prec, w, \mathcal{I})$.

  i.e. : les variables du premier ordre sont vraie exactement à une position, et à chaque position on a exactement un P_a de vrai
- si $\varphi = \exists x.\varphi'$, $L(\varphi)$ est le langage $h(L(\varphi'))$, où h est l'homomorphisme qui efface la ligne de x
- si $\varphi = \exists X.\varphi'$: $L(\varphi)$ est le langage $h(L(\varphi'))$, où h est l'homomorphisme qui efface la ligne de X

Décidabilité de wMSO sur les mots

Corollaire : Le problème de satisfaisabilité

- Etant donné : φ
- a-t-on un mot w tel que $w \models \varphi$?

est décidable.

Corollaire : Le problème de Model-Checking est décidable :

- Étant donnés w, φ
- a-t-on $w \models \varphi$?

preuve :

- φ est satisfaisable ssi $L(\varphi) \neq \emptyset$
- $w \models \varphi$ ssi $w \in L(\varphi)$