

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ



FIRST ORDER LOGIC AND KNOWLEDGE BASE

BÁO CÁO GIỮA KỲ

Ngành: Công nghệ kỹ thuật Cơ điện tử

Môn: Trí tuệ nhân tạo

HÀ NỘI – 2024

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Sinh viên: Bùi Tuấn Huy - MSV: 21021311

Nguyễn Công Quốc Huy - 21021313

Ngô Huy Hoàng - 21021307

Nguyễn Đức Huy - 21020587

Nguyễn Văn Huy – 21021315

FIRST ORDER LOGIC AND KNOWLEDGE BASE

BÁO CÁO CUỐI KỲ

Ngành: Công nghệ kỹ thuật Cơ điện tử

Môn: Trí tuệ nhân tạo

Giảng viên: Trần Hồng Việt

HÀ NỘI – 2024

Mục lục

PHẦN I –First-order logic	4
1. Giới thiệu	4
2. Cú pháp và ngữ nghĩa	4
3. Mở rộng và biến thể ký hiệu	5
4. Sử dụng logic bậc nhất : Biểu diễn quan hệ gia đình và tập hợp toán học	6
5. Tác nhân logic cho thế giới WUMPUS	6
6. Tác nhân phản xạ đơn giản	6
7. Biểu diễn sự thay đổi trong thế giới	7
8. Suy luận các thuộc tính ẩn của thế giới	7
9. Ưu tiên các hành động	7
10. Hướng đến tác nhân dựa trên mục tiêu	7
PHẦN II – Knowledge base.....	8
1. Kiến tạo cơ sở tri thức	8
2. Đặc điểm của cơ sở tri thức tốt và xấu	8
3. Kỹ thuật tri thức	9
PHẦN III - Ứng dụng First-order logic và Knowledge base	10
1. Giới thiệu ứng dụng	10
2. Các thành phần chính trong ứng dụng	10
3. Cách thức hoạt động của ứng dụng	11
4. Dữ liệu và chương trình của ứng dụng	11
5. Kết quả	12
PHẦN IV – Tham khảo	15

PHẦN I: First-order logic

1. Giới thiệu

- First-order logic - FOL (Logic bậc nhất), còn được gọi là logic vị từ hoặc tính toán vị từ bậc nhất, là một hệ thống hình thức được sử dụng trong toán học, triết học, ngôn ngữ học và khoa học máy tính. Nó cung cấp một khung để định nghĩa các mệnh đề logic về các đối tượng và mối quan hệ của chúng.

2. Cú pháp và ngữ nghĩa

- *Ký hiệu hằng số*: Một diễn giải phải xác định đối tượng nào trong thế giới được chỉ bởi mỗi ký hiệu hằng số. Ví dụ: A, B, C, John.
- *Ký hiệu vị từ*: Một diễn giải xác định rằng một ký hiệu vị từ chỉ một quan hệ cụ thể trong mô hình. Ví dụ: Round, Brother.
- *Ký hiệu hàm*: Ánh xạ đối tượng đến đối tượng. Chúng cũng có một arity (ví dụ: $f(x)$, $g(x, y)$).
- *Thuật ngữ (TERM)*: Là một biểu thức logic chỉ đối tượng.
 - Hằng số: Ví dụ: a
 - Biến số: Ví dụ: x
 - Ứng dụng hàm: Ví dụ: $f(x)$
- *Mệnh đề nguyên tử (ATOMIC SENTENCES)*: Biểu diễn các khẳng định cơ bản về các thuật ngữ. Một công thức nguyên tử bao gồm một ký hiệu vị từ áp dụng cho một chuỗi các thuật ngữ.
- *Mệnh đề phức hợp (COMPLEX SENTENCES)*: Được xây dựng từ các công thức nguyên tử sử dụng các kết nối logic và lượng từ từ.
 - Ví dụ: "John là anh em của Richard và Richard là anh em của John" được biểu diễn như: $\text{Brother}(\text{Richard}, \text{John}) \wedge \text{Brother}(\text{John}, \text{Richard})$.
- *Lượng từ phổ quát (Universal Quantification)*:
 - Ký hiệu: \forall
 - Ý nghĩa: "Cho mọi"
 - Ví dụ: "Tất cả mèo đều là động vật có vú" được biểu diễn như: $\forall x(\text{Cat}(x) \rightarrow \text{Mammal}(x))$.
 - Diễn giải: Nếu miền của chúng ta bao gồm Spot, Rebecca và Felix thì điều này có nghĩa là: $(\text{Cat}(\text{Spot}) \rightarrow \text{Mammal}(\text{Spot})) \wedge (\text{Cat}(\text{Rebecca}) \rightarrow \text{Mammal}(\text{Rebecca})) \wedge (\text{Cat}(\text{Felix}) \rightarrow \text{Mammal}(\text{Felix}))$.
- *Lượng từ tồn tại (Existential Quantification)*:
 - Ký hiệu: \exists
 - Ý nghĩa: "Có tồn tại"

- Ví dụ: "Spot có một người chị là mèo" được biểu diễn như:
 $\exists x(\text{Sister}(x, \text{Spot}) \wedge \text{Cat}(x))$.
- Diễn giải: Nếu miền của chúng ta bao gồm Spot, Rebecca và Felix thì điều này có nghĩa là: $(\text{Sister}(\text{Spot}, \text{Spot}) \wedge \text{Cat}(\text{Spot})) \vee (\text{Sister}(\text{Rebecca}, \text{Spot}) \wedge \text{Cat}(\text{Rebecca})) \vee (\text{Sister}(\text{Felix}, \text{Spot}) \wedge \text{Cat}(\text{Felix}))$.
- *Lượng từ lồng nhau (Nested Quantifiers)*:
 - Ví dụ: "Mọi người đều yêu ai đó" được biểu diễn như: $\forall x \exists y (\text{Loves}(x, y))$.
 - Thứ tự quan trọng: "Có một người được mọi người yêu" được biểu diễn như: $\exists y \forall x (\text{Loves}(x, y))$.
- *Phủ định (Negation)*: Các lượng từ này liên quan với nhau thông qua các định luật De Morgan:
 - $\forall x \neg \text{Likes}(x, \text{Parsnips}) \equiv \neg \exists x \text{Likes}(x, \text{Parsnips})$.

3. Mở rộng và biến thể ký hiệu

- *Logic bậc cao (Higher-order logic)*: Trong logic bậc cao, chúng ta có thể nói rằng hai đối tượng bằng nhau nếu và chỉ nếu tất cả các thuộc tính được áp dụng cho chúng là tương đương. Ví dụ:

$$\forall x \forall y (x=y) \leftrightarrow (\forall p (p(x) \leftrightarrow p(y)))$$
- *Biểu thức Lambda (λ)*: Để tạo các vị từ và hàm phức tạp từ những cái đơn giản hơn.
 - Ví dụ hàm: $\lambda xy (x^2 - y^2)$
 - Ứng dụng: $\lambda xy (x^2 - y^2)(25, 24) = 25^2 - 24^2 = 49$
 - Ví dụ vị từ: $\lambda xy (\text{Gender}(x) \neq \text{Gender}(y) \wedge \text{Address}(x) = \text{Address}(y))$
- *Ký hiệu duy nhất (Uniqueness Quantifier - $\exists!$)*:
 - Mục đích: Khẳng định sự tồn tại của một đối tượng duy nhất thỏa mãn một vị từ.
 - Ký hiệu: $\exists! x \text{King}(x)$
 - Ý nghĩa: "Có đúng một vua x."
 - Tương đương: $\exists x (\text{King}(x) \wedge \forall y (\text{King}(y) \rightarrow y = x))$
 - Ví dụ: $\forall c \text{Country}(c) \rightarrow \exists! r \text{Ruler}(r, c)$
 - Ý nghĩa: Mỗi quốc gia có đúng một người cai trị.
- *Ký hiệu Iota (ι)*: Để biểu thị một đối tượng duy nhất thỏa mãn một vị từ.
 - Ký hiệu: $\iota x P(x)$
 - Ví dụ: "Người cai trị duy nhất của Freedonia đã chết" có thể được viết như: $\text{Dead}(\iota r \text{Ruler}(r, \text{Freedonia}))$
 - Tương đương: $\exists! r \text{Ruler}(r, \text{Freedonia}) \wedge \text{Dead}(r)$
- *Biến thể ký hiệu (Notational Variations)*:
 - Phủ định (Not):

- Ví dụ: $\neg P, \sim P$
- Hội (And):
 - Ví dụ: $P \wedge Q, P \& Q, P \text{ Q}, PQ$
- Tuyển (Or):
 - Ví dụ: $P \vee Q, P + Q, P ; Q$
- Hàm ý (If):
 - Ví dụ: $P \rightarrow Q, P \supset Q$
- Tương đương (Iff):
 - Ví dụ: $P \leftrightarrow Q, P \equiv Q$
- Lượng từ phổ quát (All):
 - Ví dụ: $\forall xP(x), \wedge xP(x)$
- Lượng từ tồn tại (Exists):
 - Ví dụ: $\exists xP(x), \vee xP(x)$
- Quan hệ (Relation):
 - Ví dụ: $R(x,y), (R \text{ x } y), Rxy, xRy$

4. Sử dụng logic bậc nhất: Biểu diễn quan hệ gia đình và tập hợp toán học

- *Miền quan hệ họ hàng*: Chúng ta có thể ghi lại mỗi hàm và vị từ theo các ký hiệu khác. Ví dụ, mẹ của một người là cha mẹ nữ của người đó:

$$\forall m \forall c (Mother(c)=m \leftrightarrow Female(m) \wedge Parent(m,c))$$

- *Tiên đề, Định nghĩa và Định lý*:
 - Tiên đề (Axioms): Các sự thật cơ bản về một miền, là các sự thật nền tảng trong một cơ sở tri thức.
 - Định lý (Theorems): Được suy ra từ các tiên đề và định nghĩa.
 - Định nghĩa (Definitions): Tiên đề định nghĩa các vị từ; ví dụ:

$$\forall x \forall y (Sibling(x,y) \leftrightarrow Sibling(y,x))$$
- *Miền các tập hợp*: Các yếu tố cơ bản bao gồm các hằng số (ví dụ: EmptySet), các vị từ (ví dụ: Member, Subset) và các hàm (ví dụ: Intersection, Union, Adjoin).
- *Một số tiên đề chính cho các tập hợp*:
 - Tạo lập tập hợp: $\forall s (Set(s) \leftrightarrow (s=EmptySet) \vee (\exists x, s_2 (Set(s_2) \wedge s=Adjoin(x,s_2)))$
 - Tập rỗng: $\neg \exists x \exists s (Adjoin(x,s) - EmptySet)$
 - Phần tử ghép thêm: $\forall x \exists s (Member(x,s) \leftrightarrow s=Adjoin(x,s))$
 - Định nghĩa tập con:

$$\forall s_1 \forall s_2 (Subset(s_1,s_2) \leftrightarrow \forall x (Member(x,s_1) \rightarrow Member(x,s_2)))$$

5. Tác nhân logic cho thế giới WUMPUS

- Nhiệm vụ là định nghĩa giao diện giữa môi trường và tác nhân.

6. Tác nhân phản xạ đơn giản

- Loại tác nhân đơn giản nhất có các quy tắc trực tiếp kết nối các nhận thức với các hành động. Những quy tắc này giống như các phản xạ hoặc bản năng.

7. Biểu diễn sự thay đổi trong thế giới

- *Tính toán tình huống (Situation Calculus)*: Đại diện cho trạng thái của thế giới tại một thời điểm nhất định.
- *Hành động (Action)*: Chuyển đổi một tình huống thành một tình huống khác.
- *Tiên đề kết quả (Effect Axioms)*: Xác định điều gì là đúng trong tình huống kết quả sau một hành động.

8. Suy luận các thuộc tính ẩn của thế giới

- *Quy tắc nhân quả (Causal Rules)*: Phản ánh hướng nhân quả giả định trong thế giới: một số thuộc tính ẩn của thế giới gây ra các nhận thức nhất định được tạo ra.
- *Quy tắc chẩn đoán (Diagnostic Rules)*: Suy ra sự hiện diện của các thuộc tính ẩn trực tiếp từ thông tin nhận thức được.

9. Ưu tiên các hành động

- Cách duy nhất để quyết định các hành động là viết các quy tắc khuyến nghị chúng dựa trên các điều kiện nhất định trong thế giới.

10. Hướng đến tác nhân dựa trên mục tiêu

- Có ít nhất ba cách để tìm một chuỗi hành động như vậy:
 - *Suy luận (Inference)*: Viết các tiên đề cho phép hỏi KB để có một chuỗi hành động đảm bảo đạt được mục tiêu an toàn.
 - *Tìm kiếm (Search)*: Sử dụng quy trình tìm kiếm tốt nhất đầu tiên để tìm đường đến mục tiêu.
 - *Lập kế hoạch (Planning)*: Sử dụng các hệ thống lý luận đặc biệt được thiết kế để lý luận về hành động.

PHẦN II: Knowledge base

1. Kiến tạo cơ sở tri thức

- Kiến tạo cơ sở tri thức: là quá trình tổ chức, lưu trữ và quản lý thông tin đã được định dạng để hỗ trợ tìm kiếm, truy vấn và phân tích dữ liệu hiệu quả.
- Các cơ sở kiến trúc này có thể được ứng dụng trong nhiều lĩnh vực khác nhau như khoa học máy tính, trí tuệ nhân tạo, công nghệ thông tin, y tế, tài chính và nhiều lĩnh vực khác.

2. Đặc điểm của cơ sở tri thức tốt và xấu

- *Ngôn ngữ đại diện tri thức tốt:*
 - Phải rõ ràng, chính xác, không mập mờ và hiệu quả.
 - Các mối quan hệ quan trọng nên được định nghĩa rõ ràng và các chi tiết không liên quan cần được loại bỏ.
 - Sự tách biệt giữa cơ sở tri thức và quy trình suy luận cần được duy trì để người tạo cơ sở tri thức chỉ cần quan tâm đến nội dung tri thức.
- *Sai lầm phổ biến trong kỹ thuật tri thức:*
 - Chọn tên các thuộc tính có ý nghĩa với con người nhưng không rõ ràng đối với quy trình suy luận. Ví dụ, câu "BearOfVerySmallBrain(Pooh)" không giúp suy luận rằng: Pooh là gấu hoặc có bộ não nhỏ.
 - Kỹ sư tri thức sẽ nhận ra sai lầm này khi quy trình suy luận không thể kết luận chính xác.
- *Thiết kế cơ sở tri thức tốt:*
 - Đề suy luận đúng, cần tránh việc chỉ định tên thuộc tính quá cụ thể. Thay vào đó, nên sử dụng các khái niệm chung và tổng quát hơn.
 - Ví dụ, thay vì "BearOfVerySmallBrain(Pooh)", nên dùng các câu như "Pooh là gấu; gấu là động vật; động vật là vật thể vật lý."

Bear(Pooh)

$\forall b \text{ } \textit{Bear}(b) \Rightarrow \textit{Animal}(b)$

$\forall a \text{ } \textit{Animal}(a) \Rightarrow \textit{PhysicalThing}(a)$

- *Sử dụng các quy tắc thiết kế cơ sở tri thức:*

- Sử dụng các mối quan hệ chung và liên kết các khái niệm ở mức độ tổng quát để mở rộng khả năng tái sử dụng thông tin.

All animals (and only animals) have a brain, which is a part of the animal.

$\forall a \text{ } \textit{Animal}(a) \Leftrightarrow \textit{Brain}(\textit{BrainOf}(a))$

$\forall a \text{ } \textit{PartOf}(\textit{BrainOf}(a), a)$

If something is part of a physical thing, then it is also a physical thing:

$$\forall x, y \text{ PartOf}(x, y) \wedge \text{PhysicalThing}(y) \Rightarrow \text{PhysicalThing}(x)$$

- Ưu điểm của viết các quy tắc ở mức độ tổng quát nhất mà tri thức đó có thể áp dụng được:
 - Có thể tái sử dụng cơ sở tri thức cho nhiều nhiệm vụ khác nhau mà không cần sửa đổi.
 - Giảm lỗi cơ sở tri thức dễ dàng hơn vì mỗi câu đều đúng hoặc sai độc lập, trong khi tính đúng đắn của một câu lệnh chương trình phụ thuộc nhiều vào ngữ cảnh.

3. Kỹ thuật tri thức

- Kỹ sư tri thức phải hiểu đủ về lĩnh vực được đề cập để thể hiện các đối tượng và mối quan hệ quan trọng.
- Để giúp tập trung phát triển nền tảng kiến thức và tích hợp tư duy của kỹ sư ở ba cấp độ (nhận biết, thông hiểu, vận dụng cao), có thể sử dụng phương pháp năm bước sau:
 - Quyết định những gì để nói về.
 - Quyết định từ vựng về mệnh đề, hàm và hằng số.
 - Mã hóa kiến thức chung về miền.
 - Mã hóa mô tả của trường hợp vấn đề cụ thể.
 - Đặt truy vấn cho thủ tục suy luận và nhận câu trả lời.
- Kỹ thuật tri thức so với lập trình:
 - Cả hai hoạt động có thể được coi là bao gồm bốn bước:

Kỹ thuật tri thức	Lập trình
Lựa chọn logic	Lựa chọn ngôn ngữ lập trình
Xây dựng nền tảng kiến thức	Viết một chương trình
Thực hiện lý thuyết chứng minh	Chọn hoặc viết một trình biên dịch
(Suy ra sự thật mới)	Chạy một chương trình

- Mặc dù có sự tương đồng, kỹ thuật tri thức có những lợi thế riêng:
 - Kỹ sư tri thức chỉ cần quyết định đối tượng và mối quan hệ nào đáng được đại diện, và mối quan hệ nào giữ giữa các đối tượng. Trái lại, lập trình viên phải xác định cách tính toán các mối quan hệ đó từ đầu vào ban đầu.
 - Kỹ sư tri thức chỉ định điều gì là đúng, và quy trình suy luận sẽ tìm ra cách chuyển đổi các sự kiện thành giải pháp cho vấn đề.
 - Cơ sở tri thức có thể được tái sử dụng cho nhiều nhiệm vụ khác nhau mà không cần sửa đổi.

- Việc gỡ lỗi cơ sở tri thức dễ dàng hơn vì mỗi câu đều đúng hoặc sai độc lập, trong khi tính đúng đắn của một câu lệnh chương trình phụ thuộc rất nhiều vào ngữ cảnh của nó.

PHẦN III - Ứng dụng First-order logic và Knowledge base

1. Giới thiệu ứng dụng

- Qua các phần trên, chúng ta có thể hiểu rõ phần nào First-order logic và Knowledge base.
⇒ Từ những kiến thức đó, chúng em làm 1 demo ứng dụng có tác dụng xây dựng và sử dụng một hệ thống suy luận dựa trên logic để trả lời các truy vấn về cơ sở tri thức (Knowledge Base - KB).
- Chương trình của ứng dụng mà chúng em làm gồm:
 - *Data*: Lưu trữ file giữ liệu đầu vào và đầu ra (KnowledgeBase.txt, Queries.txt, transferQueries.txt, transferSentences.txt)
 - *loadData.py*: Đọc các câu từ file văn bản và trả về danh sách các câu.
 - *Transfer.py*: Dùng để chuyển các câu sẽ được chuyển sang dạng logic bằng cách sử dụng model gpt-3.5-turbo thông qua gọi API.
 - *engine.py*: Thực hiện suy luận logic mệnh đề bậc nhất (First-Order Predicate Logic) trong một hệ thống cơ sở tri thức (Knowledge Base - KB)

2. Các thành phần chính trong ứng dụng

- *Định nghĩa các lớp và hàm hỗ trợ xử lý logic*:
 - *Parameter*: Quản lý các biến và hằng số trong các câu logic.
 - *Predicate*: Quản lý các vị từ (predicates) trong các câu logic.
 - *Sentence*: Quản lý các câu logic, bao gồm các vị từ và các biến, chuyển đổi câu sang dạng chuẩn tắc (CNF - Conjunctive Normal Form).
 - *KB*: Quản lý cơ sở tri thức, lưu trữ các câu logic, chuyển đổi câu sang CNF, và thực hiện suy luận để trả lời truy vấn.
- *Các hàm hỗ trợ khác*:
 - *performUnification*: Thực hiện việc hợp nhất (unification) các vị từ để kiểm tra sự đồng nhất và tạo ra các phép thay thế.
 - *negatePredicate*, *negateAntecedent*: Phủ định vị từ và các mệnh đề logic.

- *Đọc và ghi dữ liệu từ file:*
 - `read_sentences_from_file`: Đọc các câu từ file văn bản và trả về danh sách các câu.
 - `getKB`, `getQueries`: Đọc cơ sở tri thức và truy vấn từ file.
 - `printOutput`: Ghi kết quả truy vấn vào file.
- *Tích hợp với OpenAI API:*
 - `UpdateKB`, `UpdateQueries`: Sử dụng OpenAI API để chuyển đổi các câu truy vấn và câu tri thức từ ngôn ngữ tự nhiên sang dạng logic chuẩn tắc.
- *Phần thực hiện của ứng dụng:*
 - Tạo ra cơ sở tri thức và truy vấn từ các file đầu vào.
 - Chuẩn bị cơ sở tri thức và thực hiện các truy vấn.
 - Ghi kết quả truy vấn vào file đầu ra.

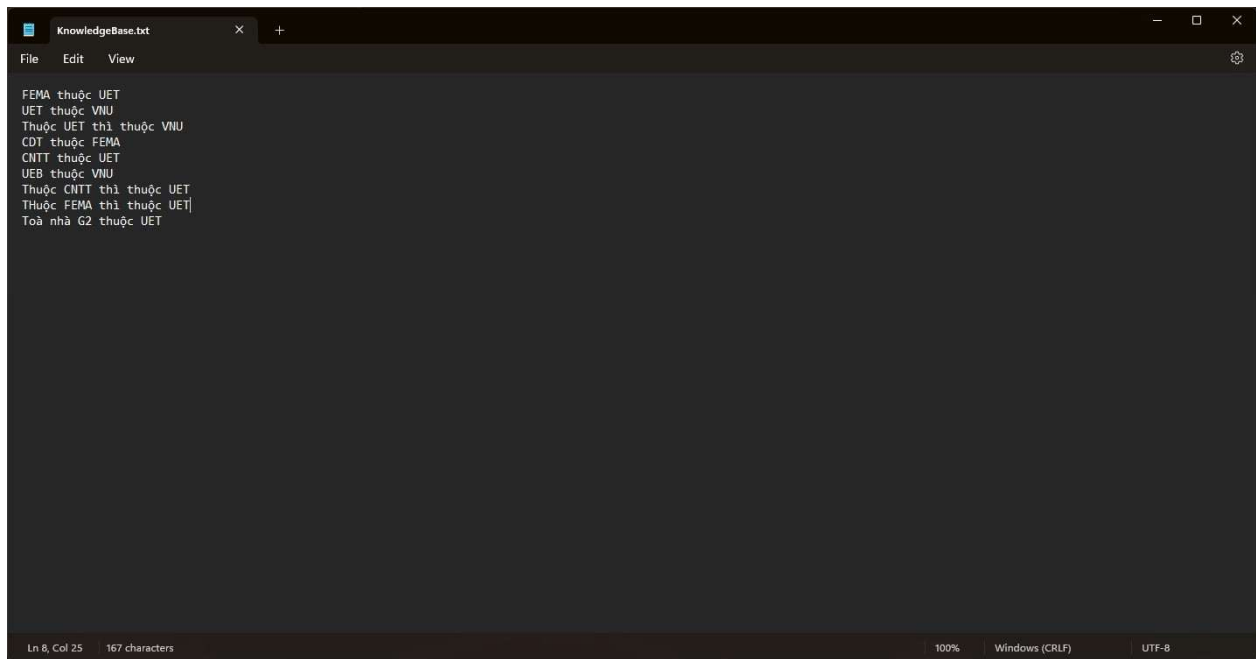
3. Cách thức hoạt động của ứng dụng

- **Bước 1:** Đọc dữ liệu: Đọc các câu và truy vấn từ các file văn bản.
- **Bước 2:** Chuẩn bị dữ liệu: Sử dụng OpenAI API để chuyển đổi các câu từ ngôn ngữ tự nhiên sang dạng logic.
- **Bước 3:** Xây dựng cơ sở tri thức: Chuyển đổi các câu logic sang dạng CNF và lưu trữ chúng trong cơ sở tri thức.
- **Bước 4:** Thực hiện suy luận: Trả lời các truy vấn dựa trên cơ sở tri thức đã chuẩn bị bằng cách sử dụng các phương pháp logic như hợp nhất và phủ định.
- **Bước 5:** Ghi kết quả: Ghi lại kết quả của các truy vấn vào file.

4. Dữ liệu và chương trình của ứng dụng

- Mọi người có thể tham khảo qua link Github sau:
<https://github.com/NCQH/First-Oder-Logic-And-Knowledge-Base>

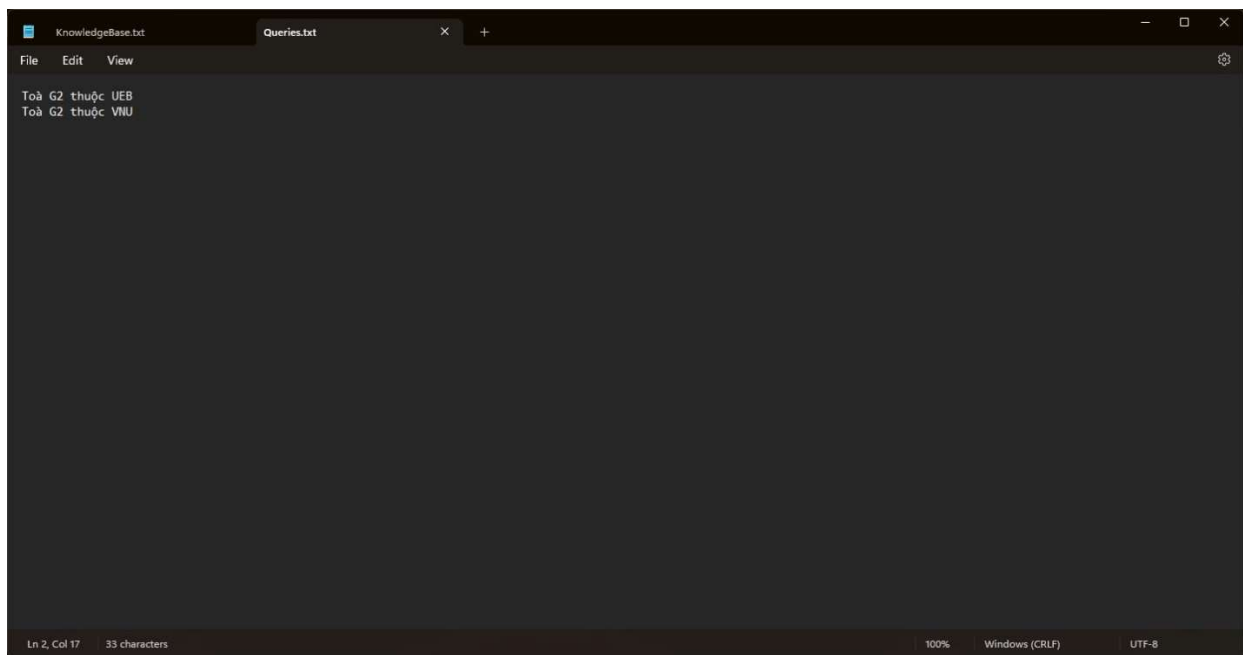
5. Kết quả



A screenshot of a text editor window titled 'KnowledgeBase.txt'. The editor has a dark theme and shows a list of knowledge base entries. The status bar at the bottom indicates 'Ln 8, Col 25' and '167 characters'.

```
FEMA thuộc UET
UET thuộc VNU
Thuộc UET thì thuộc VNU
CĐT thuộc FEMA
CNTT thuộc UET
UEB thuộc VNU
Thuộc CNTT thì thuộc UET
Thuộc FEMA thì thuộc UET
Toà nhà G2 thuộc UET
```

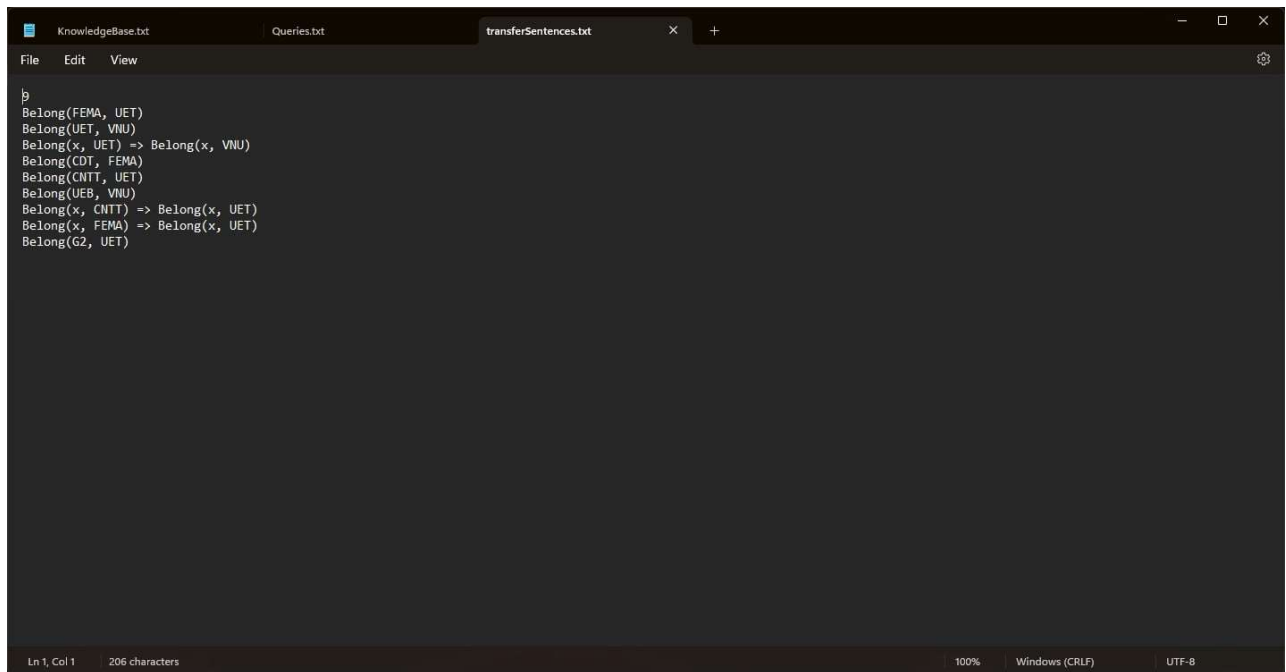
Các câu thuộc cơ sở tri thức



A screenshot of a text editor window titled 'KnowledgeBase.txt' with a sub-tab 'Queries.txt'. The editor shows a list of queries. The status bar at the bottom indicates 'Ln 2, Col 17' and '33 characters'.

```
Toà G2 thuộc UEB
Toà G2 thuộc VNU
```

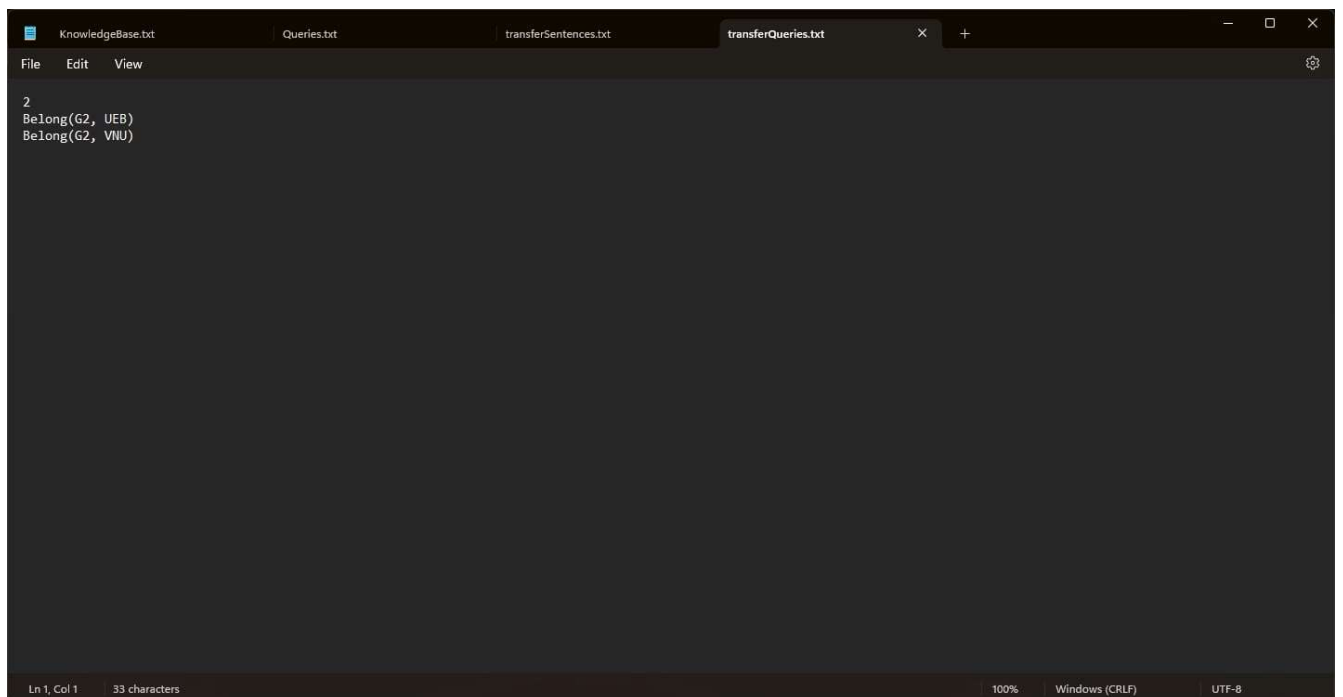
Các mục queries



A screenshot of a text editor window with three tabs: KnowledgeBase.txt, Queries.txt, and transferSentences.txt. The transferSentences.txt tab is active, displaying a list of logical sentences. The status bar at the bottom indicates 'Ln 1, Col 1' and '206 characters'.

```
9
Belong(FEMA, UET)
Belong(UET, VNU)
Belong(x, UET) => Belong(x, VNU)
Belong(CDT, FEMA)
Belong(CNTT, UET)
Belong(UEB, VNU)
Belong(x, CNTT) => Belong(x, UET)
Belong(x, FEMA) => Belong(x, UET)
Belong(G2, UET)
```

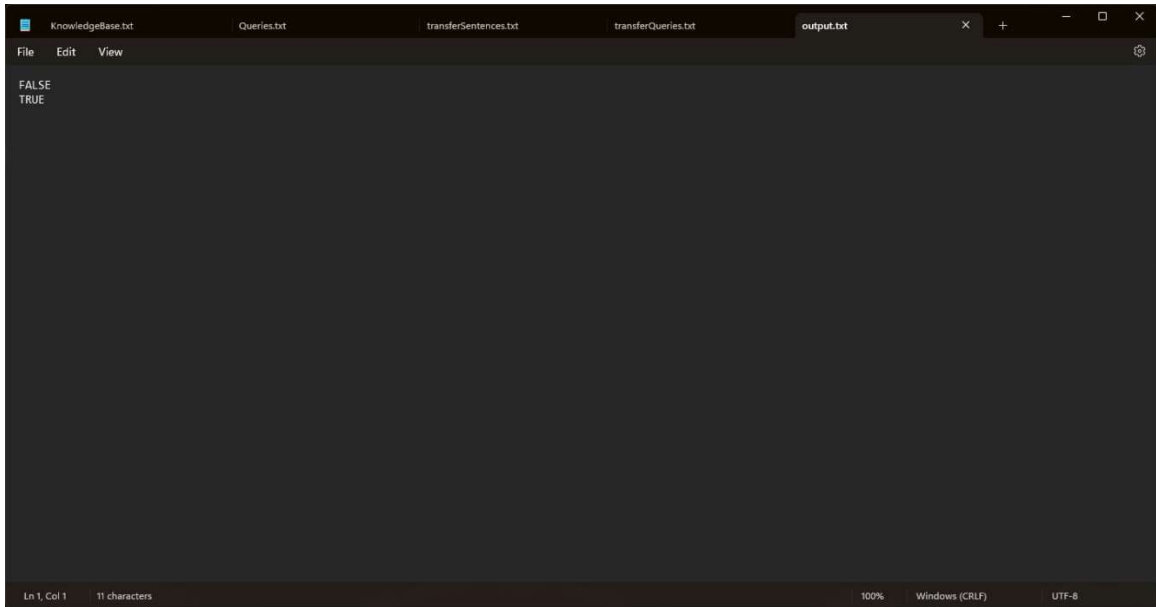
Các câu sẽ được chuyển sang dạng logic bằng cách sử dụng model gpt-3.5-turbo thông qua gọi API



A screenshot of a text editor window with four tabs: KnowledgeBase.txt, Queries.txt, transferSentences.txt, and transferQueries.txt. The transferQueries.txt tab is active, displaying a list of logical queries. The status bar at the bottom indicates 'Ln 1, Col 1' and '33 characters'.

```
2
Belong(G2, UEB)
Belong(G2, VNU)
```

Các queries được chuyển sang dạng logic bằng cách sử dụng model gpt-3.5-turbo thông qua gọi API



Kết quả trả lời các truy vấn

PHẦN IV: Tham khảo

1. Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd. ed.). Prentice Hall Press, USA.
2. https://github.com/Keerthivasan13/CSCI561_Artificial_Intelligence