PHÁT TRIỂN CHƯƠNG TRÌNH LOGIC MÔ TẢ CHO VIỆC TÍCH HỢP CÁC QUY TẮC VÀ CÁC ONTOLOGY CHO WEB NGỮ NGHĨA

Hoàng Nguyễn Tuấn Minh

Phòng Công tác Học sinh, Sinh viên, Trường Đại học Khoa học – Đại học Huế

Email: hntminh83@yahoo.com

TÓM TẮT

Web ngữ nghĩa ngày càng phát triển, một yêu cầu quan trọng của kiến trúc được phân lớp của web ngữ nghĩa là tích hợp các các quy tắc và các ontology đang được nhiều nhà nghiên cứu quan tâm. Trong bài báo, chúng ta xem xét các vấn đề trong việc tích hợp các quy tắc và các ontology hiện nay và cũng như phân loại các đề xuất theo các phương pháp tiếp cận lý thuyết khác nhau. Ngoài ra chúng ta tập trung vào hướng tiếp cận chương trình logic mô tả trong việc tích hợp các quy tắc và các ontology cho web ngữ nghĩa cùng các vấn đề liên quan của nó.

Từ khóa : *Lập trình logic, logic mô tả, ontology, Web ngữ nghĩa.*

1. MỞ ĐẦU

Web ngữ nghĩa [1,2,3] là một sự phát triển mới trên nền của Web hiện tại theo tiêu chuẩn và công nghệ có thể giúp máy tính hiểu các thông tin trên Web, chúng có thể hỗ trợ cho các công việc khám phá, tích hợp dữ liệu, chuyển hướng dữ liệu và tự động hóa các nhiệm vụ một cách giàu ngữ nghĩa hơn. Kiến trúc phân tầng của Web ngữ nghĩa ngày càng hoàn thiện, một yêu cầu quan trọng của kiến trúc được phân tầng này là để tích hợp tầng Rules và tầng Ontology, hướng đến việc tích hợp các quy tắc và các ontology trong Web ngữ nghĩa.

Trong bài báo này tôi sẽ giới thiệu về các kỹ thuật kết hợp các quy tắc với các ontology và *chương trình logic mô tả*, nó gồm cơ sở tri thức *L* và một tập hữu hạn *các quy tắc logic mô tả P*. Các quy tắc này tương tự như quy tắc trong chương trình logic, nhưng chúng có thể chứa các truy vấn đến *L* trong thân của chúng. Một đặc điểm quan trọng là truy vấn như vậy cũng cho phép xác định một đầu vào từ *P*, nó như là một luồng thông tin từ *P* đến *L*, bên cạnh luồng thông tin từ *L* đến *P* được đưa ra bởi truy vấn bất kỳ đến *L*. Ta cũng sẽ xác định một bộ các ngữ nghĩa cho các lớp khác nhau của chương trình logic mô tả. Cụ thể hơn, chúng ta khái quát các lớp của các chương trình logic dương và chương trình logic phân tầng và xác định ngữ nghĩa mô hình Herbrand.

2. CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Ontology: Thuật ngữ "Ontology" bắt nguồn từ triết học nó liên quan đến việc nghiên cứu của con người về sự tồn tại của tự nhiên. Các nhà nghiên cứu trong khoa học máy tính, đặc biệt là trong Trí tuệ nhân tạo (AI) mượn thuật ngữ này nhằm mục đích hỗ trợ việc chia sẻ và tái sử dụng kiến thức trong hệ thống AI. Cách tiếp cận này đã được Neches và các cộng sự đề xuất *"Một ontology định nghĩa các thuật ngữ và các mối quan hệ cơ bản gồm từ vựng của một chủ đề cũng như các quy tắc kết hợp các thuật ngữ và mối quan hệ để định nghĩa các mở rộng cho từ vựng"*. Theo định nghĩa này một ontology không chỉ bao gồm các thuật ngữ được định nghĩa một cách tường minh trong nó mà còn có tri thức có thể suy diễn được từ ontology. Vào năm 1998, Studer và các cộng sự đã đưa ra định nghĩa ontology khá phù hợp và chính xác hơn. *"Ontology là một đặc tả tường minh, mang tính hình thức của sự khái niệm hóa có thể chia sẽ được. Sự khái niệm hóa đề cập đến một mô hình trừu tượng của một số hiện tượng trong thế giới thực bằng cách xác định khái niệm liên quan đến hiện tượng đó. Tường minh có nghĩa là các khái niệm được sử dụng và các ràng buộc trên chúng được định nghĩa một cách rõ ràng. Hình thức đề cập đến máy có khả năng đọc và hiểu Ontology. Chia sẽ phản ánh quan điểm rằng một Ontology nắm bắt tri thức được chấp nhận bởi một cồng đồng."*

2.2. Chương trình logic chính tắc:

2.2.1. Cú pháp: Cho $\Phi = (\mathcal{P}, \mathcal{C})$ là một bộ từ vựng ngôn ngữ bậc nhất với $\mathcal{C}$ là tập hữu hạn khác rỗng các hằng và $\mathcal{P}$ là tập ký hiệu vị từ không chứa ký hiệu hàm. Cho $\mathcal{X}$ là tập các biến.

Một *hạng thức* là một biến từ $\mathcal{X}$ hoặc một ký hiệu hằng từ Φ. Một *nguyên tố* là một biểu thức có dạng $p(t_1, t_2,..., t_n)$ trong đó *p* là ký hiệu vị từ n ngôi, $n \geq 0$ từ Φ, và $t_1, t_2,..., t_n$ là các hạng thức. Một *literal l* là một nguyên tố *p* (*l* là literal dương) hoặc nguyên tố phủ định $\neg p$ (*l* là literal âm). *Phần bù* của *l* dương là $\neg p$ và của *l* âm là *p*. Một *literal phủ định ngầm (viết tắt NAF-literal)* là một literal *l* hoặc một literal phủ định mặc định *not l*. *Một quy tắc r* là biểu thức có dạng : $a \leftarrow b_1, b_2,..., b_k, not\ b_{k+1},..., not\ b_m$ với $m \geq k \geq 0$ *(1)* trong đó *a* là literal và $b_1,..., b_m$ là các literal hoặc các nguyên tố *đẳng thức (bất đẳng thức)* có dạng $t_1=t_2$ $(t_1 \neq t_2)$ với t_1 và t_2 là các hạng thức. Literal *a* được gọi là *đầu* của quy tắc *r* và phép hội $b_1, b_2,..., b_k, not\ b_{k+1},..., not\ b_m$ là *thân* của quy tắc *r*, trong đó $b_1, b_2,..., b_k$ (hoặc, $not\ b_{k+1},...,not\ b_m$) là *thân dương (hoặc thân âm)*. Người ta dùng *H(r)* để ký hiệu literal *a* đầu của quy tắc, và *B(r)* để ký hiệu tập tất cả literal $B^+(r) \cup B^-(r)$ thân của quy tắc trong đó $B^+(r) = \{ b_1, b_2,..., b_k\}$ và $B^-(r) = \{ b_{k+1},..., b_m\}$. Nếu thân của quy tắc *r* rỗng (trong trường hợp $k = m = 0$) thì *r* là một *dữ kiện (Fact)***,** chúng ta sẽ bỏ "←" trong trường hợp này. Một *chương trình chính tắc P* là một tập hợp hữu hạn các quy tắc. *P* là *một chương trình dương* nếu mọi quy tắc của nó đều không chứa phủ định "*not"*.

2.2.2. Ngữ nghĩa: Một *vũ trụ Herbrand* của một chương trình *P*, ký hiệu HU_P, là một tập hợp tất cả các ký hiệu hằng xuất hiện trong *P*. Nếu không có ký hiệu hằng trong *P* thì $HU_P = \{c\}$, trong đó *c* là một ký hiệu hằng tùy ý trong Φ. Như thường lệ, các hạng thức, các nguyên tố, các literal, các quy tắc, các chương trình … là *nền* nếu và chỉ nếu chúng không chứa biến nào. Một *cơ sở Herbrand* của một chương trình *P* được ký hiệu là HB_P là tập tất cả các literal nền được xây dựng từ các ký hiệu vị từ xuất hiện trong *P* và các ký hiệu hằng trong HU_P. Một *hiện hành nền* của một quy tắc $r \in P$ nhận được từ quy tắc *r* bằng cách thay thế mỗi biến trong *r* bằng một ký hiệu hằng trong HU_P, bằng cách sử dụng một phép thế q cho mỗi biến trong *r* và loại bỏ tất cả nguyên tố đẳng thức và bất đẳng thức $t_1 q = t_2 q$ và $t_1 q \neq t_2 q$. Một hiện hành nền của một quy tắc *r* là *nhất quán* khi và chỉ khi nó không chứa các nguyên tố đẳng thức hoặc bất đẳng thức nào. Chúng ta ký hiệu *ground(P)* là một tập hợp tất cả các hiện hành nền nhất quán của các quy tắc trong *P*.Một tập $X \subseteq HB_P$ của các literal *nhất quán* khi và chỉ khi $\{p, \neg p\} \subsetneq X$ cho mọi nguyên tố $p \in HB_P$. Một *diễn dịch I* trong một chương trình *P* là một tập con nhất quán của HB_P. Một *mô hình* của một chương trình dương *P* là một diễn dịch $I \subseteq HB_P$ sao cho $B(r) \subseteq I$ dẫn đến $H(r) \subseteq I$ đối với mọi $r \in ground(P)$. *Phép chuyển đổi (hay phép chuyển đổi Gelfond-Lifschitz)* của một chương trình *P* liên quan đến một diễn dịch $I \subseteq HB_P$ (ký hiệu là P^I) là một chương trình dương nền nó được nhận được từ *ground(P)* bằng cách xóa đi tất cả các quy tắc *r* mà $B^-(r) \cap I = \emptyset$ và xóa đi phần thân phủ định trong các quy tắc còn lại. Một *tập trả lời* của một chương trình *P* là một diễn dịch $I \subseteq HB_P$ mà *I* là một tập trả lời của P^I.

2.3. Logic mô tả $\mathcal{SHIF}$ (D) và $\mathcal{SHOIN}$ (D) :

2.3.1. Cú pháp. Cho tập **E** là *các kiểu dữ liệu cơ bản* và tập **V** gồm *các giá trị dữ liệu*. Một *lí thuyết kiểu dữ liệu* $\mathbf{D} = (\Delta^{\mathbf{D}}, \cdot^{\mathbf{D}})$ bao gồm một *miền kiểu dữ liệu* $\Delta^{\mathbf{D}}$ và một ánh xạ $\cdot^{\mathbf{D}}$ nó ánh xạ đến mỗi kiểu dữ liệu cơ bản một tập con của $\Delta^{\mathbf{D}}$ và đến mỗi giá trị dữ liệu một phần tử của $\Delta^{\mathbf{D}}$. Cho $\Psi = (\mathbf{A} \cup \mathbf{R}_A \cup \mathbf{R}_D, \mathbf{I} \cup \mathbf{V})$ là một bộ từ vựng, trong đó **A**, $\mathbf{R}_A$, $\mathbf{R}_D$, và **I** là lần lượt là các tập hợp (đếm được) đôi một phân biệt của các *khái niệm nguyên tố, các vai trò tượng trưng, các vai trò kiểu dữ liệu*, và *các cá thể*. Chúng ta ký hiệu R_A^- là tập hợp các nghịch đảo R^- của tất cả $R \in \mathbf{R}_A$.

Vai trò là một phần tử của $\mathbf{R}_A \cup R_A^- \cup \mathbf{R}_D$. *Khái niệm* được định nghĩa quy nạp như sau. Mỗi khái niệm nguyên tố $C \in \mathbf{A}$ là một khái niệm. Nếu $o_1, o_2,...$ là các cá thể thuộc **I** thì $\{o_1, o_2,..\}$ là một khái niệm. Nếu *C* và *D* là những khái niệm, thì $(C \sqcap D)$, $(C \sqcup D)$, và $\neg C$ cũng là những khái niệm (lần lượt được gọi là *phép giao, phép giao*, và *phủ định*). Nếu *C* là một khái niệm, *R* là một vai trò tượng trưng từ $\mathbf{R}_A \cup R_A^-$ và *n* là một số nguyên không âm, thì $\exists R.C$, $\forall R.C$, $\leq nR$, và $\geq nR$ là những khái niệm (lần lượt được gọi là *lượng từ tồn tại, lượng từ với mọi, giới hạn số lượng lớn nhất*, và *giới hạn số lượng nhỏ nhất*). Nếu *D* là một kiểu dữ liệu, *U* là một vai trò kiểu dữ liệu từ $\mathbf{R}_D$, và *n* là một số nguyên không âm, thì $\exists U.D$, $\forall U.D$, $\leq nU$, $\geq nU$ là những khái niệm. Chúng ta sử dụng $\top$ và $\bot$ để lần lượt viết tắt cho các khái niệm $C \sqcup \neg C$ *và* $C \sqcap \neg C$.

Một *tiên đề* là một biểu thức có một trong các dạng sau đây:

(1) $C \sqsubseteq D$, được gọi là *tiên đề bao hàm khái niệm*, trong đó *C* và *D* là những khái niệm;

(2) $R \sqsubseteq S$, được gọi là *tiên đề bao hàm vai trò*, với hoặc R, S $\in \mathbf{R}_A$ hoặc R, S $\in \mathbf{R}_D$;

(3) Trans(*R*), gọi là *tiên đề bắc cầu*, trong đó $R \in \mathbf{R}_A$;

(4) *C(a)* được gọi là *tiên đề thành viên khái niệm*, trong đó *C* là một khái niệm và $a \in \mathbf{I}$;

(5) *R(a,b)* (*hoặc U(a,v)*), được gọi là *tiên đề thành viên vai trò*, trong đó $R \in \mathbf{R}_A$ (hoặc $U \in \mathbf{R}_D$) và $a,b \in \mathbf{I}$ (hoặc $a \in \mathbf{I}$ và *v* là một giá trị dữ liệu);

(6) $a = b$ (hoặc $a \neq b$), gọi là *tiên đề đẳng thức (hoặc bất đẳng thức)*, trong đó $a, b \in \mathbf{I}$.

Ta cũng sử dụng $F \equiv G$ để rút gọn hai tiên đề khái niệm hoặc vai trò $F \sqsubseteq G$ và $G \sqsubseteq F$. Một *cơ sở tri thức logic mô tả L* là một tập hữu hạn các tiên đề.

Ví dụ 1. Giả sử chúng ta muốn phân công người phản biện các bài báo, dựa trên một số thông tin đã biết về các bài báo và những người đã biết trước, được mã hóa trong cơ sở tri thức logic mô tả L_S. L_S phân loại các bài báo vào các lĩnh vực nghiên cứu. Một lĩnh vực nghiên cứu thuộc về một khái niệm *Linhvuc*. Một *Baibao* được phân loại tùy thuộc vào thông tin từ khóa. Vai trò *tukhoa* và *trongLinhvuc* kết hợp với mỗi bài báo thông qua các từ khóa và các lĩnh vực mà nó được phân loại vào. Một bài báo thuộc về một lĩnh vực nếu nó kết hợp với một từ khóa của lĩnh vực đó. Vai trò *chuyengia* liên quan đến những người mà lĩnh vực của họ rất thành thạo. Để đơn giản, một người là một chuyên gia trong một lĩnh vực nếu người đó đã viết một bài báo trong lĩnh vực đó. Khái niệm *Trongtai* gồm tất cả các trọng tài. Vai trò *baogom* kết hợp từng lĩnh vực với một nhóm các từ khóa, trong khi đó vai trò *coThanhvien* liên quan đến các nhóm từ khóa có liên quan đến nhau. Sau đây là một số tiên đề trong L_S:

$Baibao \sqsubseteq Anban; Trongtai \sqsubseteq Nguoi;$

$Baibao(anban_1);Trongtai(nguoi_1);Trongtai(nguoi_2);$

$coThanhvien(C_1,SemanticWeb); Tacgia(anban_1,nguoi_2);$

$Linhvuc(A);Linhvuc(B);Linhvuc(C);Linhvuc(D);Linhvuc(E);$

$baogom(A,Tactu); tukhoa(anban_1,Tactu);$

$\exists trongLinhvuc.\{c\} \equiv \exists tukhoa.(\exists baogom^{-}.\{c\}),$ với mọi $c \in \{A,B,C,D,E\}$

$\exists chuyengia.\{c\} \equiv \exists tacgia^{-}.(\exists trongLinhvuc.\{c\}),$ với mọi $c \in \{A,B,C,D,E\}$

2.3.2. Ngữ nghĩa: Bây giờ chúng ta định nghĩa ngữ nghĩa dựa vào phép diễn dịch bậc nhất và chúng ta cũng xem lại một và vấn đề suy luận quan trọng trong các logic mô tả.

Một *diễn dịch* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ đối với một lý thuyết kiểu dữ liệu **D** = $(D^{D}, \cdot^{D})$ bao gồm một miền không rỗng $\Delta^{\mathcal{I}}$ phân biệt với miền D^{D} và một ánh xạ $\cdot^{\mathcal{I}}$ nó ánh xạ mỗi khái niệm nguyên tố $C \in \mathbf{A}$ với một tập con của $\Delta^{\mathcal{I}}$, mỗi cá thể $o \in \mathbf{I}$ với một phần tử của $\Delta^{\mathcal{I}}$, mỗi vai trò tượng trưng $R \in \mathbf{R}_A$ với một tập con của $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, và mỗi vai trò kiểu dữ liệu $U \in \mathbf{R}_D$ với một tập con của $\Delta^{\mathcal{I}} \times D^{D}$. Ánh xạ $\cdot^{\mathcal{I}}$ được mở rộng cho tất cả các khái niệm và vai trò như sau:

$(R^{-})^{\mathcal{I}} = \{(a,b) \mid (b,a) \in R^{\mathcal{I}}\};$

$\{o_1,\ldots,o_n\}^{\mathcal{I}} = (o_1^{\mathcal{I}},\ldots,o_n^{\mathcal{I}}\}$

$(C \sqcap \mathcal{D})^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$; $(C \sqcup \mathcal{D})^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$; $(\neg C)^{\mathcal{I}} = \triangle^{\mathcal{I}} \backslash C^{\mathcal{I}}$;

$(\exists R.C)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \exists y : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$;

$(\forall R.C)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \forall y : (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$;

$(\geq nR)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\} \geq n)$;

$(\leq nR)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in R^{\mathcal{I}}\} \leq n)$;

$(\exists U.D)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \exists y : (x,y) \in U^{\mathcal{I}} \wedge y \in D^{D})$;

$(\forall U.D)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \forall y : (x,y) \in U^{\mathcal{I}} \rightarrow y \in D^{D})$;

$(\geq nU)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in U^{\mathcal{I}}\} \geq n)$

$(\leq nU)^{\mathcal{I}} = \{x \in \triangle^{\mathcal{I}} \mid \#(\{y \mid (x,y) \in U^{\mathcal{I}}\} \leq n)$;

Tính *thỏa mãn* của một tiên đề logic mô tả F trong diễn dịch $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ đối với **D** = $(D^{D}, \cdot^{D})$ ký hiệu $\mathcal{I} \models F$ được định nghĩa như sau :

(1) $\mathcal{I} \models \mathcal{C} \sqsubseteq \mathcal{D}$ khi và chỉ khi $\mathcal{C}^{\mathcal{I}} \subseteq \mathcal{D}^{\mathcal{I}}$

(2) $\mathcal{I} \vDash \mathcal{R} \sqsubseteq \mathcal{S}$ khi và chỉ khi $\mathcal{R}^{\mathcal{I}} \subseteq \mathcal{S}^{\mathcal{I}}$

(3) $\mathcal{I} \vDash$ Trans(*R*) khi và chỉ khi $R^{\mathcal{I}}$ có tính bắc cầu.

(4) $\mathcal{I} \vDash$ *C(a)* khi và chỉ khi $a^{\mathcal{I}} \in \mathcal{C}^{\mathcal{I}}$

(5) $\mathcal{I} \vDash$ *R(a,b)* khi và chỉ khi $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in \mathcal{R}^{\mathcal{I}}$ (hoặc $\mathcal{I} \vDash$ *U(a,v)* khi và chỉ khi $(a^{\mathcal{I}}, v^{D}) \in U^{\mathcal{I}}$)

(6) $\mathcal{I} \vDash$ *a* = *b* khi và chỉ khi $a^{\mathcal{I}} = b^{\mathcal{I}}$ (hoặc $\mathcal{I} \vDash a \neq b$ khi và chỉ khi $a^{\mathcal{I}} \neq b^{\mathcal{I}}$)

Phép diễn dịch $\mathcal{I}$ *thỏa mãn* một tiên đề *F* hay $\mathcal{I}$ là một *mô hình* của *F* khi và chỉ khi $\mathcal{I} \vDash F$. Một diễn dịch $\mathcal{I}$ *thỏa mãn* một cơ sở tri thức logic mô tả *L* hay $\mathcal{I}$ là một *mô hình* của *L*, ký hiệu là $\mathcal{I} \vDash L$ khi và chỉ khi $\mathcal{I} \vDash F$ với mọi $F \in L$. Ta nói *L* là *thỏa mãn* (hoặc *không thỏa mãn*) khi và chỉ khi *L* có (hoặc không) mô hình. Một tiên đề *F* là *một hệ quả logic* của *L*, ký hiệu $F \vDash L$ khi và chỉ khi mọi mô hình của *L* thỏa mãn *F*. Một tiên đề phủ định $\neg F$ là một *hệ quả logic* của *L*, ký hiệu là $L \vDash \neg F$ khi và chỉ khi mọi mô hình của *L* không thỏa mãn *F*.

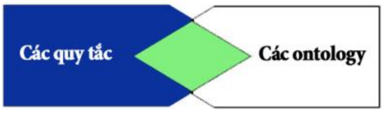
3. CHIẾN LƯỢC KẾT HỢP CÁC QUY TẮC VÀ ONTOLOGY

Có hai chiến lược để kết hợp các quy tắc và các ontology phố biến hiện nay là: tích hợp chặt chẽ ngữ nghĩa và tích hợp linh hoạt ngữ nghĩa.Bây giờ chúng ta sẽ phân tích các nguyên tắc của hai chiến lược và đánh giá công việc liên quan đến chúng.

3.1. Tích hợp chặt chẽ ngữ nghĩa : Trong chiến lược này, các quy tắc được giới thiệu trực tiếp trong lớp ontology, tức là các tên khái niệm và vai trò có thể được sử dụng như là các tên vị từ trong các quy tắc. Cách tiếp cận như vậy có thể dễ dàng dẫn đến khả năng không quyết định được, ví dụ CARIN và SWRL. Mặt khác, DPL được đề xuất bởi Grosof (2003) bảo tồn khả năng quyết định rất hạn chế trong cú pháp của nó, do đó hạn chế khả năng biểu diễn. SWRL và DLP có thể được xem như hướng đi cho phép một số lượng lớn cho cách tiếp cận khác để phù hợp, chẳng hạn như $\mathcal{DL}$-log, các quy tắc DL-safe, r-hybrid KBs, và $\mathcal{DL}$+log. Những cách tiếp cận, để duy trì khả năng quyết định và mở rộng khả năng diễn đạt, đòi hỏi các ràng buộc điều kiện an toàn của các biến trong các quy tắc.[7]

3.2. Tích hợp linh hoạt ngữ nghĩa: Trong phương pháp này, các quy tắc và các ontology (OWL/RDF) đóng vai trò trong các lĩnh vực khác nhau. Trong khi các quy tắc tập trung vào công việc lý luận thì OWL/RDF nhằm tăng mục đích của chúng trong vai trò là ngôn ngữ mô tả. Hai thành phần này không bị buộc bởi bất kỳ ràng buộc cú pháp nào, miễn là bên riêng mỗi chúng có khả năng quyết định, và sau đó chúng giao tiếp với nhau thông qua một "giao diện an toàn".

Nhìn từ quan điểm lớp các quy tắc, các ontology phục vụ như một nguồn thông tin mở rộng với một ngữ nghĩa độc lập có thể được cập nhật hoặc truy vấn thông qua một vị từ đặc biệt. Cách tiếp cận như vậy là chương trình logic mô tả [Eiter và cộng sự, 2005, 2007, Łukasiewicz, 2005, Eiter và cộng sự, 2008] và công cụ thực thi các quy tắc TRIPLE [SINTEK và Decker, 2002] chúng gọi các bộ suy luận logic mô tả bên ngoài.Chiến lược này thu hút sự quan tâm rất lớn của các nhà nghiên cứu trong những năm gần đây, trong nội dung của báo cáo chúng ta sẽ tập trung vào các vấn đề của việc tích hợp này bằng cách tiếp cận chương trình logic mô tả [4].



Hình 1. Tích hợp chặt chẽ.

Hình 2. Tích hợp linh hoạt.

4. CHƯƠNG TRÌNH LOGIC MÔ TẢ

4.1. Cú pháp : Một chương trình logic mô tả chứa một cơ sở tri thức logic mô tả *L* và một chương trình chuẩn tắc tổng quát *P*. *P* là một tập hợp hữu hạn các quy tắc tổng quát có thể chứa các truy vấn đến *L* trong phần thân của chúng. Trong một truy vấn như vậy sẽ hỏi liệu tiên đề logic mô tả đã chắc chắn chưa hoặc phủ định của nó có hợp logic theo *L* không.

Một cơ sở tri thức mô tả *L* được định nghĩa thông qua một bộ từ vựng $\Psi = (\mathbf{A}\cup\mathbf{R}_A\cup\mathbf{R}_D, \mathbf{I}\cup\mathbf{V})$ và chương trình chuẩn tắc tổng quát *P* được định nghĩa thông qua một bộ từ vựng $\Phi = (\mathcal{P}, \mathcal{C})$ đã được giới thiệu ở mục trên, chúng tả giả sử rằng $\mathbf{A}\cup\mathbf{R}_A\cup\mathbf{R}_D$ rời trong $\mathcal{P}$ trong khi $I_P \subseteq \mathcal{C} \subseteq \mathbf{I}\cup\mathbf{V}$ trong đó I_P là tập tất cả các ký hiệu hằng xuất hiện trong *P*.

Bây giờ ta định nghĩa các khái niệm *truy vấn logic mô tả (dl-queries) và nguyên tố logic mô tả (dl-atoms)*, hai đối tượng này được sử dụng trong thân quy tắc để biểu diễn các truy vấn trong cơ sở tri thức mô tả *L*. *Một truy vấn logic mô tả (dl—query) Q(**t**)* thuộc một trong các dạng sau :

(a) hoặc là một tiên đề bao hàm khái niệm *F* hoặc phủ định của nó $\neg F$;
(b) hoặc có dạng biểu thức $C(t)$ hoặc $\neg C(t)$ trong đó *C* là một khái niệm và *t* là hạng thức;
(c) hoặc là có dạng $R(t_1,t_2)$ hoặc $\neg R(t_1,t_2)$ trong đó *R* là vai trò và t_1, t_2 là các hạng thức;
(d) hoặc là có dạng biểu thức $=(t_1,t_2)$ hoặc $\neq(t_1,t_2)$ trong đó t_1, t_2 là các hạng thức;

Ở đây chú ý rằng **t** là một danh sách tham số mà nó rỗng trong (a), $\mathbf{t} = t$ trong (b), $\mathbf{t} = (t_1, t_2)$ trong (c) và (d); và các hạng thức được định nghĩa tương tự ở trên. *Một nguyên tố logic mô tả* có dạng: $DL[S_1op_1p_1,..., S_mop_mp_m; Q](\mathbf{t})$ *với* $m\geq 0$ *(4)*

Trong đó mỗi S_i hoặc là một khái niệm hoặc là một vai trò hoặc là một ký tự đặc biệt $\theta\in\{=,\neq\}$; $op_i \in\{\uplus, \cup^{-}, \cap^{-}\}$, p_i là ký hiệu vị từ một ngôi khi S_i là một khái niệm và một ký hiệu vị từ 2 ngôi trong trường hợp khác và $Q(\mathbf{t})$ là một dl-query. Chúng ta gọi $p_i, ..., p_m$ là các *ký hiệu vị từ đầu vào* của chúng. Ta có $op_i = \uplus$ (hoặc $op_i = \cup^{-}$) tăng S_i (hoặc $\neg S_i$) bằng việc thêm p_i trong khi $op_i = \cap^{-}$ ràng buộc S_i với p_i.

Một *dl-rule r* có dạng ($a \leftarrow b_1, b_2,..., b_k$, *not* $b_{k+1},...,$ *not* b_m *với* $m\geq k\geq 0$) trong đó bất kỳ literal $b_1, b_2,..., b_m \in B(r)$ có thể là một dl-atom. Một *chương trình logic mô tả KB = (L, P)* gồm một cơ sở tri thức mô tả *L* và một tập hữu hạn các quy tắc logic mô tả *P*.

4.2. Ngữ nghĩa: Trước khi xác định ngữ nghĩa của các chương trình logic mô tả, đầu tiên chúng ta đưa ra một ví dụ trực quan sau đó chúng ta giới thiệu các mô hình Herbrand của chương trình logic mô tả.Nói chung chương trình logic mô tả là một phương tiện để nối hai nguồn tri thức, cụ thể là một cơ sở kiến thức logic mô tả và một chương trình logic

Ví dụ 2: (Lựa chọn người phản biện) Cho $KB_S = (L_S, P_S)$ là chương trình logic mô tả bao gồm cơ sở tri thức logic mô tả L_S được mô tả trong ví dụ 1, và chương trình logic mô tả P_S:

(1) Baibao(p_1); kw(p_1, SemanticWeb);

(2) Baibao(p_2); kw(p_2, Bioinformatics);

(3) kw(p_2, Answer_Set_Programing);

(4) kw(P,K_2) ← kw(P,K_1), DL[coThanhvien](S,K_1), DL[coThanhvien](S,K_2);

(5) baibaoLinhvuc(P,A ← DL[tukhoa ⊎ kw;trongLinhvuc](P, A);

(6) ungcuvien(X, P) ← baibaoLinhvuc(P,A), DL[Trongtai](X), DL[chuyengia](X, A).

(7) phancong(X,P) ← ungcuvien(X, P), not ¬ phancong(X,P);

(8) ¬ phancong(Y,P ← ungcuvien(Y, P), phancong(X,P), X≠Y;

(9) a(P) ← phancong(X,P);

(10) loi(P) ← Baibao(P), not a(P).

Các ngữ nghĩa hình thức kết hợp KB_S với một bộ của các *tập trả lời*, giống như đối với trường hợp của các chương trình logic thông thường. Mục đích của P_S là để xác định làm thế nào các tập câu trả lời sẽ giống như thế.

Các dữ kiện (1) đến (3) mô tả hai bài báo p_1 và p_2 để được đăng ký những người phản biện với những từ khóa của chúng. Quy tắc (4) chúng ta lấy thông tin từ khóa từ L_S. Cụ thể hơn vị từ *kw* được bổ sung thông qua các nguyên tố logic mô tả bằng các từ khóa của chúng trong L_S mà chúng chia sẽ cùng lĩnh vực. Một cách trực quan, nguyên tố logic mô tả nền *DL[coThanhvien](s,k)* đúng cho tất cả các cặp *(s,k)* mà $L_S \vDash$ *coThanhvien(s,k).* Quy tắc (5) chứa loại nguyên tố logic mô tả giàu ngữ nghĩa hơn, đầu tiên chúng làm giàu vai trò *tukhoa* trong L_S bằng mở rộng của vị từ *kw* trong P_S thông qua toán tử ⊎. Sau đó ta truy vấn vai trò *trongLinhvuc* thông qua phiên bản đã được sửa đổi L'_S của L_S (tức là chúng ta lấy từ L'_S những lĩnh vực mà mỗi bài báo được phân lớp vào trong đó) Ở đây thông tin mới đến từ *kw* có thể kích hoạt thông tin mới trong L'_S.

Trong quy tắc (6), chúng ta định nghĩa các ứng cử viên là người phản biện cho một bài báo. Cụ thể hơn, một người phản biện *x* là một ứng cử viên phản biện cho bài báo *p*, nếu *x* được biết đến trong L_S như là một *trongtai* và là một *chuyengia* trong lĩnh vực tài liệu tham khảo *a* của *p*. Quy tắc (7) và (8) mã hóa một sự lựa chọn không đơn định. Như chúng ta sẽ thấy, ngữ nghĩa của chúng ta đưa ra một ý nghĩa đặc biệt với các quy tắc mà nó xuất hiện trong các quy tắc đệ quy và liên quan đến việc phủ định. Một cách trực quan, cho bất kỳ cặp ứng cử viên *ungcuvien(x, p)* có thể nào, chúng ta đoán liệu *phancong(x, p)* là đúng hay không, có nghĩa là *p* phải được gán cho *x* phản biện hay không. Như vậy, có những câu trả lời mà *phancong(x, p)* là đúng, và có câu trả lời khác là sai. Cuối cùng, quy tắc (9) và (10) kiểm tra xem bài báo đã được đăng ký, nếu không một lỗi sẽ được đánh dấu; các câu trả lời *loi(p)* là đúng cho một *p* được cho thể được lọc ra. Lưu ý rằng các quy tắc (4) đến (6) các tri thức chuyển từ L_S đến P_S. Trong quy tắc (5), tri thức cũng được chuyển từ P_S đến L_S. Do đó, thông tin di chuyển theo cả hai hướng giữa cơ sở tri thức logic mô tả L_S và chương trình tổng quát P_S. Ý nghĩa trực quan của toán tử ⊎ là để bổ sung thông tin từ P_S các khẳng định có tính phủ định. Ví dụ, *tukhoa* ⊎ *kw* có nghĩa là L_S được mở rộng với các khẳng định ¬*tukhoa(k)* cho bất kỳ *kw(k)* đúng nào.

4.3. Mô hình của chương trình logic mô tả: Trước hết ta định nghĩa các diễn dịch Herbrand và tính thỏa của chương trình logic mô tả trong các diễn dịch Herbrand. Sau đó ta định nghĩa tính đúng đắn của các nguyên tố logic mô tả nền trong các diễn dịch Herbrand. Trong các khái niệm sau ta đều xét đến *KB = (L, P)* là chương trình logic mô tả với bộ từ vựng $\Phi = (\mathcal{P}, \mathcal{C})$ trong đó $\mathcal{C}$ là tập các ký hiệu hằng và $\mathcal{P}$ là tập các ký hiệu vị từ.

Cơ sở Herbrand của *P*, ký hiệu là HB_P, là tập tất cả các literal nền được xây dựng bằng các ký hiệu vị từ trong $\mathcal{P}$ xuất hiện trong *P* và các ký hiệu hằng trong $\mathcal{C}$.

Một *diễn dịch I* liên quan với *P* là một tập con nhất quán của HB_P. Ta gọi *I* la một *mô hình* của một literal nền hay nguyên tố logic mô tả nền *l* (hay *I* thỏa *l*) theo *L*, ký hiệu $P \models_L l$ nếu những điều sau đây thỏa :

- Nếu $l \in HB_P$ thì $I \models_L l$ khi và chỉ khi $l \in I$;
- Nếu *l* là một nguyên tố logic mô tả nền $DL[\lambda;Q](\mathbf{c})$, $\lambda = S_1 op_1 p_1, \dots, S_m op_m p_m$ thì $I \models_L l$ khi và chỉ khi $L(I;\lambda) \models Q(\mathbf{c})$ trong đó $L(I;\lambda) = L \cup \bigcup_{i=1}^{m} A_i(I)$ và $1 \le i \le m$, trong đó $A_i = \{S_i(e) \mid p_i(e) Î I\}$ *nếu* $op_i = ⊎$ hoặc $A_i = \{ØS_i(e) \mid p_i(e) Î I\}$ *nếu* $op_i = ∪$ hoặc $A_i = \{ØS_i(e) \mid p_i(e) \notin I\}$ *nếu* $op_i = ∩$ (trong đó $e = t_1, \dots, t_k$ với t_j là hằng và k là bậc của vị từ p_i).

Ta nói phép diễn dịch *I* là *mô hình* của quy tắc logic mô tả nền *r* nếu và chỉ nếu $I \models_L l$ với mọi $l \in B^+(r)$ và $I \not\models_L l$ với mọi $l \in B^-(r)$ dẫn đến $I \models_L H(r)$. Chúng ta nói phép diễn dịch *I* là *mô hình* của chương trình logic mô tả *KB=(L, P)* hay *I thỏa mãn KB*, ký kiệu $I \models KB$ nếu $I \models_L r$ với mọi $r \in ground(P)$. Ta nói *KB* là *thỏa mãn* (hoặc *không thỏa mãn*) nếu *KB* có (hoặc không có) mô hình.

Để ý rằng tính thỏa mãn ở trên của nguyên tố logic mô tả *a* trong phép diễn dịch *I* cũng liên quan đến các phủ định tiên đề bao hàm khái niệm $\neg(C \sqsubseteq D)$, phủ định tiên đề thành viên khái niệm $\neg C(a)$ và phủ định tiên đề thành viên vai trò $\neg R(a,b)$ và $\neg U(a, v)$. Với lí do này, ta mở rộng thêm một ít cú pháp và ngữ nghĩa thông thường của $\mathcal{SHIF}$(**D**) và $\mathcal{SHOIN}$(**D**) bằng cách cho phép những tiên đề phủ định như trên. Các khái niệm tính thỏa mãn, và tính kế thừa được mở rộng một cách hiển nhiên để xử lý các tiên đề như trên. Đặc biệt, phép diễn dịch $\mathcal{I}$ thỏa mãn $\neg(C \sqsubseteq D)$ (hoặc $\neg C(a)$, $\neg R(a,b)$, $\neg U(a, v)$) khi và chỉ khi $C^{\mathcal{I}} \nsubseteq D^{\mathcal{I}}$ (hoặc $a^{\mathcal{I}} \notin C^{\mathcal{I}}$, $(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin R^{\mathcal{I}}$, $(a^{\mathcal{I}}, v^{D}) \notin U^{\mathcal{I}}$).

Tính kế thừa (cho nguyên tố logic mô tả) trong các mở rộng yếu của $\mathcal{SHIF}$(**D**) và $\mathcal{SHOIN}$(**D**) có thể được rút gọn lần lượt thành tính kế thừa của $\mathcal{SHIF}$(**D**) và $\mathcal{SHOIN}$(**D**) tương ứng như sau. Đầu tiên ta chú ý, tính kế thừa của một bao hàm khái niệm, thành viên khái niệm, thành viên vai trò, hoặc tiên đề đẳng thức F (hoặc phủ định của nó $\neg F$) từ một cơ sở tri thức mô tả L là tương đương với tính không thỏa mãn $L\cup\{\neg F\}$ (hoặc $L\cup\{F\}$) Ở đây, phủ định tiên đề bao hàm khái niệm $\neg\ (C\sqsubseteq D)$ là tương đương với tiên đề thành viên khái niệm $(C\sqcap\neg D)(a)$ (trong đó a là một các thể mới), và phủ định tiên đề thành viên khái niệm $\neg(C(a))$ là tương đương với tiên đề thành viên khái niệm $(\neg C)(a)$. Sau đó, mỗi phủ định tiên đề thành viên vai trò trừu tượng trong một cơ sở tri thức logic mô tả L có thể được loại bỏ bằng cách sử dụng : $L' \cup \{\neg R(a,b)\}$ không thỏa mãn nếu và chỉ nếu $L'\cup\{A(a),B(b),\exists R.B\sqsubseteq\neg A\}$ không thỏa mãn (trong đó A và B là hai khái niệm nguyên tố mới và L' là một cơ sở tri thức mô tả bất kỳ) [6]. Các phủ nhận tiên đề vai trò kiểu dữ liệu thành viên có thể được loại bỏ trong một cách tương tự.

4.4. Ngữ nghĩa mô hình nhỏ nhất của chương trình logic mô tả dương: Bây giờ chúng ta định nghĩa chương trình mô tả dương, đó là chương trình logic mô tả không chứa phủ định mặc định và có liên quan đến nguyên tố logic mô tả đơn điệu. Cũng giống như các chương trình dương thông thường, mọi chương trình logic mô tả dương thỏa mãn có một mô hình nhỏ nhất duy nhất, đó cũng là đặc trưng ngữ nghĩa của nó.

Một nguyên tố logic mô tả nền a liên quan $KB = (L,P)$ là *đơn điệu* khi và chỉ khi $I \vDash_L a$ thì $I' \vDash_L a$ với mọi $I \subseteq I' \subseteq HB_P$; ngược lại *a không đơn điệu*. Chú ý rằng nguyên tố logic mô tả chỉ chứa toán tử $\uplus$ và $\cup$ luôn luôn đơn điệu. Một chương trình logic mô tả $KB = (L, P)$ *dương* khi và chỉ khi không có quy tắc nào trong P có chứa "*not*" và mọi nguyên tố logic mô tả nền trong *ground(P)* đều đơn điệu trong *KB*. Với chương trình dương P thông thường, phép giao hai mô hình của P cũng là một mô hình của P. Định lý sau chỉ ra kết quả tương tự đối với chương trình logic mô tả dương.

Định lý 1. *Cho $KB =(L,P)$ là chương trình logic mô tả dương. Nếu diễn dịch I_1, $I_2 \subseteq HB_P$ là các mô hình của KB thì $I_1 \cap I_2$ cũng là mô hình của KB.*

Chứng minh: Do $I_1, I_2 \subseteq HB_P$ là các mô hình của *KB* nên $I_i \vDash_L r$ với mọi $r \in ground(P)$ *và* $i \in\{1,2\}$. Chúng ta chỉ ra rằng $I = I_1 \cap I_2$ là một mô hình của *KB* tức là $I \vDash_L r$ với mọi $r \in ground(P)$. *Xét* r $\in$ *ground(P)* bất kỳ, giả sử $I \vDash_L l$ với mọi $l \in B^+(r)=B(r)$. Lúc đó, $I \vDash_L l$ cho mọi literal $l \in B(r)$ và $I \vDash_L a$ cho mọi nguyên tố logic mô tả $a \in B(r)$. *Vì vậy,* $I_i \vDash_L l$ với mọi $l \in B(r)$ *và* $i \in \{1,2\}$.Hơn thế nữa ta có mọi nguyên tố logic mô tả trong *ground(P)* là đơn điệu trong *KB*, điều này dẫn đến, $I_i \vDash_L a$ với mọi a $\in B(r)$ *và* $i \in\{1,2\}$. Do I_1, I_2 là các mô hình của *KB* nên, $I_i \vDash_L H(r)$ với mọi $i \in\{1,2\}$ dẫn đến $I \vDash_L H(r)$. Điều này chỉ ra rằng $I \vDash_L r$. *Vậy I* là mô hình của *KB*.

Ta sẽ có một quy tắc hệ quả tất yếu.

Hệ quả 2. *Cho KB =(L,P) là một chương trình logic mô tả dương. Nếu KB thỏa mãn thì tồn tại một mô hình duy nhất $I \subseteq HB_P$ của KB sao cho $I \subseteq J$ với mọi mô hình $J \subseteq HB_P$ của KB, tức I là mô hình nhỏ nhất duy nhất của KB.*

Mô hình đặc biệt này được xem là ngữ nghĩa của *KB*.

Định nghĩa 3. Với mỗi chương trình logic mô tả thỏa mãn KB = (L,P), chúng ta ký hiệu mô hình nhỏ nhất duy nhất của *KB* là M_{KB}.

Ví dụ 3. Cho *KB* là kết quả của chương trình logic mô tả KB_S trong ví dụ 3 bằng cách loại bỏ đi các quy tắc (7) – (10). Rõ ràng *KB* không chứa phủ định. Mặt khác vì các nguyên tố nền logic mô tả không chứa ∩ nên chúng đơn điệu. Vì vậy *KB* dương cho nên nó có mô hình nhỏ nhất duy nhất M_{KB} chứa tất cả các ứng cử viên người phản biện cho các bài báo p_1 và p_2 được cho.

4.5. Ngữ nghĩa mô hình nhỏ hồi quy nhất của chương trình logic mô tả phân tầng: Chương trình logic mô tả phân tầng được cấu tạo từ các lớp có phân cấp của chương trình logic mô tả dương. Giống như các chương trình phân tầng thông thường, một mô hình cực tiểu chính tắc của chương trình logic mô tả phân tầng có thể được chỉ ra bởi một số các mô hình hồi quy nhỏ nhất cung cấp cho một số mô hình tồn tại. Chúng ta có thể thực hiện điều này với các nguyên tố logic mô tả không đơn điệu bằng cách xử lý tương tự như literal phủ định ngầm. Điều này đặc biệt hữu ích, bởi vì nói chung ta không biết liệu một nguyên tố logic mô tả được cho là đơn điệu hay không, và việc xác định này có thể là tốn chi phí nhiều hay không. Tuy nhiên chú ý rằng, sự vắng mặt của toán tử ∩ trong (4) là một tiêu chuẩn cú pháp đơn giản dẫn đến tính đơn điệu của một nguyên tố logic mô tả (Ví dụ 3).Với chương trình logic mô tả *KB=(L,P)* bất kì, ta ký hiệu DL_P là tập tất cả các nguyên tố logic môt tả nền trong *ground(P)*. Ta giả thiết rằng *KB* được kết hợp bởi tập $DL_P^+ \subseteq DL_P$ của các nguyên tố nền đơn điệu và $DL_P^? = DL_P \setminus DL_P^+$ là tất cả các nguyên tố logic mô tả khác. Một *literal đầu vào* của một số nguyên tố logic mô tả $a \in DL_P$ là một literal nền với một vị từ đầu vào của *a* và các ký hiệu hằng trong Φ. Khái niệm về một phân tầng đối với chương trình logic mô tả định nghĩa là một phân hoạch có thứ tự tập hợp của tập hợp tất cả các nguyên tố nền và các nguyên tố logic mô tả nền như sau :

Định nghĩa 4. Cho *KB =(L,P)* là một chương trình logic mô tả. *Phép phân tầng* của *KB* (xét trong DL_P^+) là là một ánh xạ μ: $HB_P \cup DL_P \rightarrow \{0, 1,..., k\}$ sao cho (i) với $r \in ground(P)$, thì $\mu(H(r)) \geq \mu(l')$ với mọi $l' \in B^+(r)$ và $\mu(H(r)) > \mu(l')$ với mọi $l' \in B^-(r)$ (ii) $\mu(a) \geq \mu(l)$ với mỗi literal đầu vào *l* của mỗi $a \in DL_P^+$ và $\mu(a) > \mu(l)$ với mỗi literal đầu vào *l* của mỗi $a \in DL_P^?$. Ta gọi $k \geq 0$ là *độ dài* của μ. Với mỗi $i \in \{0,..., k\}$, ta định nghĩa chương trình logic mô tả KB_i là $(L,P_i) = (L,\{r \in ground(P) \mid \mu H(r))= i\})$ và HB_{P_i} (hoặc $HB_{P_i}^*$) là tập tất cả các $l \in HB_P$ sao cho $\mu(l)=i$ (hoặc $\mu(l) \leq i$).

Ta nói một chương trình logic mô tả *KB = (L,P)* là *phân tầng* nếu và chỉ nếu nó có một phép phân tầng μ với độ dài $k \geq 0$. Mô hình chính tắc của nó được xác định như sau.

Định nghĩa 5. Cho *KB= (L, P)* là một chương trình logic mô tả với phép phân tầng có độ dài $k \geq 0$. Ta định nghĩa các mô hình hồi quy nhỏ nhất của nó $M_i \subseteq HB_P$ với $i \in \{0,..., k\}$ bằng: (i) M_0 là mô hình nhỏ nhất của KB_0; (ii) nếu $i > 0$ thì M_i là tập con nhỏ nhất *M* của HB_P sao cho *M* là mô hình của KB_i và $M \cap HB_{P_{i-1}}^* = M_{i-1} \cap HB_{P_{i-1}}^*$

Ta gọi KB là *nhất quán* nếu mọi M_i với $i \in \{0,..., k\}$ tồn tại và KB *không nhất quán* nếu ngược lại. Nếu KB nhất quán thì M_{KB} ký hiệu mô hình chính tắc là M_K. Chú ý rằng M_{KB} là xác định đúng đắn, nó không phụ thuộc vào μ.

Định lý 6. *Cho $KB = (L,P)$ là một chương trình logic mô tả phân tầng, thì M_{KB} là mô hình cực tiểu của KB.*

Chứng minh : Cho π: $HB_P \cup DL_P \rightarrow \{0, 1,..., k\}$ là một phân tầng của $KB=(L,P)$ trong DL_P^+. Ta có M_0 là mô hình nhỏ nhất của $KB_0 = (L_0,P_0)$ và với mọi $i \in \{0,..., k\}$. Ta có Mi là mô hình nhỏ nhất của $KB_i = (L_i,P_i)$ có $M_i \cap HB^*_{P_{i-1}} = M_{i-1} \cap HB^*_{P_{i-1}}$. Do đó $M_k = M_{KB}$ là mô hình của KB. Tiếp theo chúng ta sẽ chỉ ra rằng M_k là mô hình nhỏ nhất của KB. Để dẫn đến mâu thuẫn, ta giả sử rằng tồn tại một ô hình $J \subseteq HB_P$ của KB mà $J \subset M_k$. Do đó, tồn tại $i \in \{0,..., k\}$ sao cho $J \cap HB^*_{P_i} \neq M_{i-1} \cap HB^*_{P_i}$. Cho j là một nhỏ nhất như i. Lúc đó J là mô hình của KBj . Hơn nữa nếu $j > 0$ ta có $J \cap HB^*_{P_{j-1}} = M_k \cap HB^*_{P_{j-1}}$. Điều này mâu thuẫn với M_j là mô hình nhỏ nhất của KBj có $M_j \cap HB^*_{P_{j-1}} = M_{j-1} \cap HB^*_{P_{j-1}}$. Điều này chỉ ra rằng M_k là mô hình nhỏ nhất của KB.

Ví dụ 4. Cho KB là chương trình logic mô tả được rút ra từ KB_S trong ví dụ 3 bằng cách loại bỏ đi các quy tắc (11) và (12). Chương trình này có phép phân tầng với độ dài là 2 với tập thành phần DL_P^+ gồm tất cả các nguyên tố logic mô tả có trong P. Mô hình nhỏ nhất của P chứa tất cả các ứng cử viên làm người phê bình của các bài báo đã cho, cùng với các cờ lỗi của chúng bởi vì không có bài báo nào được đăng ký trước đó.

5. KẾT LUẬN

Chúng ta đã trình bày các chiến lược tích hợp các quy tắc và các ontology phổ biến hiện nay và đi sâu vào hướng tiếp cận chương trình logic mô tả, trong đó bao gồm một cơ sở tri thức logic mô tả L và một tập các quy tắc logic mô tả P mà chúng cũng có thể chứa các truy vấn đến L trong thân của chúng và cho phép sử dụng của phủ định không đơn điệu. Điều này tạo điều kiện như một cầu nối liên kết thế giới rất đa dạng của logic mô tả và lập trình logic và hơn nữa cung cấp một cơ sở ngữ nghĩa cho một sự kết hợp của các công cụ lý luận có sẵn từ các lập trình logic với các các cộng đồng logic mô tả.

Với tinh thần của lập trình logic, chúng ta đã xác định các mô hình Herbrand cho chương trình logic mô tả, và chúng ta tổng quát hóa nhiều khái niệm trong lập trình logic vào trong chương trình logic mô tả, bao gồm mô hình nhỏ nhất, phép phân tầng và tập trả lời. Sau đó chúng ta đã khái quát các lớp của các chương trình logic dương và chương trình logic phân tầng và xác định ngữ nghĩa một mô hình Herbrand cho chúng. Chúng ta cũng chỉ ra rằng tính thỏa mãn của chương trình logic mô tả dương có một mô hình Herbrand nhỏ nhất, và tính thỏa mãn của một chương trình logic mô tả phân tầng có thể được kết hợp với duy nhất một mô hình Herbrand cực tiểu, mà nó được đặc trưng thông qua một số hữu hạn các mô hình Herbrand hồi quy nhỏ nhất. Ngoài ra chúng ta cũng đã chứng minh giao của hai diễn dịch là mô hình nếu bản thân các diễn dịch là mô hình trong chương trình mô tả dương.

TÀI LIỆU THAM KHẢO

[1]. T. Berners-Lee (1999). *Weaving the Web*, Harper, San Francisco, CA.

[2]. T. Berners-Lee, J. Hendler, O. Lassila (2001). *The Semantic Web*, Scientific American 34–43.

[3]. D. Fensel, W. Wahlster, H. Lieberman, J. Hendler (2002). *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*, MIT Press.

[4]. I. Horrocks, P.F. Patel-Schneider (2003). Reducing OWL entailment to description logic satisfiability, in: *Proceedings ISWC-2003, in: LNCS, vol. 2870, Springer*, pp. 17–29.

[5]. Thomas Eiter, Thomas Lukasiewicz, Roman Schindlauer, and Hans Tompits (2004). Combining Answer Set Programming with Description Logics for the Semantic Web. In Didier Dubois, Christopher Welty, and Mary-Anne Williams, editors, *Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR 2004), Whistler, British Columbia, Canada,* pages 141-151. AAAI Press.

[6]. T. Lukasiewicz (2007). A novel combination of answer set programming with description logics for the Semantic Web, in: *Proceedings ESWC-2007, vol. 4519, Springer*, pp. 384–398.

[7]. Thomas Eiter, Giovambattista Ianni, Axel Polleres, Roman Schindlauer, Hans Tompits (2006). Reasoning with Rules and Ontologies, *Lecture Notes in Computer Science*, *vol. 4126*, *Springe*, pp 93-127.

DEVELOPING DESCRIPTION LOGIC PROGRAMS FOR INTEGRATING RULES AND ONTOLOGIES IN SEMANTIC WEB

Hoang Nguyen Tuan Minh

Office for Student Affairs, Hue University College of Sciences

Email: hntminh83@yahoo.com

ABSTRACT

An important requirement of the layered architecture of the Semantic Web is to integrate the rules and ontologies in development of Semantic Web. This issue has attracted researchers of the semantic web community in recent years. In the article, we consider the technical issues in the integration of rules and the current ontologies as well as classify representative proposals for different theoretical integration approaches. Furthermore, we focus on description logic programs approach underlying the integration of the rules and ontologies for semantic Web with its related problems.

Keyworks: *Logic programing, Description logics, Ontology, Semantic Web.*