Automaticité des propriétés modales des structures finies

Anuj Dawar

Computer Laboratory
University of Cambridge

avec Stephan Kreutzer
RWTH, Aix-la-Chapelle

Logique modale

Logique modale parametrée (ML) :

Pour un ensemble P de propositions atomiques et un ensemble A d'actions

Les formules de la logique sont :
- T et F
- p $\qquad (p \in P)$
- $\varphi \wedge \psi$; $\varphi \vee \psi$; $\neg\varphi$
- $[a]\varphi$; $\langle a\rangle\varphi$ $\qquad (a \in A)$.

Systèmes de transitions

Les formules sont interprétées dans un *système de transitions* :

$$M = (S, A^M, \mathcal{P})$$

– S est un ensemble d'*états* ;
– pour tout $a \in A$, $a^M \subseteq S \times S$ est une *relation de transition* dans A^M ;
– $\mathcal{P} : S \to \mathsf{Pow}(P)$;

Pour tout $s \in S$
– $(M, s) \models p$ si $p \in \mathcal{P}(s)$.
– $(M, s) \models \langle a \rangle \varphi$ s'il existe t avec $(s, t) \in a^M$, $(M, t) \models \varphi$.
– $(M, s) \models [a]\varphi$ si pour tout t avec $(s, t) \in a^M$, $(M, t) \models \varphi$.
– les connecteurs booléens comme toujours.

Bisimulation

$M = (S, A^M, \mathcal{P}^M) \sim N = (T, A^N, \mathcal{P}^N)$
s'il existe une relation $B \subseteq S \times T$ *telle que :*
– si $B(s,t)$ $p^M(s) \Leftrightarrow p^N(t)$;
– si $B(a,b)$ et $a^M(s,u)$ il existe un v tel que $a^N(,d)$ et $B(u,v)$;
– si $B(s,t)$ et $a^N(t,u)$ il existe un v tel que $a^M(s,v)$ and $B(v,u)$.

Toute classe de structures définissable en ML est close par bisimulation.

Par *propriété modale* on entend une classe $\mathcal{C}$ close par bisimulation.

Le μ-calcul

Le μ-calcul L_μ est une extension de la logique modale de base.

Additionnellement un ensemble de variables $X_1, X_2, \ldots$.

Les formules du μ-calcul :
– toutes les règles de la logique modale
– X
– $\mu_X\varphi$; $\nu_X\varphi$ — si toute occurrence de X dans φ est positive.

$(M, \bar{X}^M, s) \models \mu_X\varphi$ si
s est l'ensemble X minimal tel que $X \leftrightarrow \varphi$ dans $M, \bar{X}$

Autres extensions

L_μ^ω permet de former des points fixes de relations d'arité quelconque.

Sur les systèmes de transitions finies, elle est capable d'exprimer toute les proprietés décidables en temps polynomial et qui sont close par bisimulation.

(Otto)

MIC resemble L_μ, sauf qu'elle emploie une sémantique basée sur les points fixes inflationnaires à la place des points fixes minimaux.

$$\text{ML} \leq L_\mu \leq \text{MIC} \leq L_\mu^\omega.$$

Points fixes inflationnaires

Les points fixes inflationnaires sont une alternative aux points fixes minimaux pour introduire les définitions récursives.

$$\mathbf{ifp}X.\varphi$$

est la limite de la série :

$$\begin{array}{rcll} X^0 & = & \emptyset & \\ X^{\alpha+1} & = & X^\alpha \cup \varphi(X^\alpha) & \\ X^\lambda & = & \bigcup\limits_{\alpha<\lambda} X^\alpha & \quad\quad \text{pour les ordinaux limites } \lambda \end{array}$$

L'ordinal minimal α pour lequel $X^\alpha = X^{\alpha+1}$ est *l'ordinal clôture* de φ.

Si toute occurrence de X dans φ est positive, la limite est égale a l'ensemble défini par $\mu_X \varphi$.

MIC

IFP est l'extension de la logique du premier ordre par des points fixes inflationnaires.

Elle a le même pouvoir d'expression que LFP (l'extension avec points fixes minimaux).

(D. Grädel, Kreutzer 2001) introduit MIC—l'extension de la logique modale par des points fixes inflationnaires.

En général, on permet dans MIC des récursions simultanées.

Pour un système S de règles :

$$S = \begin{cases} X_1 \leftarrow \varphi_1 \\ \quad\vdots \\ X_k \leftarrow \varphi_k \end{cases}$$

on a la formule $(\mathbf{ifp} X_i : S)$.

MIC

MIC *est plus expressive que* L_μ.

On peut y exprimer des langages non-rationnels.

Il existe des formules satisfaisable de MIC qui n'ont pas de modèle fini.

Le problème de la satisfaisabilité des formules de MIC n'est pas décidable.

On peut y représenter l'arithmétique du premier ordre.

Les points fixes simultanés sont plus expressifs que les points fixes simples.

Le problème de la satisfaction $M \models \varphi$ *est* $\mathrm{PS}\textsc{pace}$*-complet.*

MIC *est moins expressive que* L_μ^ω.

Automaticité

L'automaticité est une mesure de la complexité des langages.

On dit qu'un langage $L \subseteq \Sigma^*$ *a automaticité* $O(f)$*, s'il existe une constante* c *telle que :*

pour tout n, il existe un automate fini avec au plus $c \cdot f(n)$ états qui accepte un mot w de $\leq n$ lettres si et seulement si $w \in L$.

ex, un langage est rationnel ssi il a automaticité $O(1)$.

Pour tout L (même non-décidable), l'automaticité de L est $O(2^n)$.

Il existe un langage dont l'automaticité *non-déterministe* est polynomiale mais dont l'automaticité déterministe est polynomiale.

(Shallit and Breitbart)

L_μ vs. MIC

Sur les mots, toute formule de L_μ défini un langage rationnel.

Par contre, il existe des langages définissables en MIC qui ont l'automaticité exponentielle.

Par example,

$$\{cwdw \mid w \in \{a,b\}^*\}.$$

Automates sur arbres finis

Un automate fini déterministe pour les arbres finis est :

$$\mathcal{A} = (Q, A, \delta, F, S),$$

où Q est un ensemble d'états, A est l'alphabet, $S \subseteq Q$ est l'ensemble des états initiaux, $F \subseteq Q$ est l'ensemble des états finals, et $\delta : Q \times A \to 2^Q$ est la fonction de transition.

On dit que $\mathcal{T}$ est accepté par $\mathcal{A}$, s'il existe une fonction d'étiquetage $l : \mathcal{T} \to \mathcal{Q}$ sur les noeuds de $\mathcal{T}$ telle que
– la racine de $\mathcal{T}$ est étiquetté $q_0 \in S$;
– si $l(v) = q$, $\{l(w) : v \xrightarrow{a} w\} = \delta(q, a)$, pour tout $a \in A$;
– pour toute feuille v, $l(v) \in F$.

L'automaticité sur les arbres finis

NB : Il existe des automates qui définissent des classes d'arbres non-closes par bisimulation. Mais, si F est un singleton et δ est injective sur Q, la classe est nécessairement close par bisimulation.
Celà correspond aux automates bottom-up.

Si $\mathcal{T}$ est une classe d'arbres finis, *close par bisimulation*, on définit son automaticité comme étant la fonction $A_{\mathcal{T}}$ qui à n associe

la taille de l'automate minimal acceptant un arbre T *de hauteur* $\leq n$ si et seulement si $T \in \mathcal{T}$.

Il existe des classes d'arbres d'automaticité non-élementaire.

Les résultats sur les arbres s'appliquent, *a fortiori*, aux systèmes de transition finis *acycliques*.

L'automaticité de bisimulation

L'automaticité maximale est celle du *problème de bisimulation*, malgré qu'il soit décidable en temps polynomial.

Le problème de bisimulation est la classe de structures (M, r) (avec racine r) telle que pour toute paire s_1, s_2 de successeurs de r on a :

$$(M, s_1) \sim (M, s_2).$$

Si $b(n)$ dénote le nombre de classes d'équivalence par bisimulation parmi les arbres de hauteur n, on a :

$$b(n+1) = 2^{b(n)}.$$

MIC vs. L_μ^ω

Le *problème de bisimulation* est définissable en L_μ^ω.

Par contre, toute classe d'arbres finis définissable en MIC a automaticité (au plus) exponentielle.

Pour une formule φ avec $\leq k$ variables liées, si Φ est l'ensemble des sous-formules de φ, l'ensemble de fonctions

$$f : (k(n+1)+1)^k \to 2^\Phi$$

peut servir d'étiquettes.

(D.,Grädel, Kreutzer 2001)

Structures cycliques

Pour généraliser l'automaticité à une mesure sur les structures qui ne sont pas nécessairement acycliques, il nous faut :

- une mesure de *rang* qui généralise la hauteur d'un arbre ;
- une généralisation des automates bottom-up au paravant défini sur les arbres.

Definition

Pour une structure M, le *rang d'induction* de M est le plus grand des ordinaux clôtures de toutes les formules de MIC dans M.

Défini à partir des formules de L_μ plutôt que MIC, le rang d'induction reste inchangé.

Rang bloc

Pour toute structure M, le graphe bloc G_M

- a comme sommets les composantes fortement connexes de M ;
- a un arc (u, v) si $u \neq v$ et $(s, t) \in M$ pour un $s \in u$ et $t \in v$.
- Chaque sommet a un *poids* égal au nombre d'élements de M dans la composante qui lui correspond.

Le *rang* d'un sommet v de G_M est :

$$\mathrm{poids}(v) + \max\{\mathrm{rang}(u) \mid (v, u) \in G_M\}.$$

Le rang bloc de la racine r d'une structure M est le rang de $[r]$ dans $G_{M_{/\sim}}$,
où $M_{/\sim}$ est le quotient de M par bisimulation.

Lemme
Le rang d'induction d'une structure est le même que son rang bloc.

Systèmes d'étiquetage

Un système d'étiquetage $\mathcal{L}$ est un 5-uplet
$\mathcal{L} = (Q, A, P, \delta, \mathcal{F})$ *où*
– Q est un ensemble finis d'états ;
– A est un ensemble finis d'actions ;
– P est un ensemble finis de symboles propositionels ;
– $\mathcal{F} \subseteq Q$ un ensemble *d'états acceptants* ;
– $\delta : 2^{Q \times A} \times 2^{P} \to Q$ est une fonction totale.

On dit qu'un système de transition M avec racine r est accepté par $\mathcal{L}$ s'il existe un étiquetage $f : |M| \to Q$ tel que :

$$f(s) = \delta(\{(f(s'), a) \mid a \in A, s \xrightarrow{a} s'\},\\ \{p \mid p \in P, P(s)\})$$

et $f(r) \in \mathcal{F}$.

Index d'étiquetage

Pour une classe $\mathcal{C}$ de systèmes finis, close par bisimulation, On définit son *index d'étiquetage* (généralisant l'automaticité) comme étant la fonction $A_{\mathcal{C}}$ qui à n associe

la taille minimale d'un système d'étiquetage qui accepte une structure de *rang* n si et seulement si elle est dans $\mathcal{C}$.

Le *problème de bisimulation* a index d'étiquetage non-elementaire.

Il s'agit du plus grand index possible.
La propriété est exprimable en L_{μ}^{ω}.

Résultats

Toute formule de ML exprime une propriété avec index d'étiquetage constant.

Toute formule de L_μ exprime une propriété avec index d'étiquetage au plus polynomial.
Il existe des formules de L_μ exprimant des propriétés avec index d'étiquetage au moins linéaire (polynomial ?).

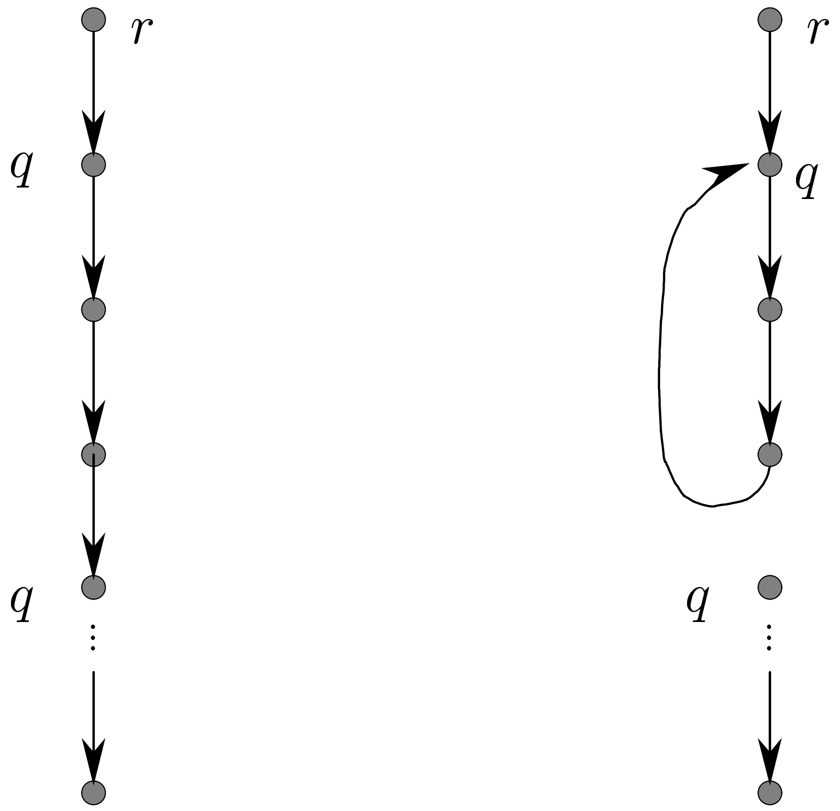
Toute formule de MIC exprime une propriété avec index d'étiquetage au plus exponentiel.
Il existe des propriétés exprimable dans MIC qui ont index au moins exponentiel.

De même façon, l'index des propriétés définissables avec les points fixes partiels est doublement exponentiel.

Il existe des formules de IFP monadique qui expriment des propriétés closes par bisimulation mais non-exprimables dans MIC.

Borne inférieure

La formule $\mu X.\Box X$ a index linéaire.



Equivalence de trace

Etant donné une structure $M = (S, A^M, \mathcal{P}^M)$, une racine $r \in S$ et une partie $F \subseteq S$ d'états finals, le *langage de trace* $L(M, r) \subseteq A^*$ est l'ensemble des mots dans l'alphabet A répresentants les séquences de transitions allant de r pour arriver à un état de F.

Le problème d'équivalence de trace est la classe de structures (M, r) telle que pour toute paire s_1, s_2 de successeurs de r on a :

$$L(M, s_1) = L(M, s_2).$$

On dit que M est unaire si A est un singleton.

Index de l'equivalence de trace

Sur les *structures unaires acycliques*, l'index d'étiquetage du problème d'equivalence de trace est exponentiel.

Sur les *structures unaires* en général, l'index d'étiquetage du problème d'equivalence de trace est doublement exponentiel.

Sur les *structures acycliques* l'index d'étiquetage du problème d'equivalence de trace est doublement exponentiel.

L'index d'étiquetage du problème général d'equivalence de trace est triplement exponentiel.

En conclusion

L'automaticité est une mesure de complexité des propriétés modales de systèmes de transition finis qui est à la fois subtile et utile.