Soutenance de stage de Master 2 Recherche

Gestion des théories par les foncteurs logiques

Étienne André

Projet LIS

Encadreur : Sébastien Ferré

Motivation : gestion de documents

- Gestion de documents électroniques
  - Nombre important
  - Formats variés (photos, données BibTeX, fichiers `.ml`, etc.)

- Questions
  - Organisation de ces documents
  - Recherche dans ces documents

Motivation : logique

- Souhait d'un langage
  - Description des objets et représentation des requêtes
  - Relation de déduction entre objets et requêtes
- Souhait d'un formalisme logique
  - Avantage : expressivité
  - Représentation logique des données et des requêtes
  - Représentation logique de la relation de déduction
    - Notion essentielle de *subsomption* : relation de généralisation
    - Exemple : $\mathit{Article} \wedge \mathit{Long} \sqsubseteq \mathit{Article} \vee \mathit{These}$

Exemple de logique

- Logique propositionnelle d'attributs valués par des chaînes
  - Exemple de description d'objet
    - $(\textit{titre}, \text{"}\texttt{Foncteurs Logiques}\text{"}) \wedge (\textit{annee}, \text{"2006"})$
  - Exemple de requête
    - $(\textit{annee}, \text{"2005"}) \vee (\textit{annee}, \text{"2006"})$
  - Exemple de test valide de subsomption
    - $$\begin{array}{ll} & (\textit{titre}, \text{"}\texttt{Foncteurs Logiques}\text{"}) \wedge (\textit{annee}, \text{"2006"}) \\ \sqsubseteq & (\textit{annee}, \text{"2005"}) \vee (\textit{annee}, \text{"2006"}) \end{array}$$

- Logique différente selon les applications
  - Exemples : description de fichiers, coordonnées géographiques

$\Rightarrow$ besoin de généricité au niveau logique

Présentation des foncteurs logiques

- Réponse au besoin de généricité logique
  - Composants logiques réutilisables 
  - Création d'une logique par composition de foncteurs logiques
- Expressivité
  - Domaines concrets : entiers *(Int)*, chaînes *(String)*, etc.
    - Exemple : $x \in [0; 2] \wedge x \in [1; 4]$ s'écrit $[0..2] \wedge [1..4]$
  - Structures algébriques : somme *(Sum)*, produit *(Prod)*, etc.
- Exemple : foncteur logique *Prop(X)*
  - Logique propositionnelle *Prop(Atom)*
  - Atomes abstraits par le paramètre formel X
  - Nouvelle logique : *Prop(Interval(Int))*
- Fonctions d'une ou plusieurs logiques vers une nouvelle logique

La logique de description $\mathcal{ALC}$

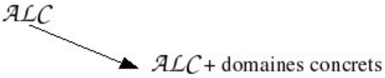
- Logiques de description
  - Classe de logiques permettant de décrire des objets
- Expressivité de $\mathcal{ALC}$
  - Connecteurs logiques (logique propositionnelle)
    $Publication \sqcap (EnFrancais \sqcup EnAnglais) \sqcap \neg Court$
  - Quantification
    - Existentielle : $\exists cite.Article$
    - Universelle : $\forall auteur.Chercheur$
- Logiques de description utilisées pour la définition d'ontologies
  - Ontologie : connaissances du domaines (forme de *théorie*)
    - Définition d'axiomes
    - Exemple : $Article \dot{\leq} Publication$
  - Exemples d'utilisation : bioinformatique, Web sémantique

Recomposition de $\mathcal{ALC}$ par les foncteurs logiques

- $\mathcal{ALC}$ recomposable par des foncteurs logiques
- Exemples de buts dans $\mathcal{ALC}$
  - But prouvable : $Article \sqcap Court \sqsubseteq Article \sqcup These$
  - But non prouvable : $Article \sqcap Court \sqsubseteq Publication$
    - Rien ne dit qu'un article est une publication
- Besoin de l'utilisation d'une théorie

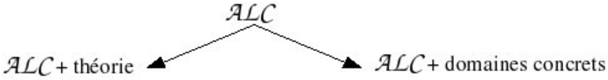
Objectifs

- Existant
  - Recomposition de $\mathcal{ALC}$ par les foncteurs logiques
    - Avec domaines concrets
    - Sans théorie



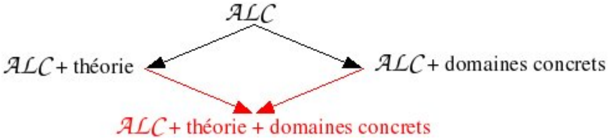
Objectifs

- Existant
  - Recomposition de $\mathcal{ALC}$ par les foncteurs logiques
    - Avec domaines concrets
    - Sans théorie
  - Solveurs classiques de $\mathcal{ALC}$
    - Sans domaines concrets
    - Avec théorie



Objectifs

- Existant
  - Recomposition de $\mathcal{ALC}$ par les foncteurs logiques
    - Avec domaines concrets
    - Sans théorie
  - Solveurs classiques de $\mathcal{ALC}$
    - Sans domaines concrets
    - Avec théorie
- Objectifs
  - Ajout de la théorie aux foncteurs logiques
  - Recomposition de $\mathcal{ALC}$ par les foncteurs logiques
    - Avec domaines concrets
    - Avec théorie



Plan

Définition d'une logique

Définition (Logique)

Une logique est un tuple (AS, S, P, T) *constitué d'une syntaxe abstraite, d'une sémantique, d'un ensemble de procédures et d'un type.*

- Syntaxe : ensemble des formules
- Sémantique : signification des formules de la *syntaxe*
- Procédures : interface utilisable de la logique
  - Procédure essentielle : relation de subsomption $\sqsubseteq$
- Type : ensemble des propriétés des procédures vis-à-vis de la sémantique
  - Propriétés essentielles liées à la subsomption : $cs_{\sqsubseteq}$ (cohérence) et $cp_{\sqsubseteq}$ (complétude)

Définition d'une théorie

Définition (Théorie)

Une théorie est un ensemble d'axiomes de la forme

$f \stackrel{.}{\leq} g \quad$ *(« est une sorte de »)*

où f *et* g *sont des formules de la syntaxe abstraite de la logique considérée.*

- Note
  - $f \doteq g$ peut se réécrire en $f \stackrel{.}{\leq} g$ et $g \stackrel{.}{\leq} f$
- Exemples d'axiomes

$$\begin{array}{rcl} Article & \stackrel{.}{\leq} & Publication \\ These & \stackrel{.}{\leq} & Publication \sqcap (EnFrancais \sqcup EnAnglais) \sqcap \neg Court \end{array}$$

Plan

Bibliothèque LogFun

- Bibliothèque de foncteurs logiques LogFun
  - Nombreux foncteurs logiques implémentés
  - Implémentation en Caml
  - http ://www.irisa.fr/LIS/ferre/logfun/
- Intégrable comme plugin dans des systèmes d'organisation et de recherche d'information
  - Camelis
    http ://www.irisa.fr/LIS/ferre/camelis/
  - LISFS : système de fichiers logique
    https ://gforge.inria.fr/projects/lisfs/

Définition d'un foncteur logique (1/2)

Définition (Foncteur logique sans théorie)

Un foncteur logique F est une fonction prenant en argument un tuple de logiques $(L_1, \ldots, L_n)$, et se décompose en un tuple (AS_F, S_F, P_F, T_F) où :

$AS_{F(L_1,\ldots,L_n)} = AS_F(AS_{L_1}, \ldots, AS_{L_n})$; *(de même pour S, P et T)*

et retourne une logique sans théorie.

- Exemple : $Prod(L_1, L_2)$
  - $AS_{Prod(L_1,L_2)} = AS_{L_1} \times AS_{L_2}$ (couples de formules de L_1 et L_2)
  - $(f_1, f_2) \sqsubseteq_{Prod(L_1,L_2)} (g_1, g_2) =_{def} f_1 \sqsubseteq_{L_1} g_1 \wedge f_2 \sqsubseteq_{L_2} g_2$
  - $cs_{\sqsubseteq_{Prod(L_1,L_2)}}$ si $cs_{\sqsubseteq_{L_1}}$ et $cs_{\sqsubseteq_{L_2}}$
- Certificat de validité obtenu par composition des propriétés

Définition d'un foncteur logique (2/2)

Définition (Foncteur logique avec théorie)

Un foncteur logique F est une fonction prenant en argument un tuple de logiques $(L_1, \ldots, L_n)$, *et se décompose en un tuple* (AS_F, S_F, P_F, T_F) *où :*

$AS_{F(L_1,\ldots,L_n)} = AS_F(AS_{L_1}, \ldots, AS_{L_n})$; *(de même pour* S, P *et* T*)*

et retourne

- *une logique sans théorie,*

Définition d'un foncteur logique (2/2)

Définition (Foncteur logique avec théorie)

Un foncteur logique F est une fonction prenant en argument un tuple de logiques $(L_1, \ldots, L_n)$, et se décompose en un tuple (AS_F, S_F, P_F, T_F) où :
$AS_{F(L_1,\ldots,L_n)} = AS_F(AS_{L_1}, \ldots, AS_{L_n})$; *(de même pour S, P et T)*
et retourne

- *une logique sans théorie,*
- *ou une fonction G prenant en paramètre une théorie th, et retournant une logique.*
  Dans ce cas, soit $G = F(L_1, \ldots, L_n) = (AS_G, S_G, P_G, T_G)$. Alors,
  $G(th) = (AS_G(th), S_G(th), P_G(th), T_G(th))$

Exemples de foncteurs logiques

- Foncteurs logiques autorisant une théorie
  - *Taxo()(th)* : axiomes sur des atomes
    - Axiomes de la forme $f \stackrel{.}{\leq} g$
    - Exemple : $Article \stackrel{.}{\leq} Publication$
    - Subsomption : $a \sqsubseteq b =_{def} (a = b) \vee (\exists c : a \stackrel{.}{\leq} c \wedge c \sqsubseteq b)$
  - *Prop(X)(th)* : terminologies
    - Exemple dans *Prop(Sum(String, Interval(Int)))(th)*
      $PubliRecenteNapoli \stackrel{.}{\leq} (annee, 2005..2007) \wedge (auteur, \texttt{"Napoli"})$

Foncteur logique *Prop(X)* sans théorie

- Syntaxe de *Prop*
  - Atomes du foncteur X en paramètre
  - Fermeture de la syntaxe par les connecteurs booléens $\vee$, $\wedge$ et $\neg$
- Relation de subsomption $f \sqsubseteq g$
  - Preuve de $\vdash \neg f \vee g$ dans un calcul de séquents
  - Basé sur le calcul de séquents pour la logique propositionnelle
  - Ajout d'axiomes ayant trait à la logique X en paramètre

Foncteur logique *Prop(X)(th)* avec théorie

- Ajout d'axiomes de la forme $f \dot{\leq} g$
  - f : terme défini
  - g : expressions complexes contenant éventuellement f
- Ajout à la syntaxe des termes définis par les axiomes (ensemble disjoint de X)
- Modification du calcul de séquents
  - Ajout de règles pour la prise en compte des termes
  - Mémorisation pour éviter les boucles dues à la possible récursivité dans la composition des foncteurs
- Preuves des propriétés à reprendre entièrement

Recomposition de $\mathcal{ALC}$

- Soit une théorie *th*

$$L = Prop(Set(Prod(Atom, L)))(th)$$

- Validité de la logique (cohérence et complétude des opérations) prouvée automatiquement par composition des propriétés des différents foncteurs

- Exemple de théorie *th* :

$$\begin{array}{rcl} Article & \dot{\leq} & Publication \\ ArticleLD & \dot{\leq} & Article \sqcap \exists cite.ArticleLD \end{array}$$

- Exemple de but prouvable :

$$\begin{array}{ll} & ArticleLD \sqcap \forall auteur.Chercheur \\ \sqsubseteq & Publication \sqcap \exists cite.ArticleLD \end{array}$$

Conclusion

- Recomposition de logiques à l'aide de foncteurs logiques
  - Foncteurs implémentables par un logicien
  - Foncteurs composables aisément par un développeur
    - Certificat automatique (cohérence et complétude des opérations)
  - Théorie définissable par un expert du domaine
    - Recomposition de la logique de description $\mathcal{ALC}$ avec théorie

- Implémentation
  - Foncteur *Taxo* créé
  - Foncteur *Prop* modifié
  - Recomposition de $\mathcal{ALC}$ effectuée et testée

Perspectives de recherche

- Amélioration de l'expressivité des axiomes de *Prop*
  - Autorisation d'axiomes de la forme $f \doteq g$
    - f terme, g expression quelconque
  - Autorisation d'expressions quelconques à gauche
- Recomposition de la logique de description $\mathcal{SHOIQ}$
  - Intégration au Web sémantique
    $\mathcal{SHOIQ}$ standard du W3C pour le Web sémantique

¬¬-Rule: $\dfrac{\Delta \vdash X, \Gamma}{\Delta \vdash \neg\neg X, \Gamma}$ literal-Rule: $\dfrac{\overline{L_a}, \Delta_1 \mid \Delta_2 \vdash \Gamma}{\Delta_1 \mid \Delta_2 \vdash L_a, \Gamma}$

β-Rule: $\dfrac{\Delta \vdash \beta_1, \ldots, \beta_n, \Gamma}{\Delta \vdash \beta, \Gamma}$ α-Rule: $\dfrac{\Delta \vdash \alpha_1, \Gamma \quad \ldots \quad \Delta \vdash \alpha_n, \Gamma}{\Delta \vdash \alpha, \Gamma}$

$\top$-Axiom: $\neg b, \Delta_1 \mid \Delta_2 \vdash \Gamma$ if $def_{\top_A}$ and $\top_A \sqsubseteq_A b$

$\bot$-Axiom: $a, \Delta_1 \mid \Delta_2 \vdash \Gamma$ if $def_{\bot_A}$ and $a \sqsubseteq_A \bot_A$

$\sqsubseteq$ -Axiom: $a, \neg b, \Delta_1 \mid \Delta_2 \vdash \Gamma$ if $a \sqsubseteq_A b$

$\sqcap$-Rule: $\dfrac{a \sqcap_A b, \Delta_1 \mid \Delta_2 \vdash \Gamma}{a, b, \Delta_1 \mid \Delta_2 \vdash \Gamma}$ if $def_{\sqcap}(a, b)$

$\sqcup$-Rule: $\dfrac{\neg(a \sqcup_A b), \Delta_1 \mid \Delta_2 \vdash \Gamma}{\neg a, \neg b, \Delta_1 \mid \Delta_2 \vdash \Gamma}$

term-Axiom: $t, \neg t, \Delta_1 \mid \Delta_2 \vdash \Gamma$

term-Rule: $\dfrac{\Delta_1 \mid \Delta_2, \overline{L_t} \vdash \Gamma}{\Delta_1 \mid \Delta_2 \vdash L_t, \Gamma}$ if $\overline{L_t} \notin \Delta$

term'-Rule: $\dfrac{\Delta_1 \mid \Delta_2 \vdash \Gamma}{\Delta_1 \mid \Delta_2 \vdash L_t, \Gamma}$ if $\overline{L_t} \in \Delta$

def-Rule: $\dfrac{t, \Delta_1 \mid \Delta_2 \vdash \neg C, \Gamma}{\Delta_1 \mid \Delta_2, t \vdash \Gamma}$ if $t \stackrel{.}{\sqsubseteq} C \in th$

$\neg def_{\stackrel{.}{\sqsubseteq}}$-Rule: $\dfrac{\neg t, \Delta_1 \mid \Delta_2 \vdash \Gamma}{\Delta_1 \mid \Delta_2, \neg t \vdash \Gamma}$