Tạp chí Khoa học Đại học Huế: Kỹ thuật và Công nghệ; pISSN 2588-1175 | eISSN 2615-9732

Tập 131, Số 2A, 2022, Tr. 191–201; DOI: 10.26459/hueunijtt.v131i2A.6768

MỘT PHƯƠNG PHÁP ĐỊNH VỊ TIÊN ĐỀ CẦN XÓA ĐỂ BẢO ĐẢM TÍNH NHẤT QUÁN TRONG QUÁ TRÌNH TIẾN HÓA ONTOLOGY

Nguyễn Văn Trung*, **Nguyễn Thị Bích Lộc**

Khoa Công nghệ Thông tin, Trường Đại học Khoa học, Đại học Huế, Việt Nam

Tóm tắt. Ontology biểu diễn tri thức của một lĩnh vực cụ thể và đóng vai trò quan trọng trong các ứng dụng Web ngữ nghĩa. Khi có sự thay đổi về tri thức của lĩnh vực thì ontology cũng phải được thay đổi – gọi là tiến hóa ontology – thông qua việc bổ sung, chỉnh sửa hoặc xóa các tiên đề để phản ánh tri thức mới. Trong trường hợp ontology hiện tại không nhất quán khi bổ sung các tiên đề mới, người ta sẽ tìm cách chỉnh sửa hoặc xóa một số tiên đề từ ontology hiện tại. Bài báo này đề xuất phương pháp khoanh vùng tập tiên đề cần xóa, sau đó từ tập tiên đề này lựa chọn chính xác tiên đề cần xóa để xây dựng ontology nhất quán trong quá trình tiến hóa ontology.

Từ khóa: tiến hóa ontology, xử lý tri thức không nhất quán, khoảng cách ngữ nghĩa

Localizing and resolving inconsistency in ontology evolution

Nguyen Van Trung*, Nguyen Thi Bich Loc

Department of Information Technology, University of Sciences, Hue University, Vietnam

Abstract. An ontology captures domain knowledge and plays an essential role in Semantic Web applications. When domain knowledge changes, an ontology should adapt to changes by adding, editing or deleting its axioms. The big challenge of this process – called ontology evolution – is how to guarantee the consistent state of an ontology. If the ontology is inconsistent because new axioms are added, one will attempt to modify or delete some axioms from the existing ontology. This paper proposes a method to localize the set of axioms to be deleted for building a consistent ontology in ontology evolution.

Keywords: ontology evolution, inconsistency, semantic web application

** Liên hệ: nvtrung@husc.edu.vn*

Nhận bài: 18–04–2022; Ngày nhận đăng: 25–04–2022

1 Mở đầu

Ontology đóng vai trò quan trọng trong sự phát triển của Web ngữ nghĩa. Một ontology chứa một tập các thực thể (bao gồm khái niệm, thuộc tính, cá thể) và mối quan hệ giữa chúng, được thể hiện dưới dạng các tiên đề theo một cú pháp ngôn ngữ ontology cụ thể. Ngôn ngữ ontology thông dụng nhất ngày nay được tổ chức W3C là OWL2 chuẩn hóa và khuyến nghị sử dụng. Ngôn ngữ này dựa trên Logic Mô tả SROIQ(D) [3]. Bằng cách sử dụng các bộ lập luận, người ta có thể truy vấn và nhận được thông tin không được phát biểu tường minh trong ontology. Đây chính là lợi ích logic của ontology so với các giải pháp biểu diễn và lưu trữ cơ sở tri thức khác.

Ontology biểu diễn tri thức cho một lĩnh vực cụ thể trong thế giới thực thường có nhu cầu thay đổi theo thời gian để phản ánh sự thay đổi tương ứng trong thực tế. Quá trình thay đổi này được gọi là tiến hóa ontology. Trong kỹ nghệ ontology, bài toán tiến hóa tập trung giải quyết các vấn đề [8]: (i) cho phép quản lý được yêu cầu thay đổi ontology; (ii) bảo đảm tính nhất quán của ontology; (iii) quản lý (truy vết) các thay đổi của ontology; (iv) đưa ra gợi ý cho người dùng về các thay đổi tiếp theo của ontology.

Thách thức quan trọng nhất của quá trình tiến hóa ontology là đảm bảo ontology được nhất quán. Nếu ontology là không nhất quán, mọi tiên đề đều là hệ quả logic của nó. Trong trường hợp này, ontology bị mất đi lợi ích logic của một cơ sở tri thức.

Tồn tại hai chiến lược để đảm bảo tính nhất quán của ontology trong quá trình tiến hóa [2]. Chiến lược thứ nhất là xây dựng một tập các quy tắc về cấu trúc và ngữ nghĩa để áp dụng trong quá trình phát triển ontology nhằm bảo đảm ontology không chứa mâu thuẫn [1]. Những thao tác chỉnh sửa ontology vi phạm tập quy tắc này sẽ không được thực hiện. Chiến lược thứ hai là, chấp nhận mọi thao tác chỉnh sửa ontology trong quá trình tiến hóa. Hiển nhiên, trong quá trình tiến hóa, các tiên đề mới được thêm vào có thể làm cho ontology trở nên không nhất quán. Để duy trì tính nhất quán, một số tiên đề trong ontology hiện tại cần được xóa hoặc chỉnh sửa. Theo cách tiếp cận này, quá trình tiến hóa ontology được thực hiện qua hai bước [4, 7]. Bước thứ nhất tìm ra ontology con không nhất quán – chứa các tiên đề cần xóa (gọi là định vị sự không nhất quán). Bước thứ hai là chọn tiên đề trong ontology con vừa tìm được để thực hiện xóa để thu được ontology kết quả.

Bài báo này tập trung phân tích và đề xuất phương pháp bảo đảm tính nhất quán trong quá trình tiến hóa ontology theo hướng chấp nhận mọi thao tác chỉnh sửa ontology. Ở các công trình theo nhóm này, *trong bước thứ nhất*, việc định vị sự không nhất quán được thực hiện dựa vào các khái niệm không thoả được của ontology sau khi bổ sung tiên đề mới. Với mỗi khái niệm không thoả được như thế, người ta có thể xây dựng các ontology con không nhất quán cực tiểu

(gọi là MUPS – minimal unsatisfiability-preserving sub-TBox) [7]. Ontology con MUPS có đặc điểm khi xóa một tiên đề bất kỳ từ nó thì thu được ontology nhất quán. Chú ý rằng, mỗi khái niệm không thoả được có thể có nhiều hơn một ontology con MUPS. Ngoài ra, có thể có nhiều hơn một khái niệm không thoả được từ ontology không nhất quán. Điều này dẫn đến các câu hỏi: (1) làm thế nào để xác định thứ tự xử lý các khái niệm không thoả được (nếu có nhiều hơn một khái niệm như vậy); (2) nếu có nhiều ontology con MUPS của khái niệm này, chọn MUPS nào để xử lý tiếp theo trong bước thứ hai?

Ở *bước thứ hai*, sau khi đã định vị được sự không nhất quán, cần phải xác định một (hoặc một số) tiên đề phải xóa để thu được ontology kết quả. Với ontology con MUPS đã xác định được, việc xóa bất kỳ tiên đề nào cũng thu được ontology nhất quán. Theo [4] thì quyết định lựa chọn tiên đề để xóa được dành cho người dùng. Điều này là không thích hợp đặc biệt với tình huống MUPS có số lượng tiên đề lớn.

Trong bài báo này, chúng tôi đề xuất phương án giải quyết ở cả hai bước nêu trên của quá trình tiến hóa ontology. Để định vị sự không nhất quán, chúng tôi sử dụng chiến lược mở rộng tuyến tính tập tiên đề dựa trên mối quan hệ liên quan về cú pháp. Chiến lược này được Huang và cs. [5] đề xuất để sử dụng trong trong khung lập luận với ontology không nhất quán nhằm xây dựng tập tiên đề diễn giải truy vấn. Chúng tôi nhận thấy rằng, việc xây dựng tập tiên đề diễn giải này cũng có sự tương đồng với quá trình định vị sự không nhất quán. Sau khi đã định vị được sự không nhất quán, chúng tôi xác định các tiên đề cần xóa dựa vào khoảng cách ngữ nghĩa.

Các phần tiếp theo của bài báo được tổ chức như sau. Phần 2 trình bày các khái niệm và một số phân tích để định vị sự không nhất quán của ontology trong quá trình tiến hóa. Phần 3 trình bày các khái niệm và cách sử dụng khoảng cách ngữ nghĩa trong quá trình xác định tiên đề cần xóa. Phần 4 trình bày thuật toán kết hợp hai bước định vị và xóa tiên đề để thu được ontology tiến hóa cùng với ví dụ minh hoạ. Phần 5 trình bày kết luận và hướng phát triển.

2 Định vị sự không nhất quán sử dụng chiến lược mở rộng tuyến tính tập tiên đề theo mối quan hệ liên quan về cú pháp

Bài toán định vị sự không nhất quán khi bổ sung tập tiên đề vào một ontology trong quá trình tiến hóa được phát biểu hình thức như sau:

Gọi Σ là tập tiên đề của ontology ban đầu. A_{add} là tập tiên đề cần bổ sung trong quá trình tiến hóa và khiến cho $\Sigma \cup A_{\text{add}}$ không nhất quán. Hãy xác định tập tiên đề $A \subseteq \Sigma$ sao cho $A \cup A_{\text{add}}$ không nhất quán. Tập tiên đề A chứa tiên đề cần xóa để thu được ontology kết quả của quá trình tiến hóa.

Phần này của bài báo trình bày cách áp dụng chiến lược mở rộng tuyến tính tập tiên đề [5] để định vị sự không nhất quán. Chiến lược này được Huang và cs. đề xuất để xây dựng khung lập luận với ontology không nhất quán. Việc mở rộng tập tiên đề được thực hiện dựa trên mối quan hệ *có liên quan về cú pháp*.

Cho một tiên đề ϕ gọi $I(\varphi), C(\varphi), R(\varphi)$ là tập chứa tên cá thể, tên khái niệm, tên vai trò xuất hiện trong tiên đề φ. Người ta định nghĩa mối quan hệ có liên quan trực tiếp về cú pháp giữa hai tiên đề và giữa một tiên đề với một tập tiên đề như sau.

Định nghĩa 1 [5]**.** Hai tiên đề $\varphi,\ \psi$ được gọi là có liên quan trực tiếp về cú pháp nếu và chỉ nếu tồn tại tên cá thể hoặc tên khái niệm hoặc tên vai trò xuất hiện trong cả hai tiên đề. Tức là $I(\varphi) \cap I(\psi) \neq \emptyset \ \vee \ C(\varphi) \cap C(\psi) \neq \emptyset \ \vee R(\varphi) \cap R(\psi) \neq \emptyset$.

Định nghĩa 2 [5]**.** Tiên đề φ được gọi là có liên quan trực tiếp về cú pháp với tập tiên đề Σ nếu và chỉ nếu tồn tại một tiên đề $\psi \in \Sigma$ sao cho φ và ψ là có liên quan trực tiếp với nhau về cú pháp.

Với một tập tiên đề Σ không nhất quán, việc truy vấn $\Sigma \vDash \phi$? hiển nhiên trả về kết quả đúng theo logic cổ điển với mọi tiên đề ϕ. Người ta đã xây dựng phép lập luận không chuẩn để đánh giá truy vấn này dựa trên việc tìm tập tiên đề Σ' – là tập tiên đề con, nhất quán của Σ – mà có thể trả lời được truy vấn $\Sigma' \vDash \phi$? (trả về kết quả đúng hoặc sai). Tập tiên đề $\Sigma' \subset \Sigma$ được gọi là tập tiên đề diễn giải truy vấn $\Sigma \vDash \phi$?. Việc phát triển tuyến tính tập tiên đề diễn giải truy vấn được thực hiện thông qua hàm chọn [5] như sau.

- $s(\Sigma, \{\phi\}, 0) \coloneqq \emptyset$;
- $s(\Sigma, \{\phi\}, 1) \coloneqq \{\psi \in \Sigma \mid \psi \ v\grave{a}\ \phi\ c\acute{o}\ quan\ h\hat{e}\ tr\psi c\ ti\acute{e}p\ v\hat{e}\ m\breve{a}t\ c\acute{u}\ ph\acute{a}p\}$;
- $s(\Sigma, \{\phi\}, k) \coloneqq \{\psi \in \Sigma \mid \psi\ v\grave{a}\ s(\Sigma, \{\phi\}, k-1)\ c\acute{o}\ quan\ h\hat{e}\ tr\psi c\ ti\acute{e}p\ v\hat{e}\ c\acute{u}\ ph\acute{a}p\};\ k \in \boldsymbol{N}^*$

Quá trình phát triển tập tiên đề diễn giải theo hàm chọn $s(\Sigma, \{\phi\}, k)$ là mở rộng tuyến tính theo $k \in \mathbf{N}$, theo nghĩa tập tiên đề này sẽ được mở rộng dần (xuất phát từ tập hợp rỗng). Quá trình mở rộng này sẽ dừng lại khi:

- Hoặc $s(\Sigma, \{\phi\}, k)$ trả lời được truy vấn $s(\Sigma, \{\phi\}, k) \vDash \phi$? ;
- hoặc $(\Sigma, \{\phi\}, k)$ là không nhất quán.

Trong trường hợp $s(\Sigma, \{\phi\}, k)$ thu được là không nhất quán, người ta phải thực hiện quy trình gọi là *xử lý quá xác định (overdetermine)*: tìm cách chọn một hoặc nhiều tiên đề từ tập hợp $\Delta := s(\Sigma, \{\phi\}, k) \setminus s(\Sigma, \{\phi\}, k-1)$ để loại khỏi tập hợp $s(\Sigma, \{\phi\}, k)$ nhằm nhận được tập tiên đề nhất quán Σ' và có thể trả lời được truy vấn, tức là: $\Sigma' \subset s(\Sigma, \{\phi\}, k-1) \cup \Delta \ \wedge \ \Sigma'$ nhất quán $\wedge\, \Sigma' \vDash \phi$? trả về kết quả đúng hoặc sai.

Kết quả truy vấn $\Sigma' \vDash \phi$ được sử dụng làm kết quả của phép truy vấn ϕ với ontology không nhất quán Σ. Bằng cách này, chiến lược mở rộng tuyến tính tập tiên đề diễn giải được xem là một phương án để có thể đánh giá truy vấn với ontology không nhất quán.

Việc mở rộng tuyến tính tập tiên đề diễn giải như trên có thể áp dụng để định vị sự không nhất quán. Thật vậy, chúng ta xây dựng hàm chọn như sau:

- $s(\Sigma, A_{add}, 0) \coloneqq A_{add}$;
- $s(\Sigma, A_{add}, 1) \coloneqq \{\psi \in \Sigma \mid \psi \text{ và } A_{add} \text{ có } quan\ hệ\ trực\ tiếp \text{ về cú } pháp\}$;
- $s(\Sigma, A_{add}, k) \coloneqq \{\psi \in \Sigma \mid \psi \text{ và } s(\Sigma, A_{add}, k-1) \text{ có } quan\ hệ\ trực\ tiếp \text{ về cú } pháp\}$;

Quá trình mở rộng sẽ dừng lại khi $s(\Sigma, A_{add}, k)$ là không nhất quán. Khi đó, tiên đề cần xóa sẽ nằm trong tập tiên đề sau đây:

$$\Delta \coloneqq s(\Sigma, A_{add}, k) \setminus s(\Sigma, A_{add}, k-1)$$

Có thể thấy rằng, quá trình định vị tập tiên đề chứa các tiên đề cần xóa là không phụ thuộc vào các khái niệm không thoả được như những cách tiếp cận khác. Vấn đề còn lại là, làm thế nào để lựa chọn tiên đề cần xóa từ Δ. Phần tiếp theo của bài báo sẽ thảo luận về vấn đề này.

3 Xác định tiên đề cần xóa sử dụng khoảng cách ngữ nghĩa

Khoảng cách ngữ nghĩa giữa hai khái niệm được nghiên cứu trong nhiều công trình về đối sánh ontology. Một trong những phương pháp phổ biến là đánh giá bằng cách dựa trên số cung nối của hai khái niệm này khi đặt chúng trên cây phân cấp khái niệm của một ontology [6]. Ontology này được gọi là ontology tham chiếu để tính khoảng cách ngữ nghĩa. Một cách tương tự, cũng có thể đánh giá khoảng cách ngữ nghĩa giữa các thuộc tính (thuộc tính dữ liệu hoặc thuộc tính đối tượng) khi dặt chúng trên cây phân cấp thuộc tính của một ontology tham chiếu.

Khoảng cách ngữ nghĩa giữa hai tiên đề có thể được lượng hóa nhờ vào khoảng cách ngữ nghĩa của các thực thể khái niệm, thuộc tính xuất hiện trong hai tiên đề này. Có thể sử dụng công thức do [huang2008usingsemanticdistances] đề xuất như sau:

Định nghĩa 3 [huang2008usingsemanticdistances]. Gọi O là một ontology tham chiếu. Khoảng cách ngữ nghĩa giữa hai tiên đề Φ, ψ, ký hiệu là $d_O(\Phi, \psi)$, được định nghĩa như sau:

$$d_{\boldsymbol{O}}(\Phi, \psi) = \frac{\sum_{c_1 \in \boldsymbol{C}(\Phi), c_2 \in \boldsymbol{C}(\psi)} \delta_O(c_1, c_2)}{2.|\boldsymbol{C}(\Phi)|.|\boldsymbol{C}(\psi)|},$$

trong đó $\delta_O(c_1, c_2)$ là khoảng cách ngữ nghĩa giữa hai khái niệm c_1, c_2 khi đặt trên ontology tham chiếu O. Cách tính $\delta_O(c_1, c_2)$ có thể xem tại *[6]*; $|\boldsymbol{X}|$ là lực lượng của tập hợp $\boldsymbol{X}$.

Khi đã rõ ontology O, chúng ta có thể viết d thay vì d_O.

Có thể thấy rằng hàm khoảng cách d_O có những tính chất sau với mọi tiên đề Φ, ψ:

- $0 \leq d_O(\Phi, \psi) \leq 1$;
- $d_O(\Phi, \Phi) = 0$;
- $d_O(\Phi, \psi) = d_O(\psi, \Phi)$;
- Nếu tất cả ký hiệu trong Φ đều có khoảng cách ngữ nghĩa cực đại (=1) với các ký hiệu trong ψ thì khoảng cách ngữ nghĩa giữa Φ và ψ đạt cực đại.

Khoảng cách ngữ nghĩa giữa hai tiên đề chính là cơ sở cho việc chọn lựa tiên đề cần xóa từ một tập tiên đề cho trước. Thật vậy, với A_{add} là tập tiên đề cần bổ sung, Δ là tập chứa các tiên đề cần xóa, có thể sắp xếp thứ tự ưu tiên các tiên đề cần xóa $\alpha \in \Delta$ bằng cách xét khoảng cách ngữ nghĩa từ α đến các tiên đề trong A_{add}. Chúng ta sẽ ưu tiên xóa tiên đề *có khoảng cách ngữ nghĩa nhỏ nhất* đến các tiên đề trong A_{add}. Lý do cho lựa chọn này là, tiên đề có khoảng cách ngữ nghĩa nhỏ nhất đến các tiên đề trong A_{add} thì sẽ có "ngữ nghĩa" gần giống với các tiên đề này. Nói cách khác, việc xóa nó đi sẽ làm mất ít ngữ nghĩa hơn so với việc xóa các tiên đề khác!

4 Thuật toán xây dựng tập tiên đề nhất quán trong quá trình tiến hóa ontology

Tiến hóa ontology được hiểu là quá trình điều chỉnh (bổ sung, chỉnh sửa, xóa) một hoặc nhiều tiên đề trong ontology để đáp ứng sự thay đổi của thế giới thực mà ontology này biểu diễn. Điều kiện cốt lõi nhất của quá trình tiến hóa ontology là phải giữ cho ontology tính nhất quán.

Thao tác chỉnh sửa tiên đề trong quá trình tiến hóa ontology có thể được xem là tổng hợp của hai thao tác tuần tự là xóa và bổ sung tiên đề. Mặt khác, nếu ontology đang là nhất quán thì việc xóa một hoặc nhiều tiên đề khỏi ontology sẽ không làm mất đi tính nhất quán của nó. Như vậy, vấn đề bảo đảm tính nhất quán trong quá trình tiến hóa ontology có thể quy về bài toán "bảo đảm tính nhất quán của ontology khi bổ sung mới một hoặc nhiều tiên đề". Có thể phát biểu một cách hình thức cho bài toán này như sau:

Gọi O là ontology nhất quán với tập tiên đề Σ. A_{add} là tập tiên đề cần bổ sung trong quá trình tiến hóa ontology O. Trong trường hợp $\Sigma \cup A_{\text{add}}$ không nhất quán, hãy tìm ra tập con $A^* \subset A \cup A_{\text{add}}$ sao cho A^* là tập tiên đề nhất quán.

Dưới đây là mô tả mã giả cho toàn bộ quá trình chọn tập tiên đề để thực hiện tiến hóa ontology:

Đầu vào:
– Ontology O với tập tiên đề Σ nhất quán.
– Tập tiên đề cần thêm vào: A_{add} với $\Sigma \cup A_{add}$ là không nhất quán.

Đầu ra:
Tập tiên đề mới A' nhất quán, sao cho $A' = A^* \cup A_{add}$, $A^* \subset \Sigma$ và $|A^*|$ lớn nhất có thể được.

Thủ tục:

$A_{del} := \emptyset$;

repeat // vòng lặp ngoài

$s := A_{add}$;

repeat // vòng lặp trong

$\Delta := \{\phi \mid \phi \in \Sigma \setminus A_{del}; \phi$ *liên quan trực tiếp về cú pháp với* $s\}$;

if ($(s \cup \Delta)$ là nhất quán) **then**

$s := s \cup \Delta$;

until $(s \cup \Delta)$ là không nhất quán;

Tìm $\alpha^* \in \Delta$ sao cho:
$\forall \alpha \in \Delta, \beta \in A_{add}: d_O(\alpha^*, \beta) \leq d_O(\alpha, \beta)$

$A_{del} := A_{del} \setminus \{\alpha^*\}$;

$A^* := \Sigma \setminus A_{del}$;

until $A^* \cup A_{add}$ là nhất quán;

Kết luận tập tiên đề tiến hóa $A' = A^* \cup A_{\text{add}}$.

Giải thích thuật toán

Vòng lặp phía trong thực hiện việc mở rộng tuyến tính tập tiên đề s, xuất phát từ tập hợp tiên đề muốn bổ sung A_{add}. Mỗi bước lặp sẽ bổ sung vào s các tiên đề $\phi \in A^*$có liên quan trực tiếp với s – tập tiên đề được bổ sung này được ký hiệu là Δ. Việc lặp sẽ dừng lại khi $s \cup \Delta$ là không nhất quán. Khi đó, tiên đề gây nên sự không nhất quán là một trong các tiên đề trong Δ. Việc chọn tiên đề trong Δ để xóa được thực hiện theo phân tích ở Phần 3.

Thuật toán có tính dừng do hai nguyên nhân:

– Thứ nhất, vòng lặp phía trong luôn luôn có tính dừng, do quá trình mở rộng tuyến tính về mặt cú pháp tập s từ tập A_{add} luôn luôn có thể dừng với $s \cup \Delta$ là không nhất quán. Trường

hợp duy nhất mà vòng lặp này không dừng với $s \cup \Delta$ không nhất quán chỉ là khi $\Sigma \cup A_{\text{add}}$ nhất quán – trái với giả thiết ban đầu.

– Thứ hai, vòng lặp phía ngoài đảm bảo có tính dừng bởi vì với mỗi tiên đề α^* tìm được ở vòng lặp trong (luôn luôn tìm được ít nhất một tiên đề như vậy từ Δ), A_{del} sẽ mở rộng ra, hay $\Sigma \setminus A_{\text{del}}$ sẽ thu hẹp lại. Sau một số bước hữu hạn, $\Sigma \setminus A_{\text{del}} \cup A_{\text{add}}$ là nhất quán.

Ví dụ minh hoạ

Chúng ta kết thúc phần này bằng một ví dụ minh hoạ cách làm việc theo chiến lược chọn tập tiên đề tiến hóa dựa vào khoảng cách ngữ nghĩa. Chẳng hạn, cho ontology $\mathcal{O}$ với tập tiên đề ban đầu Σ

$$\begin{aligned}\Sigma = \{&Employee \sqsubseteq Person,\\ &Student \sqsubseteq Person,\\ &PhDStudent \sqsubseteq Student,\\ &Staff \sqsubseteq Employee,\\ &Staff \sqcap Student = \bot,\\ &PhDStudent(X)\}.\end{aligned}$$

Tập tiên đề Σ hiện tại là nhất quán. Giả sử, trong quá trình tiến hóa, người ta muốn bổ sung thêm tập tiên đề $A_{\text{add}} = \{ PhDStudent \sqsubseteq Staff \}$. Có thể thấy rằng $\Sigma \cup A_{\text{add}}$ là không nhất quán. Áp dụng thuật toán đã mô tả ở trên, chúng ta tìm cách loại ra khỏi Σ một số tiên đề để việc bổ sung A_{add} tạo nên tập tiên đề nhất quán như dưới đây.

Trước hết, xác định $A_{\text{del}} = \emptyset$.

Ở bước lặp đầu tiên của vòng lặp ngoài, $s = \{ PhDStudent \sqsubseteq Staff \}$

$$\begin{aligned}\Delta = \{&PhDStudent \sqsubseteq Student,\\ &Staff \sqsubseteq Employee,\\ &Staff \sqcap Student = \bot,\\ &PhDStudent(X)\}\end{aligned}$$

Ngay bước lặp đầu tiên của vòng lặp phía trong, $s \cup \Delta$ đã không nhất quán. Chúng ta tìm $\alpha^* \in \Delta$ đạt khoảng cách cực tiểu đến các tiên đề trong A_{add}:

Ta có:

$$\delta(PhDStudent, Student) = \frac{1}{1+2\times 2} = \frac{1}{5}$$

$$\delta(PhDStudent, Staff) = \frac{1}{1+2\times 3} = \frac{1}{7}$$

$$\delta(Staff, Student) = \frac{1+2}{1+2+2\times 1} = \frac{3}{5}$$

$$\delta(PhDStudent, Employee) = \frac{2+1}{2+1+2\times 1} = \frac{3}{5}$$

$$\delta(Staff, Employee) = \frac{1}{1+2\times 2} = \frac{1}{5}$$

Gọi $\alpha = PhDStudent \sqsubseteq Staff$. Như vậy:

$$d(\alpha, PhDStudent \subseteq Student)$$
$$= \frac{1}{4} \times \left(\delta(PhDStudent, Student) + \delta(Staff, PhDStudent) + \delta(Staff, Student)\right)$$
$$= \frac{1}{4} \times \left(\frac{1}{5} + \frac{1}{7} + \frac{3}{5}\right) = \frac{1}{4} \times \frac{33}{35} \approx 0.2357$$

$$d(\alpha, Staff \sqsubseteq Employee)$$
$$= \frac{1}{4}$$
$$\times \left(\delta(PhDStudent, Staff) + \delta(PhDStudent, Employee) + \delta(Staff, Employee)\right)$$
$$= \frac{1}{4} \times \left(\frac{1}{7} + \frac{3}{5} + \frac{1}{5}\right) = \frac{1}{4} \times \frac{33}{35} \approx 0.2357$$

$$d(\alpha, Staff \sqcap Student = \bot)$$
$$= \frac{1}{4} \times \left(\delta(PhDStudent, Staff) + \delta(PhDStudent, Student) + \delta(Staftt, Student)\right)$$
$$= \frac{1}{4} \times \left(\frac{1}{7} + \frac{3}{5} + \frac{3}{5}\right) = \frac{1}{4} \times \frac{47}{35} \approx 0.3357$$

$$d\left(\alpha, PhDStudent(X)\right)$$
$$= \tfrac{1}{2} \times \left(\delta(PhDStudent, PhDStudent) + \delta(Staff, PhDStudnet)\right)$$
$$= \tfrac{1}{2} \times \left(0 + \tfrac{1}{7}\right) = \tfrac{1}{14} \approx 0.0714$$

Vậy $A_{del} = \{PhDStudent \subseteq Student\}$ hoặc $A_{del} = \{Staff \sqsubseteq Employee\}$. Tương ứng, chúng ta có hai tập tiên đề $\Sigma \setminus A_{del} \cup A_{add}$ như sau:

- $A'_1 = \{\, Employee \sqsubseteq Person,$
  $Student \sqsubseteq Person,$
  $PhDStudent \sqsubseteq Staff,$
  $Staff \sqsubseteq Employee,$
  $Staff \sqcap Student = \bot,$
  $PhDStudent(X)\,\}$
- $A'_2 = \{\, Employee \sqsubseteq Person,$
  $Student \sqsubseteq Person,$
  $PhDStudent \sqsubseteq Staff,$

$PhDStudent \subseteq Student$,
$Staff \sqcap Student = \bot$,
$PhDStudent(X)$ }

Cả A'_1, A'_2 đều đáp ứng điều kiện (là tập tiên đề nhất quán). Thuật toán dừng với tập tiên đề kết quả là A'_1 hoặc A'_2.

5 Kết luận và hướng mở rộng

Bài báo trình bày phương pháp sử dụng *chiến lược mở rộng tuyến tính dựa trên mối quan hệ liên quan về mặt cú pháp* để định vị sự không nhất quán; sau đó dựa trên khoảng cách ngữ nghĩa để lựa chọn tiên đề cần loại bỏ trong quá trình tiến hóa ontology. Việc áp dụng các kỹ thuật này đem lại các lợi ích có thể nhìn thấy được. Thứ nhất, quá trình định vị sự không nhất quán diễn ra rất nhanh. Thứ hai, khoảng cách ngữ nghĩa là một tiêu chí mang tính định lượng, giúp tự động chọn được tiên đề cần xóa, hoặc gợi ý đề người dùng lựa chọn tiên đề cần xóa.

Trong một số trường hợp, việc dựa trên khoảng cách ngữ nghĩa cũng không định ra được duy nhất tiên đề cần loại bỏ. Hướng phát triển tiếp theo của bài báo là xét thêm một số tiêu chí nữa, chẳng hạn quy tắc được đề xuất trong [1] để thu gọn lại tập hợp tiên đề gợi ý loại bỏ. Một hướng phát triển nữa là tính khoảng cách ngữ nghĩa giữa hai tiên đề dựa trên biểu thức khái niệm, biểu thức vai trò thay vì dựa trên các tên thực thể như hiện tại. Việc này sẽ giúp loại bỏ sự phụ thuộc về mặt cú pháp trong quá trình tính toán độ liên quan giữa hai tiên đề.

Tài liệu tham khảo

1. Leila Bayoudhi, Najla Sassi, and Wassim Jaziri. Overview and Reflexion on OWL 2 DL Ontology Consistency Rules. In Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing, ICC '17, pages 133:1—-133:8, New York, NY, USA, 2017 ACM.
2. R. Djedidi and M.-A. Aufaure. Ontology Evolution: State of the Art and Future Directions. Ontol. Theory, Manag. Des. Adv. Tools Model., jan 2010.
3. B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. Web Semant. Sci. Serv. Agents World Wide Web, 6(4):309322, 2008.
4. P. Haase, F. Van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In Semant. Web - SWC 2005, volume 3729 LNCS, pages 353367. Springer, 2005.
5. Z. Huang, F. van Harmelen, and A. Ten Teije. Reasoning with inconsistent ontologies. IJCAI Int. Jt. Conf. Artif. Intell., 3:454459, 2005.
6. G. Jike and Q. Yuhui. Concept similarity matching based on semantic distance. Proc. 4th Int. Conf. Semant. Knowl. Grid, SKG 2008, pages 380383, 2008.

7. P. Plessers and O. De Troyer. Resolving inconsistencies in evolving ontologies. In Semant. Web Res. Appl., volume 4011 LNCS, pages 200214. Springer, 2006.L. Stojanovic. Methods and tools for ontology evolution. 2004.