Logique Monadique du Second Ordre et Automates

January 5, 2011

La logique monadique du second ordre est la logique du premier ordre à laquelle on a rajouté le quantificateur sur les ensembles $\exists X$ et la relation $\in$. Plus formellement, les MSO($\mathcal{L}$) sont les plus petits ensembles de formules vérifiant :

- Pour toute relation d'arité $r \in \mathbb{N}$ $R \in \mathcal{L}$, pour toutes constantes du premier ordre $c_1, c_2, ..., c_r \in \mathcal{L}$, la formule "$Rc_1c_2...c_r$" $\in MSO(\mathcal{L})$
- Pour toutes constante du premier ordre $c \in \mathcal{L}$ et du second ordre $C \in \mathcal{L}$, la formule "$c \in C$" $\in MSO(\mathcal{L})$
- $\forall$"ϕ" $\in MSO(\mathcal{L})$,"$\neg\phi$" $\in MSO(\mathcal{L})$
- $\forall$"ϕ","ψ" $\in MSO(\mathcal{L})$,"$\phi \wedge \psi$" $\in MSO(\mathcal{L})$
- $\forall$"ϕ" $\in MSO(\mathcal{L} \cup \{c\})$,"$\exists x\phi[c/x]$" $\in MSO(\mathcal{L})$ où c est une constante du premier ordre dans $\mathcal{L} \cup \{c\}$, et x une variable du premier ordre dans $\mathcal{L}$.
- $\forall$"ϕ" $\in MSO(\mathcal{L} \cup \{C\})$,"$\exists X\phi[C/X]$" $\in MSO(\mathcal{L})$ où C est une constante du second ordre dans $\mathcal{L} \cup \{C\}$, et X une variable du second ordre dans $\mathcal{L}$.

On rajoute souvent des opérateurs logique supplémentaires, comme " $\vee$ ", "$\exists$", " $\Longrightarrow$ ","$\Longleftrightarrow$ "

exemple : "$\exists X[\forall x \forall y[(\forall z(z \leq x \vee y \leq z) \wedge x < y) \implies (x \in X \iff \neg y \in X)] \wedge \exists x(\forall y, x \leq y \wedge x \in X)] \wedge \forall x, R_a(x)$" $\in MSO(\{R_a\})$où R_a est un prédicat unaire.

Soit $\mathcal{M}$ une $\mathcal{L}$-structure. On dit que $\mathcal{M}$ est un modèle de Φ (noté $\mathcal{M} \models \Phi$) ssi $\mathcal{M}$ satisfait Φ. Nous renvoyons au cours de Logique pour une définition rigoureuse.

exemple : Soit $\mathcal{M}$ une $\{<, R_a, R_b\}$-structure totalement ordonnée par $<$, de cardinal fini et pair, et telle que les seules lettres soient $\{a\}$. Alors elle satisfera la formule ci-dessus. (X sera l'ensemble des éléments en position impaire, en commençant à 1.)

Nous chercherons à démontrer le théorème suivant :

Un langage propre est reconnu par un automate ssi c'est l'ensemble des $\{R_{a_1}, ..., R_{a_k}, <\}$-structures finies totalement ordonnées partitionées par les prédicats unaires $R_{a_1}, ..., R_{a_k}$ qui satisfont une formule de MSO.

Part I
Sens direct :

L'objectif de cette partie est de trouver pour chaque automate (Σ, Q, I, F, T) reconnaissant un langage propre une MSO-formule qui décrit le langage propre reconnu par cet automate.

On va écrire une formule indiquant qu'il existe une partition du domaine en $|T|$ ensembles

$$\exists X_1, \exists X_2, ..., \exists X_T \forall x[(x \in X_1 \wedge x \notin X_2 \wedge ... \wedge x \notin X_T) \vee ... \vee (x \notin X_1 \wedge ... \wedge x \notin X_{T-1} \wedge x \in X_T)]$$

telle que les lettres lues sont les bonnes,

$$\bigwedge \forall x[(x \in X_1 \implies R_{a_1}(x)) \wedge ... \wedge (x \in X_T \implies R_{a_T}(x))]$$

que deux transitions consécutives sont compatibles,

$$\bigwedge \forall x[\forall y(x < y \wedge \forall z \neg(x < z \wedge z < y)) \implies ((x \in X_i \wedge y \in X_j) \vee ... \vee (x \in X_k \wedge y \in X_l))]$$

que la première part de I

$$\bigwedge \forall x[(\forall y \neg y < x) \implies (x \in X_{i_1} \vee ... \vee x \in X_{i_I})]$$

et que la dernière arrive dans F.

$$\bigwedge \forall x[(\forall y \neg x < y) \implies (x \in X_{f_1} \vee ... \vee x \in X_{f_F})]$$

On vérifie aisément que tout mot accepté par l'automate considéré satisfait la formule et que tout mot satisfaisant la formule est accepté par l'automate.

Exemple : automate à deux états, mots de longueur paire constitué de a.

$$\exists X_1, \exists X_2, \forall x[(x \in X_1 \wedge x \notin X_2) \vee (x \notin X_1 \wedge x \in X_2)]$$

$$\bigwedge \forall x[(x \in X_1 \implies R_a(x)) \wedge (x \in X_2 \implies R_a(x))]$$

$$\bigwedge \forall x[\forall y(x < y \wedge \forall z \neg(x < z \wedge z < y)) \implies ((x \in X_1 \wedge y \in X_2) \vee (x \in X_2 \wedge y \in X_1))]$$

$$\bigwedge \forall x[(\forall y \neg y < x) \implies x \in X_1]$$

$$\bigwedge \forall x[(\forall y \neg x < y) \implies (x \in X_1)]$$

Part II
Jeu d'Ehrenfeucht-Fraisse : un jeu sur des structures.

Dans toute cette partie, les langages sont finis. Les structures peuvent être infinies ou finies. La logique est FO (ou MSO).

On va définir un jeu à deux joueurs, que l'on va nommer conventionnellement Spoiler et Duplicator.

Matériel : deux $\mathcal{L}$-structures $\mathcal{A}$ et $\mathcal{B}$, un entier n. On note ce jeu $G_n^{\mathcal{L}}(\mathcal{A}, \mathcal{B})$

But du jeu : pour Spoiler, démontrer la différence des structures; pour Duplicator, démontrer leur similarité.

Déroulement : Le jeu va durer n tours.

À chaque tour, Spoiler commence. Il choisit un élément (ou une partie) de $\mathcal{A}$ ou $\mathcal{B}$, puis Duplicator choisit respectivement un élément (ou une partie) de $\mathcal{B}$ ou $\mathcal{A}$.

À la fin, cela nous donne une suite $(\alpha_i)_{i\in[1,n]}$ d'éléments (et de parties) de $\mathcal{A}$ et une suite $(\beta_i)_{i\in[1,n]}$ d'éléments (et de parties) de $\mathcal{B}$.

Victoire : Duplicator gagne ssi les deux structures satisfont les mêmes relations sans quantificateur dans le langage $\mathcal{L} \cup \{\gamma_i | i \in [1, n]\}$, les γ_i étant interprétés par les α_i dans $\mathcal{A}$ et par les β_i dans $\mathcal{B}$. (Dans un langage sans symbole de constante, cela veut dire que pour toute relation R m-aire de $\mathcal{L}$ (l'égalité par exemple), $R\alpha_{i_1}\alpha_{i_2}...\alpha_{i_m} \Leftrightarrow R\beta_{i_1}\beta_{i_2}...\beta_{i_m}$.) Spoiler gagne si ce n'est pas le cas.

On remarque que pour tout $n \in \mathbb{N}$ et tout $\mathcal{L}$ la relation entre structures "Duplicator a une stratégie gagnante à $G_n^{\mathcal{L}}(\mathcal{A}, \mathcal{B})$", que l'on notera $\mathcal{A} \sim_n^{\mathcal{L}} \mathcal{B}$, est reflexive (Duplicator joue les mêmes coups que Spoiler.), symétrique et transitive (si Spoiler joue dans $\mathcal{A}$, Duplicator joue dans C comme il jouerait face à son propre coup dans $\mathcal{B}$ qu'il jouerai face au coup de Spoiler dans $\mathcal{A}$, et vice-versa). C'est donc une relation d'équivalence entre $\mathcal{L}$-structures.

On va démontrer dans cette partie que cette relation d'équivalence définit un ensemble fini de classes d'équivalence, peut s'exprimer en termes logiques, et se comporte bien avec la concaténation.

On définit par induction la longueur d'une formule :

- $lg(\phi) = 1$ où ϕ est sans connecteur logique, quantificateur ou négation.
- $lg(\exists x\phi) = 1 + lg(\phi)$
- $(lg(\exists X\phi) = 1 + lg(\phi))$
- $lg(\neg\phi) = 1 + lg(\phi)$

- $lg(\phi \wedge \psi) = 1 + lg(\phi) + lg(\psi)$

On définit par induction la hauteur d'une formule :

- $ht(\phi) = 0$ où ϕ est sans connecteur logique, quantificateur ou négation.
- $ht(\exists x\phi) = 1 + ht(\phi)$
- $(ht(\exists X\phi) = 1 + ht(\phi))$
- $ht(\neg\phi) = ht(\phi)$
- $ht(\phi \wedge \psi) = max(ht(\phi), ht(\psi))$

On définit la relation entre $\mathcal{L}$-structures $\equiv_n^{\mathcal{L}}$ par$\mathcal{A} \equiv_n^{\mathcal{L}} \mathcal{B}$ ssi $\mathcal{A}$ et $\mathcal{B}$ satisfont les memes $\mathcal{L}$-énoncés de hauteur inférieure ou égale à n.

1 Le nombre de classes d'équivalence est fini.

Démontrons-le.

Par récurrence sur n, montrons que pour tout $\mathcal{L}$, le nombre de $\mathcal{L}$-énoncés de hauteur $\leq n$ à équivalence près est fini.

Pour $n = 0$, $\mathcal{L}$ étant fini, l'arité des relations de $\mathcal{L}$ est majorée par un $N \in \mathbb{N}$, le nombre de relations de $\mathcal{L}$ par $|\mathcal{L}|$ et le nombre de constantes de $\mathcal{L}$ par $|\mathcal{L}|$.

Un énoncé de hauteur 0 est une fonction logique d'énoncés sans connecteur logique, quantification ou négation. On a donc une borne de $|\mathcal{L}|^{|R|+1}$ sur le nombre d'énoncés sans connecteur logique, quantification, ou négation; et un majorant de $2^{|\mathcal{L}|^{|R|+1}}$ sur le nombre d'énoncés de hauteur 0.

Pour $n > 0$, soit K un majorant du nombre de $\mathcal{L} \cup \{c\}$-énoncés de hauteur $n-1$ et de $\mathcal{L} \cup \{C\}$-énoncés de hauteur $n-1$ (par hypothèse de récurrence.

Un $\mathcal{L}$-énoncé de hauteur $\leq n$ est toujours équivalent à un énoncé qui est une fonction logique de $\mathcal{L}$-énoncés de la forme $\exists x\phi$ où ϕ est un $\mathcal{L} \cup \{x\}$-énoncé de hauteur $n-1$ ou de la forme $\exists x\phi$ où ϕ est un $\mathcal{L} \cup \{x\}$-énoncé de hauteur $n-1$.

On peut donc majorer par $2^{2K} = 4^K$ le nombre de $\mathcal{L}$-énoncés de hauteur $\leq n$ à équivalence près.

La section suivante nous permet de conclure.

2 Cette relation s'exprime en termes logiques.

Démontrons que $\mathcal{A} \sim_n^{\mathcal{L}} \mathcal{B} \Leftrightarrow \mathcal{A} \equiv_n^{\mathcal{L}} \mathcal{B}$

Sens direct par récurrence sur n :

Pour $n = 0$, c'est évident.

Soit $n > 0$:

Supposons $\mathcal{A} \not\equiv_n^{\mathcal{L}} \mathcal{B}$. Soit Φ un $\mathcal{L}$-énoncé minimal pour la longueur tel que $ht(\Phi) \leq n$ dont la valeur de vérité n'est pas la même dans $\mathcal{A}$ que dans $\mathcal{B}$. Il est soit sans connecteur logique, quantificateur ou négation, soit de la forme $\exists x\phi(x)$

(ou $\exists X\phi(X)$), où $ht(\phi) \leq n-1$. Dans la premier cas, Duplicator a perdu, et n'a donc pas de stratégie gagnante.

Dans le second cas, nous pouvons supposer par symétrie que $\mathcal{A} \models \exists x\phi(x)$ et $\mathcal{B} \not\models \exists x\phi(x)$ (ou $\mathcal{A} \models \exists X\phi(X)$ et $\mathcal{B} \not\models \exists X\phi(X)$, respectivement) , donc $\mathcal{B} \models \forall x\neg\phi(x)$ ($\mathcal{B} \models \forall X\neg\phi(X)$).

Spoiler choisit un u (un U) de A tel que $\phi(u)$ ($\phi(U)$). Alors pour tout v (V) de B choisi par Duplicator, dans le langage $\mathcal{L}' = \mathcal{L} \cup \{c\}$ ($\mathcal{L}' = \mathcal{L} \cup \{C\}$) et les $\mathcal{L}'$-structures $\mathcal{A}'$ et $\mathcal{B}'$ qui sont les $\mathcal{L}$-structures $\mathcal{A}$ et $\mathcal{B}$ étendues au langage $\mathcal{L}'$ où c_1 (C_1) est interprété dans $\mathcal{A}'$ par u (U) de A et dans $\mathcal{B}'$ par v (V) de B, on aura pour le $\mathcal{L}'$-énoncé $\phi(c)$ ($\phi(C)$) de hauteur $\leq n-1$ $\mathcal{A}' \models \phi(c)$ et $\mathcal{B}' \not\models \phi(c)$ car $\mathcal{B} \models \forall x\neg\phi(x)$ ($\mathcal{A}' \models \phi(C)$ et $\mathcal{B}' \not\models \phi(C)$ car $\mathcal{B} \models \forall X\neg\phi(X)$), donc $\mathcal{A}' \not\equiv^{\mathcal{L}'}_{n-1} \mathcal{B}'$, donc $\mathcal{A}' \not\sim^{\mathcal{L}'}_{n-1} \mathcal{B}'$ par récurrence, d'où $\mathcal{A} \not\sim^{\mathcal{L}}_{n} \mathcal{B}$, car en conséquence Duplicator n'a pas de stratégie gagnante après le coup u (U) donc n'a pas de stratégie gagnante tout court.

Sens indirect par récurrence sur n :

Pour $n = 0$, c'est évident.

Soit $n > 0$:

Supposons $\mathcal{A} \not\sim^{\mathcal{L}}_{n} \mathcal{B}$. Comme le nombre de coups est fini, Spoiler a une stratégie gagnante. Par symétrie, supposons que ce soit en jouant son premier coup dans A. Quelquesoit la réponse de Duplicator, on a $\mathcal{A}' \not\sim^{\mathcal{L}'}_{n-1} \mathcal{B}'$, $\mathcal{A}$' et $\mathcal{B}$' étant ceux introduits dans la récurrence précédente. Par hypothèse de récurrence, $\mathcal{A}' \not\equiv^{\mathcal{L}'}_{n-1} \mathcal{B}'$. On a donc que le formule suivante, potentiellement infinie, n'a pas la même valeur de vérité dans $\mathcal{A}$ et dans $\mathcal{B}$: $\exists x \bigwedge_{b\in B} f_{\mathcal{A}' \not\equiv^{\mathcal{L}'}_{n-1} \mathcal{B}'}(b)$ où $f_{\mathcal{A}' \not\equiv^{\mathcal{L}'}_{n-1} \mathcal{B}'}(b)$ est un $\mathcal{L}'$-énoncé de hauteur $n-1$ vrai dans $\mathcal{A}'$ mais pas dans $\mathcal{B}'$, sachant que duplicator a joué b. Comme le nombre de $\mathcal{L}'$- énoncés de hauteur $n-1$ est fini, on peut ramener cette conjonction a une conjonction finie. Donc $\mathcal{A} \not\equiv^{\mathcal{L}}_{n} \mathcal{B}$.

3 Elle réagit bien avec la concaténation

Nous allons montrer maintenant que dans un langage sans constantes, si $\mathcal{A}_1 \sim^{\mathcal{L}}_{n} \mathcal{B}_1$ et $\mathcal{A}_2 \sim^{\mathcal{L}}_{n} \mathcal{B}_2$, alors $\mathcal{A}_1 \cdot \mathcal{A}_2 \sim^{\mathcal{L}}_{n} \mathcal{B}_1 \cdot \mathcal{B}_2$. La concaténation est définie comme on le pense au niveau du domaine, des interprétations de l'ordre et des prédicats unaires. Pour les relations d'arité supérieure autre que l'ordre, leur interprétation dans $\mathcal{A}_1 \cdot \mathcal{A}_2$ est l'union de leurs interprétations dans $\mathcal{A}_1$ et $\mathcal{A}_2$ (On n'a donc jamais dans $\mathcal{A}_1 \cdot \mathcal{A}_2$ $Ra_1a_2...a_r$ si certains a_i sont dans A_1 et d'autres dans A_2 avec R d'arité supérieure à 2 et différente de l'ordre.).

C'est simple : si Spoiler joue un élément a de A_1, alors duplicator joue dans B_1 comme une de ses stratégies gagnantes $\mathcal{A}_1 \sim^{\mathcal{L}}_{n} \mathcal{B}_1$. De même si Spoiler joue dans B_1, A_2 ou B_2. (Si Spoiler joue une partie P, alors duplicator joue l'union des parties P_1 et P_2 qu'il jouerait respectivement dans B_1 et dans B_2 contre respectivement $P \cap A_1$ et $P \cap A_2$.) On vérifie aisément que Duplicator gagne ainsi.

Part III

Sens indirect :

On a $\mathcal{L} = \{<, R_a\}$. On a ϕ un énoncé de $MSO(L)$. On veut prouver que l'ensemble des modèles finis de ϕ est un langage rationnel. Soit n la hauteur de ϕ.

Pour cela, nous allons construire un automate deterministe dont les états sont les classes d'équivalence de $\equiv_n^{\mathcal{L}}$. C'est possible, car elles forment un ensemble fini. Comme $\equiv_n^{\mathcal{L}}$ réagit bien avec la concaténation, les transitions peuvent êtres définies par $\forall u \in A^*, \forall a \in A, \bar{u}.a = \bar{u.a}$. L'état initial sera $\bar{\epsilon}$. L'ensemble des états finaux sera l'ensemble des classes d'équivalence de $\equiv_n^{\mathcal{L}}$ telles qu'il existe dans cette classe un modèle de ϕ. De part la définition de $\equiv_n^{\mathcal{L}}$, et la hauteur n de ϕ, tout élément de ces classes satisfait ϕ.

On vérifie aisément par récurrence sur la longueur du mot, grace à la définition des transitions de l'automate que $\forall u \in A^*, \bar{\epsilon}.u = \bar{u}$.

Donc cet automate fonctionne, il accepte tous les mots satisfaisant ϕ.

Conclusion

On peut interpréter ce résultat comme témoignant de la simplicité de MSO. Elle ne permet de décrire que des langages rationnels et, de part notre partie 1, elle s'effondre en $\exists X \forall x$-MSO, l'ensemble des énoncés de MSO en forme "normale " avec des quantificateurs existentiels devant les ensembles et universels devant les variables.

D'autres logiques ont été inventées pour traiter de langages plus complexes.

- SO (logique du second ordre) décrit PH
- HO (logique d'ordre quelconque : on peut quantifier par exemple sur $P^{12}(P^{27}(P^{142857}(P(P^{1234567890}(P^{42}(P(E)))))))$ décrit Élémentaire.
- TC (logique du premier ordre auquel on a ajouté un opérateur de clôture transitive) décrit NL
- STC (ajout d'un opérateur de clôture transitive symétrique) décrit SL, donc L (car le problème de l'accessibilité dans un graphe non-orienté, qui est SL-complet, est dans L).
- IFP (ajout d'un opérateur de point fixe pour des fonctions logiques pour lesquelles $X \subseteq f(X)$) décrit P
- FP (ajout d'un opérateur de point fixe général) décrit PSPACE