Équivalence entre Logique Monadique du Second Ordre et Automate ou Le théorème de Büchi

Rémy Tuyéras

2 janvier 2009

1 Logique Monadique du Second Ordre

Comme l'intitulé de cet article le laisse deviner au lecteur, notre propos touche à des concepts découlant aussi bien de la Théorie des automates que celle de la Logique Monadique du Second Ordre. Nous supposerons dans la suite les notions propres aux théories des Automates et de la Logique du Première Ordre connues. Pour ce qui est de la Logique du Second Ordre, nous ne nous étalerons pas davantage sur le sujet que dans l'énoncer de quelques définitions et lemmes utilent à notre exposé. Pour définir la Logique Monadique du Second Ordre, nous devons avant tout rappeler la définition de la Logique du Second Ordre :

Définition 1 (Logique du second Ordre)**.** *La définition de la logique du second ordre (SO) étend la définition de la logique de premier ordre aux variables de sous-ensembles et de relations sur l'univers et autorise à quantifier sur celles-ci.*

Nous en venons directement à la définition qui nous intéresse :

Définition 2. *La logique Monadique du Second Ordre, ou MSO, est définit comme la restriction de la logique du Second Ordre aux variables SO d'arité 1.*

Dans ce qui suit, nous userons souventefois de la notion de *rang de quantification*. Son utilisation sera d'avantage justifiée par le fait qu'elle soit nécessaire aux théorèmes que nous manipulerons plutôt qu'elle fournisse quelque formalisme directement exploitable pour notre sujet.

Définition 3 (rang de quantification)**.** *Le rang de quantification $qr(\varphi)$ d'une formule φ est la profondeur de quantification absolue, c'est-à-dire :*

- *Si φ est atomique, alors $qr(\varphi) = 0$;*
- $qr(\varphi_1 \vee \varphi_2) = qr(\varphi_1 \wedge \varphi_2) = max(qr(\varphi_1), qr(\varphi_2))$;
- $qr(\neg\varphi) = qr(\varphi)$;
- $qr(\exists x\varphi) = qr(\forall x\varphi) = qr(\varphi) + 1$.

De la notion de rang découle la définition et le lemme suivants :

Définition 4. *On définit MSO[k] comme étant l'ensemble des formules de rang de quantification k. On appelle alors type l'ensemble des formules suivantes :*

$$mso\text{-}tp_k(\mathfrak{A}, \overrightarrow{x}, \overrightarrow{X}) = \{\varphi(\overrightarrow{x}, \overrightarrow{X}) \in MSO[k] \mid \mathfrak{A} \vDash \varphi(\overrightarrow{x}, \overrightarrow{X})\}$$

Cette définition est utile dans le fait qu'elle nous permettra de comparer les structures entre elles sur ce que nous pourrions appeller leur base consistante (nous prendrons alors $\overrightarrow{x}$ et $\overrightarrow{X}$ nuls). Le lemme qui suit nous permettra de manipuler les types entre eux tel qu'il est necessaire dans le théorème de Büchi (avec la convention $\text{mso-tp}_k(\mathfrak{N}) = \text{mso-tp}_k(\mathfrak{M}) \Leftrightarrow \mathfrak{N} \equiv_k^{\text{MSO}} \mathfrak{M}$).

Lemme 1. *Soit $\mathfrak{A}_1$,$\mathfrak{A}_2$,$\mathfrak{B}_1$,$\mathfrak{B}_2$ des σ-structures, et soit $\mathfrak{A}$ l'union disjointe de $\mathfrak{A}_1$ et $\mathfrak{A}_2$, et $\mathfrak{B}$ l'union disjointe de $\mathfrak{B}_1$ et $\mathfrak{B}_2$. Supposons $\mathfrak{A}_1 \equiv_k^{MSO} \mathfrak{B}_1$ et $\mathfrak{A}_2 \equiv_k^{MSO} \mathfrak{B}_2$. Alors $\mathfrak{A} \equiv_k^{MSO} \mathfrak{B}$*

Ayant introduit les quelques définitions nécessaires à l'utilisation des notions de la Logique Monadique du Second Ordre, nous pouvons passer à la description du formalisme sur lequel nous nous appuierons tout au long du dévellopement.

2 Préliminaires et Formalisme

2.1 Idée générale

Notre exposé a pour but de redémontrer un Théorème dû à Büchi faisant le lien entre la Théorie des Langages Rationnels et la Logique Monadique du Second Ordre (MSO). Dans cette

section nous introduisont le formalisme nécessaire à l'énoncé du théorème. Il consiste en particulier à exposé une construction permettant d'associer les mots d'un langage à des structures de la logique MSO. L'idée est de voir qu'une chaine de caractères peut s'écrire de manière mathématique à l'aide d'une relation d'ordre et d'ensembles, pour respectivement traduire l'enchaînement des caractères dans le mot et pouvoir identifier un caractère en terme d'appartenance à tel ou tel ensemble. De là, nous pouvons facilement construire un objet mathématique appartenant à la logique MSO, comme par exemple les structures, qui ne sont en quelque sorte qu'une association de relations, éventuellement d'opérations et de sous-ensembles de l'univers.

2.2 Illustration

Éclaircissons la construction précédement suggéré avec le mot $s = abaab$. Comme nous l'avons dit plus haut, lors de la traduction mathématique, il faut pouvoir conserver l'ordre sur les caractères et en même temps pouvoir reconnaitre le caractère auquel est associé un élément mathématique de la structure. Une première idée serait d'associer chaque caractère à son indice dans le mot et construire des classes rassemblant les indices représentant un même caractère. Par exemple, nous pouvons associer au mot $s = abaab$ la structure suivante

$$\langle \{1, 2, 3, 4, 5\}, <, P_a, P_b \rangle \text{ avec } P_a = \{1, 3, 4\} \text{ et } P_b = \{2, 5\}$$

où l'ensemble des indices $\{1, 2, 3, 4, 5\}$ est l'univers de notre structure et P_a et P_b sont les classes d'indices respectivement pour les caractères a et b. De cette manière, le fait que $1 < 4$ traduit bien le fait que le premier caractère soit avant le quatrième caractère dans le mot. De même le fait que $P_a(3)$ traduit bien le fait que le troisième caractère dans le mot soit un a.

2.3 Généralisation

De manière plus formel, pour un alphabet fini Σ, nous définissons le langage σ_Σ qui contient la relation binaire $<$ et la relation unaire P_a pour tout $a \in \Sigma$. Ainsi, un mot $s \in \Sigma^*$ de longueur n est représenté par une structure $M_s \in STRUCT[\sigma_\Sigma]$ dont l'univers est $\{1, 2, \ldots, n\}$, avec $<$ interprété comme l'ordre sur les entiers naturels et P_a étant l'ensemble des positions où la lettre a apparait, pour a dans Σ.

2.4 La $\mathcal{L}$-définissabilité

Nous introduisons ici la notion de $\mathcal{L}$-définissabilité, dernière étape avant de pouvoir énoncer le Théorème de Büchi. Soit ϕ un énoncé d'une logique $\mathcal{L}$, écrit dans un vocabulaire σ_Σ. Nous pouvons alors définir à partir de cette énoncé un *langage*, c'est-à-dire un sous-ensemble de Σ^*, donné par

$$L(\phi) = \{s \in \Sigma^* \mid M_s \vDash \phi\}$$

Nous dirons qu'un langage L est définissable dans une logique $\mathcal{L}$ s'il existe un $\mathcal{L}$-énoncé ϕ tel que $L = L(\phi)$.

3 Théorème de Büchi

Le théorème suivant, qui est le sujet principal de notre propos, est un résultat fondamental qui relie la MSO-définissabilité et les langages rationnels.

Théorème 1 (Büchi)**.** *Un langage est MSO-définissable si, et seulement si, il est rationnel.*

Preuve . $\boxed{\Leftarrow}$ Soit L un langage rationnel. Notre objectif va être de trouver une formule ϕ appartenant à la logique MSO tel que $L = L(\phi)$. De manière moins formel, nous voulons trouver un énoncé proche de la logique du premier ordre dont la particularité est d'user de sous-ensembles de l'univers et vérifiant $M_s \vDash \phi \Leftrightarrow s \in L$.

La possibilité de trouver un énoncé qui caractérise L demande que nous sachions décrire l'ensemble des propriétés de L par un ensemble fini d'énoncé sans autre hypothèse qu'il est rationnel, ce qui n'est pas des plus évident si nous restons dans le formalisme des langages rationnels. Or, le fait que L soit rationnel implique qu'il soit reconnaissable par un automate fini déterministe $\mathcal{A} = (\mathcal{Q}, q_0, F, \delta)$, où par la suite nous prendrons $\mathcal{Q} = \{q_0, q_1, \ldots, q_{m-1}\}$. Ainsi, caractériser $\mathcal{A}$ revient exactement à caractériser L puisqu'ils définissent le même ensemble de mots. Nous allons donc essayer de décrire $\mathcal{A}$ par une formule logique MSO.

L'avantage des automates est qu'ils sont plus facile à décrire mathématiquement. L'idée est alors de voir que la logique MSO, par rapport à la logique du premier ordre, nous autorise à utiliser des sous-ensembles, ce qui est plus avantageux pour caractériser les états d'un automate. Nous pourrions par exemple les caractériser en leur associant à chacun l'ensemble des mots qu'ils seraient succeptibles d'accepter pour la fonction de transition δ.

Cependant, nous sommes en logique monadique et ces sous-ensembles ne peuvent donc pas dépendre d'un mot mais d'un caractère. La difficulté est ici dans le fait que nous considèrons toutes les possibilités de lecture et donc l'indice d'un caractère ne peut caractériser un état. Cela tient dans le fait que pour un automate donné il n'est souvent pas très difficile de trouver deux mots dont le caractère d'indice n est pour l'un dans un état α et pour l'autre dans un état β.

Néanmoins, la définition de $L(\phi) = \{s \in \Sigma^* \mid M_s \vDash \phi\}$ nous montre que nous n'aurions besoin de considérer la formule que pour un seul mot s à chaque fois. Ce qui signifie que s pourrait fixer le contexte pour ces sous-ensembles. Ainsi un entier pourrait parfaitement traduire un mot : plus précisément il représenterait le sous-mot parcouru jusqu'à lui lors de la lecture du mot s. Cependant, il faut voir qu'à chaque fois que nous changerons de modèle pour ϕ (et donc de s) les sous-ensembles caractérisant les états changeront. Ces sous-ensembles dépenderont donc du modèle utilisé et doivent donc être nécessairement *lié* au sens logique du terme, ce qui peut se traduire par une notation existentielle. Dans le sens de cette idée, nous allons proposer la MSO-formule suivante :

$$\phi = \exists X_0 \exists X_{m-1} \varphi_{\text{part}} \wedge \varphi_{\text{start}} \wedge \varphi_{\text{trans}} \wedge \varphi_{\text{accept}}$$

où $X_0, \ldots, X_{m-1}$ sont censés former une partition de l'univers de M_s et où X_i pour $i \in \{0, \ldots, m-1\}$ est l'ensemble des positions du mot s pour lesquelles l'automate $\mathcal{A}$ est tombé dans l'état $q_i \in \mathcal{Q}$ lors de la lecture s. Cela est traduit dans l'ensemble des formules suivantes :

(i) φ_{part} assure que $X_0, \ldots, X_{m-1}$ partitionne l'univers de M_s.

$$\forall x \left(\bigvee_{i=0}^{m-1} X_i(x) \wedge \neg \left(\bigvee_{0 \leq i < j \leq m-1} X_i(x) \wedge X_j(x) \right) \right)$$

(ii) φ_{start} assure que l'automate commence à l'état q_0.

$$\forall x \bigwedge_{a \in \Sigma} \left((P_a(x) \wedge (\forall y,\ y \geq x)) \Rightarrow X_{\text{Indice}(\delta(q_0,a))}(x) \right)$$

(iii) φ_{trans} assure que les transitions sont faites en accord avec δ.

$$\forall x \forall y \bigwedge_{i=0}^{m-1} \bigwedge_{a \in \Sigma} \left(((x \prec y) \wedge X_i(x) \wedge P_a(y)) \Rightarrow X_{\text{Indice}(\delta(q_i,a))}(y) \right)$$

où $(x \prec y) \equiv (x < y) \wedge \neg(\exists z((x < z) \wedge (z < y)))$

(iv) φ_{accepts} assure qu'à la fin du mot s, $\mathcal{A}$ tombe dans un état acceptant.

$$\forall x \left((\forall y \ (y \leq x)) \Rightarrow \bigvee_{q_i \in F} X_i(x) \right)$$

Il n'est alors plus difficile de voir que $s \in L(\phi) \Leftrightarrow s \in \text{Rec}(\mathcal{A}) \Leftrightarrow s \in L$ ce qui implique $L = L(\phi)$.

$\boxed{\Rightarrow}$ Réciproquement, soit ϕ un MSO-énoncé dans le vocabulaire σ_Σ. Dans cette deuxième partie de la démonstration le but est de trouver un automate qui reconnait les mots s vérifiant $M_s \vDash \phi$. Le **Lemme 1.** que nous avons introduit dans la première partie laisse deviner que les *types* pourraient être de bons candidats pour représenter les états de cet automate. En effet, ce sont des objets fortement liés aux mots du langage, qui possèdent par le lemme une assez bonne transmission de l'information, et qui pourraient donc convenir à une construction tel que l'automate dont ils formeraient les états puisse reconnaitre les mots en questions.

Considérons avant tout le rang de quantification de ϕ pour se placer dans le contexte d'utilisation du lemme : posons $k = qr(\phi)$. La difficulté ici est que sans connaitre plus d'information sur ϕ nous pouvons assez mal décrire les structures M_s et donc les mots que serait succeptible de reconnaitre l'automate. Nous allons donc essayer de fournir une écriture plus adéquate de ϕ sans pour autant diminuer la généralité de notre démonstration.

Tout ce que nous savons de ϕ est que c'est un énoncé de rang k. Nous pouvons donc tout d'abord nous donner une énumération $\tau_0, \ldots, \tau_m$ de tous les types de MSO$[k]$ de σ_Σ-structures. Le lecteur notera que cette énumération est fini à cause du fait que k est lui-même fini. De plus, le fait que l'énoncé ϕ soit dans MSO$[k]$ nous assure que cette énumération nous fournira suffisament d'informations pour le décrire.

De manière plus parlante nous pouvons comprendre un τ_i pour $i \in \{0, 1, \ldots, m\}$ comme l'ensemble des énoncés de MSO$[k]$ *validés* par une structure de la forme M_s sans aucune contrainte sur s si ce n'est qu'il appartient à Σ^*. Ainsi nous avons à disposition l'ensemble des types de MSO$[k]$, ce qui nous autorise à décider lesquelles d'entre eux sont « compatibles » avec ϕ et ceux qui ne le sont pas. Nous préciserons ce que nous entendons par « compatible » un peu plus bas, mais remarquons déjà que de cette manière, nous commençons à distinguer quel type pourrait être un état acceptant et l'inverse. Notons aussi que la considération de tous les types de MSO$[k]$ est nécessaire pour être sûr que nous décrivons entièrement l'énoncé ϕ et donc être capable de trouver un automate caractérisant exactement cet énoncé parmi les autres énoncés de MSO$[k]$.

Pour pouvoir faire communiquer l'ensemble des types τ_i avec ϕ, nous allons essayer de les caractériser sous forme de formule. Par exemple, la disjonction de l'ensemble de formules que contient un type peut convenir. Ainsi, sans difficulté nous pouvons affirmer que nous pouvons trouver un MSO-énoncé ψ_i de rang de quantification k tel que :

$$M_s \vDash \psi_i \Leftrightarrow \text{mso-tp}_k(M_s) = \tau_i$$

Et puisque notre énumération $\tau_0, \ldots, \tau_m$ balaye l'ensemble des types MSO$[k]$, les ψ_i balayent pareillement l'ensemble des formules de MSO$[k]$ à équivalence près. Ainsi ϕ est forcément décrit par un ensemble non nul de type MSO$[k]$ et peut donc s'écrire de manière équivalente comme une disjonction de formules ψ_i.

De manière plus formel, si nous définissons l'ensemble $F \subset \{\tau_0, \ldots, \tau_m\}$ des types consistant avec ϕ (ce sont les types que nous avions appelé « compatibles » avec ϕ), alors ϕ est équivalent à la formule $\bigvee_{\tau_i \in F} \psi_i$.

Par convention nous prendrons τ_0 le type de M_ϵ, où ϵ est le mot vide. C'est alors le seul type parmi tous les autres τ_i à être consistant pour l'énoncé $\psi_0 = \neg\exists x\ (x = x)$.

Venons en à la construction de l'automate en proposant la forme suivante :

$$\mathfrak{A}_\phi = (\{\tau_0, \ldots, \tau_m\}, \tau_0, F, \delta_\phi),$$

avec $S = \{\tau_0, \ldots, \tau_m\}$ l'ensemble des états, τ_0 l'état de départ, F l'ensemble des états finaux et $\delta_\phi\ :\ S \times \Sigma \to \mathcal{P}(S)$ la fonction de transition définit de la manière suivante :

$$\tau_j \in \delta_\phi(\tau_i, a) \Leftrightarrow \exists s \in \Sigma^*\ \text{mso-tp}_k(M_s) = \tau_i \text{ et } \text{mso-tp}_k(M_{s.a}) = \tau_j$$

Avec cette définition de la fonction de transition et le **Lemme 1.** il n'est pas difficile de deviner que l'automate $\mathcal{A}_\phi$ sera déterministe. En effet, il faut voir premièrement que le lemme peut se reécrire de la manière suivante : Si nous prenons $s_1, s_2, t_1, t_2 \in \Sigma^*$ tel que $M_{s_1} \equiv_k^{\text{MSO}} M_{t_1}$ et $M_{s_2} \equiv_k^{\text{MSO}} M_{t_2}$, alors $M_{s_1.s_2} \equiv_k^{\text{MSO}} M_{t_1.t_2}$. Cela vient surtout du fait que la concaténation de deux mots est isomorphe à l'union disjointe de ces deux mots. Ensuite, pour montrer le déterminisme de l'automate, plaçons nous dans le cas où nous aurions $\text{mso-tp}_k(M_{m_1}) = \text{mso-tp}_k(M_{m_2}) = \tau_i$ (avec $m_1 \neq m_2$) ce qui revient à écrire que $M_{m_1} \equiv_k^{\text{MSO}} M_{m_2}$. Comme de manière évidente $M_a \equiv_k^{\text{MSO}} M_a$, il vient que $M_{m_1.a} \equiv_k^{\text{MSO}} M_{m_2.a}$ $(*)$. Un non-déterminisme de l'automate pourrait par exemple se traduire dans le fait que $\text{mso-tp}_k(M_{m_1.a}) = \tau_{j_1}$, $\text{mso-tp}_k(M_{m_2.a}) = \tau_{j_2}$ et $j_1 \neq j_2$. Mais la relation $(*)$ montre que, puisque $M_{m_2.a} \vDash \psi_{j_2}$ avec $qr(\psi_{j_2}) = k$, que nous avons forcément $M_{m_1.a} \vDash \psi_{j_2}$, ce qui implique par définition de ψ_{j_2} que $\text{mso-tp}_k(M_{m_1.a}) = \tau_{j_2} \neq \tau_{j_1}$. L'automate $\mathcal{A}_\phi$ ne peut donc pas être non-déterministe.

Montrons maintenant par une récurrence simple sur la longueur du mot que pour tout mot s, après lecture d'un tel mot, l'automate tombe dans l'état τ_i si $\text{mso-tp}_k(M_s) = \tau_i$. Pour le mot vide, nous avions convenu que la lecture s'arrêtait à l'état de départ $\tau_0 = \text{mso-tp}_k(M_\epsilon)$. Supposons maintenant que $\text{mso-tp}_k(M_s) = \tau_i$ et que $\mathcal{A}_\phi$ est dans l'état τ_i après lecture de s. Par définition de la fonction de transition δ_ϕ et le fait que $\mathcal{A}_\phi$ soit *déterministe*, ce dernier tombera dans l'état τ_j après lecture de a si nous avons $\tau_j = \text{mso-tp}_k(M_{s.a})$.

Ainsi, $\mathcal{A}_\phi$ accepte un mot s ssi $\text{mso-tp}_k(M_s)$ est dans F, c'est-à-dire ssi il est consitant avec ϕ, ce qui arrive ssi $M_s \vDash \phi$. Cela revient donc à dire que $\mathcal{A}_\phi$ accepte exactement le langage $L(\phi)$. Ceci termine donc la preuve du Théorème de Büchi.

4 Conclusion

Le théorème de Büchi est intéressant dans sa construction, car il donne aux langages rationels une représentation Logique explicite. Nous avons eu besoin dans notre exposé de la Logique du Second Ordre car nous avons vu que la Logique du Premier Ordre ne suffisait pas. Cependant

nous nous sommes restreint à une logique la plus élémentaire possible, dans notre cas la logique Monadique du second ordre. Pour terminer, nous aurions pu aussi essayer de savoir si la Logique Monadique du Second Ordre pouvait trouver d'autres représentations que dans la théorie des automates déterministes, ces derniers recouvrant déjà un grand nombre de domaines dont les machines de Turing.

5 Références

[1] Leonid Libkin, *Elements of Finite Model Theory*, p122-125.