

## MỘT PHƯƠNG PHÁP HỖ TRỢ XỬ LÝ TRI THỨC KHÔNG NHẤT QUÁN TRONG TIẾN HOÁ ONTOLOGY

Nguyễn Văn Trung<sup>1\*</sup>, Nguyễn Thị Bích Lộc<sup>1\*</sup>, Trần Đình Sơn<sup>2</sup>

<sup>1</sup>Khoa Công nghệ Thông tin, Trường Đại học Khoa học, Đại học Huế

<sup>2</sup>Cao đẳng Công nghệ Thông tin, Đại học Đà Nẵng

\*Email: nvtrung@gmail.com

### 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 hoá ontology - theo thông qua việc bổ sung, chỉnh sửa hoặc xoá các tiên đề để phản ánh tri thức mới. Thách thức lớn nhất trong quá trình tiến hoá ontology đó là, ngoài việc phản ánh tri thức mới, ontology còn phải bảo đảm tính nhất quán. Bài báo này đề xuất phương pháp dựa trên khoảng cách ngữ nghĩa để xây dựng tập tiên đề nhất quán trong quá trình tiến hoá ontology.

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

### 1. MỞ ĐẦU

Năm 2001, Tim Berners Lee cùng cộng sự đã đưa ra phác thảo cho một “dạng thức mới về nội dung web mà dạng thức này có ý nghĩa đối với máy tính” [2]. Thế hệ web sử dụng dạng thức nội dung này -- gọi là Web ngữ nghĩa (Semantic Web) -- cho phép máy tính có thể “hiểu” tri thức được lưu trữ, theo đó có thể chia sẻ và tái sử dụng các cơ sở tri thức trong các hệ thống thông tin thuộc nhiều lĩnh vực khác nhau.

Web ngữ nghĩa là một lĩnh vực nghiên cứu đang phát triển nhanh và nhận được sự quan tâm của cộng đồng nghiên cứu trong thập niên vừa qua. Công nghệ Web ngữ nghĩa đang được áp dụng vào nhiều lĩnh vực khác nhau trong thực tế như tin-sinh học, tin học trong y tế, quản trị tri thức, công nghệ phần mềm, ... Thành phần quan trọng trong các ứng dụng web ngữ nghĩa là ontology.

Trong một ontology người ta định nghĩa các thực thể (bao gồm khái niệm, thuộc tính, cá thể) và mối quan hệ giữa các thực thể này theo ngữ nghĩa được quy định tường minh bởi một ngôn ngữ logic xác định. Ngôn ngữ ontology được sử dụng phổ biến và chuẩn hoá bởi tổ chức tiêu chuẩn quốc tế W3C (World Wide Web Consortium) là OWL2, được xây dựng dựa trên logic mô tả  $\mathcal{SROIQ}(\mathbf{D})$ . Sử dụng các bộ lập luận, người ta có thể

truy vấn các thông tin ngay cả khi không được phát biểu tường minh trong ontology. Đây là lợi ích logic của ontology so với các giải pháp lưu trữ thông tin thế hệ trước.

Các ontology biểu diễn tri thức cho một lĩnh vực cụ thể tương ứng với thế giới thực thường có nhu cầu thay đổi theo thời gian để phản ánh đúng với thế giới thực. Quá trình thay đổi này thường được biết đến trong lĩnh vực kỹ nghệ ontology với tên gọi *tiến hoá ontology*. Thách thức quan trọng của quá trình tiến hoá ontology đó là phải bảo đảm tính nhất quán. Điều này được giải thích bởi, nếu ontology không nhất quán thì mọi tiên đề đều là hệ quả logic của nó. Nói cách khác ontology không nhất quán làm mất đi ích lợi biểu diễn và truy vấn thông tin.

Người ta đã đưa ra nhiều giải pháp để đảm bảo tính nhất quán của ontology trong quá trình tiến hoá. Các giải pháp này tựu trung lại đều được căn cứ trên các quy tắc [1] nhằm đảm bảo tính hợp lệ về cú pháp của ontology theo ràng buộc ngôn ngữ đã được xác định (ví dụ ngôn ngữ OWL 2 DL, OWL 2 Full, ...) và/hoặc đảm bảo quy tắc đảm bảo không mâu thuẫn về mặt logic của các tiên đề.

Các quy tắc về cú pháp hoàn toàn có thể được kiểm tra và gợi ý để người dùng tuân thủ trong các hệ thống soạn thảo ontology. Trong khi đó, các quy tắc về logic rất khó được đảm bảo thực hiện: Các quy tắc này hướng đến giải pháp dò tìm tập tiên đề là nguyên nhân gây lỗi trong ontology (làm cho ontology là không nhất quán) và sau đó gợi ý người dùng chỉnh sửa và/hoặc xoá một hoặc nhiều tiên đề này khỏi ontology. Tuy vậy, người dùng rất khó sử dụng các gợi ý này trong trường hợp họ phải xoá/chỉnh sửa nhiều tiên đề, do họ không biết nên áp dụng bao nhiêu lần, theo thứ tự nào. Đây chính là động lực nghiên cứu của bài báo.

Bài báo này đề xuất phương pháp sử dụng khoảng cách ngữ nghĩa để xác định mức độ ưu tiên khi chọn lựa tập tiên đề trong quá trình tiến hoá ontology. Các phần tiếp theo của bài báo được trình bày như sau: **Phần 2** giới thiệu cách xác định khoảng cách ngữ nghĩa giữa hai tiên đề của ontology. **Phần 3** phân tích và trình bày thuật toán chọn tập tiên đề tối ưu trong quá trình tiến hoá ontology dựa theo tiêu chuẩn về khoảng cách ngữ nghĩa giữa hai tiên đề. **Phần 4** thảo luận về đóng góp của bài báo và các hướng mở rộng kết quả.

## 2. KHOẢNG CÁCH NGỮ NGHĨA GIỮA HAI TIÊN ĐỀ TRONG ONTOLOGY

Một ontology  $O$  được thành lập từ hai thành phần  $O = \langle S, A \rangle$ , trong đó,  $S$  là bộ từ vựng là tập hợp tên các thực thể (gồm tên khái niệm  $N_C$ , tên thuộc tính  $N_P$ , tên kiểu dữ liệu  $N_{DT}$ , tên cá thể  $N_I$ ),  $A$  là tập tiên đề của ontology. Ontology  $O$  được gọi là nhất quán khi tập tiên đề của nó,  $A$ , là nhất quán, tức không chứa mâu thuẫn [5].

Các tiên đề trong tập hợp  $A$  của ontology  $O$  định ra các ràng buộc cũng như mối quan hệ giữa các thực thể trong  $S$ . Mối quan hệ thường được xét đến nhiều nhất giữa các thực thể trong một ontology chính là mối quan hệ phân cấp, theo đó, các thực thể

trong ontology được xếp trên các cây phân cấp: cây phân cấp khái niệm và cây phân cấp thuộc tính. Vị trí tương đối của các thực thể trên cây phân cấp cho phép xác định khoảng cách ngữ nghĩa giữa chúng. Do tiên đề được thành lập từ các thực thể nên khoảng cách ngữ nghĩa giữa hai tiên đề có thể được xây dựng từ khoảng cách ngữ nghĩa giữa các thực thể cấu tạo nên chúng.

Trên thực tế, mối quan hệ phân cấp giữa các khái niệm và mối quan hệ phân cấp giữa các thuộc tính (thuộc tính đối tượng, thuộc tính dữ liệu) có bản chất tương tự. Do đó, không mất tính tổng quát, bài báo chỉ trình bày cách tính khoảng cách ngữ nghĩa giữa hai khái niệm và sau đó mở rộng cho việc tính khoảng cách giữa hai biểu thức khái niệm. Phần này của bài báo sẽ trình bày lần lượt khoảng cách ngữ nghĩa giữa hai khái niệm trong ontology, khoảng cách ngữ nghĩa giữa hai biểu thức khái niệm và cuối cùng là khoảng cách ngữ nghĩa giữa hai tiên đề.

### 2.1. Khoảng cách ngữ nghĩa giữa hai khái niệm trong ontology

Trong ontology  $O$ , với hai khái niệm  $c_1, c_2 \in N_C$ , chúng ta ký hiệu:

- $DP_O(c_1)$  là tập các khái niệm cha trực tiếp của  $c_1$  trên  $O$ .
- $DC_O(c_1)$  là tập các khái niệm con trực tiếp của  $c_1$  trên  $O$ .
- $LCP_O(c_1, c_2)$  là tập các khái niệm cha chung nhỏ nhất của  $c_1$  và  $c_2$  trên  $O$ .

Chúng ta định nghĩa số cạnh nối giữa hai khái niệm trên cây phân cấp khái niệm của ontology  $O$  như sau:

**Định nghĩa 1.** (Số cạnh nối giữa hai khái niệm trên cây phân cấp của ontology) Gọi  $O$  là một ontology. Ta nói  $k \in \mathbf{N}$  là số cạnh nối giữa hai khái niệm  $c_1, c_2 \in N_C$ , ký hiệu là  $\mu_O(c_1, c_2) = k$  nếu tồn tại  $d_1, d_2, \dots, d_k$  là dãy ngắn nhất các khái niệm có tên trong  $O$  sao cho:

$$\begin{cases} c_1 \equiv d_1 \\ c_2 \equiv d_2 \\ d_i \in DC_O(d_{i+1}) \forall i = 1, 2, \dots, k-1 \end{cases}$$

Ta quy ước:

- $\mu_O(c, c) = 0$  với  $\forall c \in N_C$ ;
- $\mu_O(c_1, c_2) = +\infty$  nếu  $c_1 \equiv \perp$  hoặc  $c_1$  không phải là khái niệm con của  $c_2$ .
- Trong trường hợp ontology  $O$  đã được xác định rõ, chúng ta có thể bỏ qua chỉ số  $O$  để viết  $\mu$  thay vì  $\mu_O$ .

■

Khoảng cách ngữ nghĩa giữa hai khái niệm trên cây phân cấp được định nghĩa dựa trên thông số số cạnh nối đã được định nghĩa ở trên như sau.

**Định nghĩa 2.** (Khoảng cách ngữ nghĩa giữa hai khái niệm trên cây phân cấp của ontology) Gọi  $O$  là một ontology. Khoảng cách ngữ nghĩa giữa hai khái niệm  $c_1, c_2 \in N_C$  ký hiệu là  $\delta_O(c_1, c_2)$  được xác định theo công thức:

$$\delta_{\mathcal{O}}(c_1, c_2) = \min\left\{\frac{\mu_{\mathcal{O}}(c_1, c_0) + \mu_{\mathcal{O}}(c_2, c_0)}{\mu_{\mathcal{O}}(c_1, c_0) + \mu_{\mathcal{O}}(c_2, c_0) + 2 \cdot \mu_{\mathcal{O}}(c_0, \top)} \mid c_0 \in LCP_{\mathcal{O}}(c_1, c_2)\right\}$$

Trong trường hợp ontology  $\mathcal{O}$  đã được xác định rõ, chúng ta có thể bỏ qua chỉ số  $\mathcal{O}$  để viết  $\delta$  thay vì  $\delta_{\mathcal{O}}$ . ■

Chúng ta có một số nhận xét rút ra trực tiếp từ **Định nghĩa 1** và **Định nghĩa 2** như mệnh đề dưới đây:

**Mệnh đề 1.** Gọi  $\mathcal{O}$  là một ontology. Với mọi khái niệm  $c_1, c_2 \in N_C$ , ta có:

- $\delta(c_1, c_2) = \delta(c_2, c_1)$
  - $0 \leq \delta(c_1, c_2) \leq 1$
  - $\delta(c_1, c_2) = 0$  khi  $c_1 \equiv c_2$
  - $\delta(c_1, c_2) = 1$  khi  $LCP_{\mathcal{O}}(c_1, c_2) = \{\top\}$  hoặc  $c_1 \equiv \perp$  hoặc  $c_2 \equiv \perp$
- 

## 2.2. Khoảng cách ngữ nghĩa giữa hai biểu thức khái niệm bất kỳ

Với  $\mathcal{O}$  là một ontology, chúng ta mở rộng khái niệm khoảng cách ngữ nghĩa giữa hai khái niệm trong ontology thành “khoảng cách ngữ nghĩa giữa hai biểu thức khái niệm theo ontology  $\mathcal{O}$ ” như sau: Khoảng cách ngữ nghĩa giữa hai biểu thức khái niệm  $CE_1, CE_2$  theo ontology  $\mathcal{O}$  được hiểu là khoảng cách ngữ nghĩa của hai biểu thức khái niệm này khi đặt trên cây phân cấp khái niệm của ontology  $\mathcal{O}$ .

Theo công thức ở **Định nghĩa 2**, khoảng cách ngữ nghĩa giữa hai khái niệm  $CE_1, CE_2$  theo ontology  $\mathcal{O}$  là:

$$\delta_{\mathcal{O}}(CE_1, CE_2) = \min\left\{\frac{\mu_{\mathcal{O}}(CE_1, CE_0) + \mu_{\mathcal{O}}(CE_2, CE_0)}{\mu_{\mathcal{O}}(CE_1, CE_0) + \mu_{\mathcal{O}}(CE_2, CE_0) + 2 \cdot \mu_{\mathcal{O}}(CE_0, \top)} \mid CE_0 \in LCP_{\mathcal{O}}(CE_1, CE_2)\right\}$$

Chúng ta có thể tính số cung nối giữa hai biểu thức khái niệm bất kỳ  $CE_1$  và  $CE_2$  dựa theo số cung nối giữa hai khái niệm có tên của ontology  $\mathcal{O}$  theo 3 trường hợp như sau:

- Trường hợp 1)  $\exists c_1 \in C_{\mathcal{O}}: c_1 \equiv CE_1$ . Khi đó:

$$\mu(CE_1, CE_2) = \min\{\mu_{\mathcal{O}}(c_1, c) \mid c \in DC_{\mathcal{O}}(CE_2)\}$$

- Trường hợp 2)  $\exists c_2 \in C_{\mathcal{O}}: c_2 \equiv CE_2$ . Khi đó:

$$\mu_{\mathcal{O}}(CE_1, CE_2) = \min\{\mu_{\mathcal{O}}(c, c_2) \mid c \in DP_{\mathcal{O}}(CE_1)\}$$

- Trường hợp 3) Không có khái niệm có tên trên  $\mathcal{O}$  tương đương với  $CE_1$  hoặc  $CE_2$ . Khi đó:

$$\mu_{\mathcal{O}}(CE_1, CE_2) = \min\{2 + \mu_{\mathcal{O}}(c, c') \mid c \in DP_{\mathcal{O}}(CE_1), c' \in DC_{\mathcal{O}}(CE_2)\}$$

Như vậy, bằng cách tính sẵn các thông số  $\mu$  giữa các cặp khái niệm của ontology  $O$  (có thể dùng thuật toán tìm đường đi ngắn nhất giữa các cặp đỉnh như Floyd [3] chẳng hạn), chúng ta có thể tính nhanh khoảng cách ngữ nghĩa  $\delta$  cho cặp biểu thức khái niệm bất kỳ.

### 2.3. Khoảng cách ngữ nghĩa giữa hai tiên đề

Theo tài liệu đặc tả cấu trúc OWL 2, chúng ta luôn xác định được tập biểu thức khái niệm (cũng như tập biểu thức thuộc tính đối tượng, tập biểu thức thuộc tính dữ liệu) của một tiên đề bất kỳ của ontology OWL 2. Điều tương tự cũng có thể thực hiện được với các ngôn ngữ ontology khác. Khoảng cách ngữ nghĩa giữa hai tiên đề theo ontology tham chiếu có thể được tính bằng trung bình cộng của tất cả các khoảng cách ngữ nghĩa giữa hai biểu thức khái niệm có trong hai tiên đề đó. Cụ thể như sau:

**Định nghĩa 3** (Khoảng cách ngữ nghĩa giữa hai tiên đề). Gọi  $O = \langle S, A \rangle$  là một ontology. Khoảng cách ngữ nghĩa giữa hai tiên đề  $\Phi, \psi \in A$ , ký hiệu là  $d_O(\Phi, \psi)$ , được định nghĩa như sau:

$$d_O(\Phi, \psi) = \frac{\sum_{CE_1 \in CE(\Phi), CE_2 \in CE(\psi)} \delta_O(CE_1, CE_2)}{2 \cdot card(CE(\Phi)) \cdot card(CE(\psi))},$$

trong đó:

- $CE(\Phi)$  là tập các biểu thức khái niệm xuất hiện trong tiên đề  $\Phi$ .
- $card(X)$  là lực lượng của tập hợp  $X$ .

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

Như vậy, chúng ta luôn luôn xác định được khoảng cách ngữ nghĩa giữa hai tiên đề bằng cách dựa vào số cung nối giữa hai khái niệm bất kỳ của ontology.

## 3. XÂY DỰNG TẬP TIÊN ĐỀ NHẤT QUÁN TRONG QUÁ TRÌNH TIẾN HOÁ ONTOLOGY

Tiến hoá ontology được hiểu là quá trình điều chỉnh (bổ sung, chỉnh sửa, xoá) 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 hoá ontology là phải giữ cho ontology tính nhất quán.

Thao tác chỉnh sửa tiên đề trong quá trình tiến hoá ontology có thể được xem là tổng hợp của hai thao tác tuần tự là xoá và bổ sung tiên đề. Mặt khác, nếu ontology đang là nhất quán thì việc xoá 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 hoá 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  $O = \langle S, A \rangle$  là ontology nhất quán ( $A$  là tập tiên đề nhất quán).  $A_{add}$  là tập tiên đề cần bổ sung trong quá trình tiến hoá ontology  $O$ . Trong trường hợp  $A \cup A_{add}$  không nhất quán, hãy tìm ra tập con  $A^* \subset A \cup A_{add}$  sao cho  $A^*$  là tập tiên đề nhất quán.

### 3.1. Phân tích yêu cầu tiến hoá

Quá trình tiến hoá ontology thường được thực hiện để đảm bảo ontology phản ánh đúng thế giới thực. Theo đó, các tiên đề cần bổ sung vào được xem như có giá trị hơn những tiên đề đã có trong ontology hiện tại. Nói cách khác, chúng ta ưu tiên giữ lại những tiên đề này trong quá trình tiến hoá. Mặt khác, chúng ta cũng giả thiết rằng, tập tiên đề cần bổ sung  $A_{add}$  là không chứa mâu thuẫn. Chiến lược chọn tập tiên đề tiến hoá là xây dựng tập tiên đề  $A'$  bằng cách xoá một số ít nhất các tiên đề trong  $A$  sao cho  $A' \cup A_{add}$  là nhất quán.

Một ontology là không nhất quán khi trong ontology có tạo ra cá thể của một khái niệm không thoả được (*unsatisfiable concept*) (hoặc thuộc tính đối tượng không thoả được, hoặc thuộc tính dữ liệu không thoả được). Do sự tương tự của khái niệm và thuộc tính đối tượng, thuộc tính dữ liệu, bài báo này chỉ xét đến việc xử lý tri thức không nhất quán với khái niệm không thoả được.

Trên thực tế, với một khái niệm không thoả được trong ontology, người ta có thể tính ra được tập tiên đề nhỏ nhất sao cho khái niệm đó vẫn không thoả được trong tập tiên đề này, gọi là MUPS (Minimal unsatisfiability-preserving sub-ontology) [6]. Bằng cách xoá đi một tiên đề khỏi tập tiên đề MUPS, chúng ta sẽ thu được tập tiên đề nhất quán. Đây chính là ý tưởng của việc xây dựng tập tiên đề nhất quán trong quá trình tiến hoá [4]. Tuy vậy, chọn tiên đề nào để xoá cũng là một vấn đề đang gây khó khăn cho người dùng. Bài báo này đề xuất lựa chọn tiên đề có khoảng cách ngữ nghĩa nhỏ nhất đến tiên đề trong  $A_{add}$  để xoá. Lý do cho lựa chọn này là: Tiên đề có khoảng cách ngữ nghĩa nhỏ nhất đến tập tiên đề muốn thêm vào thì sẽ có “ngữ nghĩa” gần giống với các tiên đề chuẩn bị thêm vào. Do đó, việc xoá nó đi sẽ làm mất ít ngữ nghĩa hơn.

### 3.2. Chiến lược chọn tập tiên đề cho ontology tiến hoá

Gọi  $O = \langle S, A \rangle$  là ontology nhất quán ban đầu.  $A_{add}$  là tập tiên đề cần bổ sung trong quá trình tiến hoá ontology  $O$ . Giả thiết  $A \cup A_{add}$  là không nhất quán, trong đó  $c_{us}$  là khái niệm không thoả được trên tập tiên đề  $A \cup A_{add}$ . Theo phân tích ở phần trước, chiến lược xây dựng tập tiên đề nhất quán trong quá trình tiến hoá ontology  $O$  sẽ được thực hiện theo hai giai đoạn:

Giai đoạn 1: Tìm một tập tiên đề  $A_{min} := MUPS(c_{us}, A \cup A_{add})$  sao cho khái niệm  $c_{us}$  là không thoả được trong  $A_{min}$ , đồng thời  $A_{min}$  là tối thiểu theo nghĩa:

$$A_{min} \subseteq A \cup A_{add} \text{ và } \forall \alpha \in A_{min}: c_{us} \text{ là thoả được trong } A_{min} \setminus \{\alpha\}.$$

Giai đoạn 2: Tìm tiên đề  $\alpha^* \in A_{min}$  sao cho khoảng cách từ nó đến các tiên đề cần thêm là cực tiểu:

$$\forall \alpha \in A_{min}, \beta \in A_{add}: d_o(\alpha^*, \beta) \leq d_o(\alpha, \beta)$$

Hiển nhiên rằng, có thể có nhiều khái niệm không thoả trong tập tiên đề  $A \cup A_{add}$ . Trong tình huống này, với mỗi khái niệm không thoả, chúng ta áp dụng lần lượt hai giai đoạn xử lý ở trên để thu được tập tiên đề tối ưu.

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

**Đầu vào:**

- Ontology  $O = \langle S, A \rangle$ ;
- Tập tiên đề cần thêm vào:  $A_{add}$  với  $A \cup A_{add}$  là không nhất quán.

**Đầu ra:**

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

**Thủ tục:**

Xác định:  $A^* := A$ ;

Tính  $CUS := \{c_{us} \in S \mid c_{us} \text{ không thoả được trong } A \cup A_{add}\}$ ;

for  $c_{us} \in CUS$

Tính  $A_{min} := MUPS(c_{us}, A^* \cup A_{add})$ ;

Tìm  $\alpha^* \in A^*$  sao cho:

$$\forall \alpha \in A^*, \beta \in A_{add}: d_o(\alpha^*, \beta) \leq d_o(\alpha, \beta)$$

Thực hiện gán  $A^* := A^* \setminus \{\alpha^*\}$ .

Nếu  $A^* \cup A_{add}$  không còn chứa khái niệm không thoả được thì chấm dứt lặp;

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

### 3.3. 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 hoá dựa vào khoảng cách ngữ nghĩa. Chẳng hạn, cho ontology  $O$  với tập tiên đề  $A = \{Employee \sqsubseteq Person, Student \sqsubseteq Person, PhDStudent \sqsubseteq Student, Staff \sqsubseteq Employee, Staff \sqcap Student = \perp, PhDStudent(X)\}$ . Tập tiên đề  $A$  hiện tại không chứa mâu thuẫn. Giả sử, trong quá trình tiến hoá, người ta muốn bổ sung thêm

Hướng dẫn viết bài đăng trên Tạp chí Khoa học và Công nghệ Trường Đại học Khoa học, ĐH Huế

tiên đề  $\alpha = PhDStudent \sqsubseteq Staff$ . Hiển nhiên rằng  $A \cup \{\alpha\}$  có chứa mâu thuẫn. Áp dụng chiến lược đã nêu ở Phần 3.2, chúng ta tìm cách loại ra khỏi  $A$  một số tiên đề để việc bổ sung  $\alpha$  tạo nên tập tiên đề nhất quán như dưới đây.

Trước hết, xác định  $A^* = A$ .

Khái niệm không thoả xuất hiện trong tập tiên đề  $A \cup \{\alpha\}$  là  $PhDStudent$ . Tập tiên đề MUPS cho khái niệm này chính là  $\{PhDStudent \sqsubseteq Student, \alpha, Staff \sqsubseteq Employee, Staff \sqcap Student = \perp\}$ .

Cần phải tìm ra một trong ba tiên đề khác  $\alpha$  trong MUPS để loại trừ. Tiêu chí để lựa chọn là khoảng cách ngữ nghĩa giữa các tiên đề này với  $\alpha$ .

Ta có:

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

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

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

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

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

Như vậy:

$$d(\alpha, PhDStudent \sqsubseteq Student)$$

$$= \frac{1}{4} \times (\delta(PhDStudent, Student) + \delta(Staff, PhDStudent) + \delta(Staff, Student))$$

$$= \frac{1}{4} \times \left( \frac{1}{5} + \frac{1}{7} + \frac{3}{5} \right) = \frac{1}{4} \times \frac{33}{35}$$

$$d(\alpha, Staff \sqsubseteq Employee)$$

$$= \frac{1}{4}$$

$$\times (\delta(PhDStudent, Staff) + \delta(PhDStudent, Employee) + \delta(Staff, Employee))$$

$$= \frac{1}{4} \times \left( \frac{1}{7} + \frac{3}{5} + \frac{1}{5} \right) = \frac{1}{4} \times \frac{33}{35}$$



$$\begin{aligned}
d(\alpha, Staff \sqcap Student = \perp) \\
&= \frac{1}{4} \\
&\times (\delta(PhDStudent, Staff) + \delta(PhDStudent, Student) \\
&+ \delta(Staff, Student)) = \frac{1}{4} \times \left( \frac{1}{7} + \frac{3}{5} + \frac{3}{5} \right) = \frac{1}{4} \times \frac{47}{35}
\end{aligned}$$

Vậy cần phải loại bỏ tiên đề  $PhDStudent \sqsubseteq Student$  hoặc tiên đề  $Staff \sqsubseteq Employee$  ra khỏi  $A$  trước khi bổ sung tiên đề  $\alpha$  trong quá trình tiến hoá.

#### 4. KẾT LUẬN

Bài báo trình bày phương pháp sử dụng khoảng cách ngữ nghĩa để xây dựng tập tiên đề trong quá trình tiến hoá. Đây là đóng góp có ý nghĩa bởi nó sẽ giúp người dùng có được tiêu chí định lượng cụ thể khi xét duyệt các tiên đề.

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 bởi [1] để thu gọn lại tập hợp tiên đề gọi ý loại bỏ.

#### 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] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [3] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.
- [4] Peter Haase, Frank Van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. A framework for handling inconsistency in changing ontologies. In *The Semantic Web - SWC 2005*, volume 3729 LNCS, pages 353–367. Springer, 2005.
- [5] Markus Krötzsch, Frantisek Simancik, and Ian Horrocks. A Description Logic Primer. arXiv preprint arXiv:1201.4089, pages 1–17, jan 2012.
- [6] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI International Joint Conference on Artificial Intelligence*, volume 3, pages 355–360. Morgan Kaufmann, 2003.

## **A NOVEL METHOD TO RESOLVE INCONSISTENCY IN ONTOLOGY EVOLUTION**

**Nguyen Van Trung<sup>1\*</sup>, Nguyen Thi Bich Loc<sup>1\*</sup>, Tran Dinh Son<sup>2</sup>**

<sup>1</sup>Department of Information Technology, University of Sciences, Hue University

<sup>2</sup>College of Information Technology, The University of Danang

\*Email: nvtrung@hueuni.edu.vn

### **ABSTRACT**

Ontology captures domain knowledge and play an important role in Semantic Web applications. When domain knowledge change, ontology should adapt to change by adding, editing or deleting its axioms. The big challenge of this process – which called as ontology evolution – is how to guarantee the consistency state of the ontology. This paper propose a novel method based on semantic distance to formulate consistent set of axioms in ontology evolution process.

**Từ khóa:** ontology evolution, inconsistency handling, semantic distance