**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN: **TRÍ TUỆ NHÂN TẠO (Artificial Intelligence: AI)**

MÃ SỐ LỚP HP: **ARIN330585 - Nhóm 04 (Sáng thứ 2)**

Tên đề tài: **PHÁT TRIỂN HỆ THỐNG CHATBOT NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI BẰNG TIẾNG ANH**

Họ tên sinh viên: **Trịnh Công Trưởng**

**Mã số sinh viên:19110492[50]**

**Lớp: 191103B**

**Ngày nộp: 19/06/2021**

**Ký tên: Truong**

**TP.HCM, ngày 08 tháng 06 năm 2021**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

TÊN HỌC PHẦN: **TRÍ TUỆ NHÂN TẠO (Artificial Intelligence: AI)**

MÃ SỐ LỚP HP: **ARIN330585 - Nhóm 04 (Sáng thứ 2)**

Tên đề tài: **PHÁT TRIỂN HỆ THỐNG CHATBOT NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO.**

Họ tên sinh viên: **Trịnh Công Trưởng**

**Mã số sinh viên: 19110492[50]**

**Lớp: 191103B**

**Tên sản phẩm đề tài: N4S2.50.TrinhCongTruong.19110492.DoaAnHP.AI.rar**

**Công cụ sử dụng (phiên bản): Spyder4 (Anaconda3)**

**Ngôn ngữ lập trình sử dụng: Python**

**Tập dữ liệu thực nghiệm : File intents.json đính kèm**

**Nhận xét của giảng viên:**

**………………………………………………………..**

**………………………………………………………..**

**Điểm đánh giá:………….(…………………)**

**Ngày……./……../2021**

**Giảng viên Ký tên**

**TP.HCM, ngày 08 tháng 06 năm 2021**

# LỜI CẢM ƠN

Cảm ơn thầy Võ Xuân Thể đã dạy các nội dung lý thuyết và thực hành để giúp em làm đồ án được thuận lợi hơn.

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
|  | |
| CSDL hoặc DB | Cơ sở dữ liệu: DataBase |
|  | |
| AI | Trí tuệ (Trí thông minh) nhân tạo: Artificial Intelligence |
| ANN | Mạng nơ ron nhân tạo: Artificial Neural Network |
| ES | Hệ thống chuyên gia: Expert Systems |
| ML | Máy học = Học máy: Machine Learning |
| NLP | Xử lý ngôn ngữ tự nhiện: Natural Language Processing |

# DANH MỤC CÁC THUẬT NGỮ

|  |  |
| --- | --- |
| Bài toán (vấn đề) AI:  AI Problem | Là các tình huống thực tế mà con người cần giải quyết thuộc một lĩnh vực nào đó, thông thường phải dựa vào “trí khôn” con người. |
| Giao diện người dùng  (User-Interface) | Là hệ thống các màn hình giao tiếp cho phép người sử dụng tương tác với các thành phần phần mềm, điều khiển phần mềm hoạt động theo yêu cầu của người dùng - tương ứng các chức năng hiện có của phần mềm. |
| Người dùng (User):  Tài khoản (Account) | Là một quyền làm việc trên hệ thống phần mềm được cấp phát cho một cá nhân thông qua tên tài khoản (username) và mật khẩu (password). |
|  |  |
| BigData | Dữ liệu lớn: là một tập hợp dữ liệu rất lớn và phức tạp, không thể xử lý dữ liệu bằng các phương pháp truyền thống. |
| Heuristics | Là các kỹ thuật dựa trên kinh nghiệm để giải quyết vấn đề, nhằm đưa ra một giải pháp mà không được đảm bảo là tối ưu |

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc74035245)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc74035246)

[DANH MỤC CÁC THUẬT NGỮ 5](#_Toc74035247)

[MỤC LỤC 6](#_Toc74035248)

[PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN 9](#_Toc74035249)

[Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI 10](#_Toc74035250)

[**1.1.** Tổng quan về đề tài 10](#_Toc74035251)

[**1.2.** Nội dung chuyên môn chính của đề tài 10](#_Toc74035252)

[**1.3.** Ngôn ngữ lập trình và công cụ sử dụng 10](#_Toc74035253)

[**1.4.** Sản phẩm của đề tài 10](#_Toc74035254)

[**1.5.** Bố cục của báo cáo 10](#_Toc74035255)

[Chương 2: CƠ SỞ LÝ THUYẾT VỀ TRÍ TUỆ NHÂN TẠO 12](#_Toc74035256)

[**2.1.** Tổng quan về Trí tuệ nhân tạo (AI) 12](#_Toc74035257)

[2.1.1. Các khái niệm 12](#_Toc74035258)

[2.1.2. Vai trò AI 13](#_Toc74035259)

[2.1.3. Nền tảng kỹ thuật của AI 14](#_Toc74035260)

[2.1.4. Các lĩnh vực nghiên cứu và ứng dụng cơ bản của AI 14](#_Toc74035261)

[**2.2.** Giới thiệu về các bài toán (vấn đề) AI 14](#_Toc74035262)

[2.2.1. Một số khái niệm 14](#_Toc74035263)

[2.2.2. Tìm kiếm lời giải (searching) cho AI Problem 17](#_Toc74035264)

[**2.3.** Biểu diễn một số bài toán AI 17](#_Toc74035265)

[2.3.1. Các bài toán Trò chơi 17](#_Toc74035266)

[2.3.2. Cây không gian trạng thái của Bài toán AI 18](#_Toc74035267)

[**2.4.** Các phương pháp biểu diễn tri thức 18](#_Toc74035268)

[2.4.1. Phân loại tri thức theo phương pháp biểu diễn 18](#_Toc74035269)

[2.4.2. Phân loại tri thức theo nền tảng hình thành Cơ sở tri thức 19](#_Toc74035270)

[2.4.3. Logic mệnh đề và Logic vị từ 21](#_Toc74035271)

[2.4.4. Biểu diễn tri thức bằng Frame (Cấu trúc = "Khung") 24](#_Toc74035272)

[2.4.5. Suy diễn tri thức bằng luật dẫn xuất 25](#_Toc74035273)

[2.4.6. Biểu diễn suy luận bằng đồ thị AND/OR 26](#_Toc74035274)

[2.4.7. Các biểu diễn tương đương trong Logic mệnh đề và Logic vị từ 27](#_Toc74035275)

[2.4.8. Xử lý ngôn ngữ tự nhiên (Natural Language Processing = NLP): Voice Assistant 31](#_Toc74035276)

[**2.5.** Các phương pháp tìm kiếm lời giải của bài toán AI 31](#_Toc74035277)

[2.5.1. Giới thiệu 31](#_Toc74035278)

[2.5.2. Các phương pháp tìm kiếm mù (Blind Search) = tìm kiếm không có thêm thông tin (uninformed search) 32](#_Toc74035279)

[2.5.3. Các phương pháp tìm kiếm theo kinh nghiệm (Heuristic Search) = tìm kiếm với thông tin bổ sung (Informed Search) = Cải tiến BFS 36](#_Toc74035280)

[2.5.4. Lập trình Logic (Logic Programming) phát triển cơ chế lập luận logic 39](#_Toc74035281)

[Chương 3: GIỚI THIỆU VỀ MÔ HÌNH BÀI TOÁN AI: NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO 40](#_Toc74035282)

[3.1. Mô tả bài toán AI 40](#_Toc74035283)

[3.2. Các yếu tố xác định bài toán AI 40](#_Toc74035284)

[3.3. Giới thiệu nguồn dữ liệu gốc dùng thực nghiệm bài toán AI 40](#_Toc74035285)

[3.4. Một số tiền xử lý dữ liệu nguồn dùng thực nghiệm bài toán AI 41](#_Toc74035286)

[Chương 4: BIỂU DIỄN BÀI TOÁN AI VÀ CƠ SỞ TRI THỨC CỦA AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO 44](#_Toc74035287)

[4.1. Giới thiệu 44](#_Toc74035288)

[4.2. Biểu diễn bài toán AI 44](#_Toc74035289)

[4.3. Xây dựng cơ sở tri thức (KBS) cho bài toán AI 44](#_Toc74035290)

[4.4. Lập trình xử lý ngôn ngữ tự nhiên (Natural Language Processing = NLP): Voice Assistant (trợ lý ảo) 44](#_Toc74035291)

[Chương 5: PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI CHO BÀI TOÁN AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO 46](#_Toc74035292)

[5.1. Giới thiệu 46](#_Toc74035293)

[5.2. Mô tả sơ bộ về mô hình cây không gian trạng thái của bài toán AI 46](#_Toc74035294)

[5.3. Xác định phương pháp tìm kiếm lời giải cho bài toán AInghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo 46](#_Toc74035295)

[5.4. Cài đặt (lập trình) phương pháp tìm kiếm lới giải cho bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo 47](#_Toc74035296)

[Chương 6: GIỚI THIỆU SẢN PHẨM CỦA ĐỀ TÀI AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO 48](#_Toc74035297)

[6.1. Giới thiệu sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo 48](#_Toc74035298)

[6.2. Kết quả thực nghiệm sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo 54](#_Toc74035299)

[6.3. Nhận xét và đánh giá về sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo. 54](#_Toc74035300)

[Chương 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 55](#_Toc74035301)

[7.1. Kết luận 55](#_Toc74035302)

[7.1.1. Những kết quả đạt được 55](#_Toc74035303)

[7.1.2. Hạn chế 55](#_Toc74035304)

[7.2. Hướng phát triển 55](#_Toc74035305)

[7.2.1. Hướng khắc phục các hạn chế 55](#_Toc74035306)

[7.2.2. Hướng mở rộng đề tài 55](#_Toc74035307)

[DANH MỤC TÀI LIỆU THAM KHẢO 56](#_Toc74035308)

[CÁC PHỤ LỤC 57](#_Toc74035309)

[ĐÁNH GIÁ KẾT QUẢ HỌC TẬP HỌC PHẦN 58](#_Toc74035310)

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

# PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN

|  |  |
| --- | --- |
| Giảng viên giảng dạy: **VÕ XUÂN THỂ** | |
| Tên học phần: **Trí tuệ nhân tạo (AI)**  Mã số lớp HP: **ARIN330585 – N4S2** | |
| Tên đề tài: **Phát triển hệ thống chatbot nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo (AI).** | |
| Sinh viên thực hiện: **Trịnh Công Trưởng, 19110492** | |
| Thời gian thực hiện: **22/02/2021**  đến **08/06/2021** | |
| **Yêu cầu của đề tài**  **Lý thuyết:** Vận dụng AI vào thực tiễn và tiếp cận thành tựu tiên tiến trong lĩnh vực AI:  + Nền tảng và đặc trưng của các dạng bài toán AI và giải thuật tương ứng.  + Một số phương pháp biểu diễn không gian (trạng thái) giải quyết bài toán AI: biểu diễn tri thức và suy diễn logic.  + Một số phương pháp thông dụng trong tìm kiếm lời giải bài toán AI = giải quyết bài toán thông qua tìm kiếm lời giải (Solving Problems by Searching), đặc biệt là DFS và BFS,…  **Thực hành:** Phát triển hệ thống chatbot dựa vào các giải thuật trí tuệ nhân tạo (AI):  + Thư viện chính được dùng: NLTK, Keras, speechRecognition, playsound, gTTS  + Giải thuật chính: Mô hình túi từ, ANN  + Các chức năng chính của sản phẩm: Đưa ra câu trả lời (bằng chữ và giọng nói) tùy theo nội dung người dùng nhập vào, câu trả lời phải phù hợp nội dung người dùng nhập vào  + Tập thực nghiệm: File intents.json đính kèm | |
| **GIẢNG VIÊN** | **Ngày 22 tháng 02 năm 2021**  **SV Thực hiện** |

# GIỚI THIỆU VỀ ĐỀ TÀI

## Tổng quan về đề tài

Đề tài này làm về hệ thống chatbot giao tiếp cơ bản bằng tiếng Anh. Giải thuật chính được sử dụng là giải thuật mạng nơron nhân tạo. Thư viện được sử dụng chủ yếu ở đây là NLTK và Keras

## Nội dung chuyên môn chính của đề tài

+ Thư viện: NLTK(tiền xử lý văn bản),Keras(tạo mô hình mạng nơron nhân tạo),json(nạp file dữ liệu để train cũng như để chạy),pickle(dùng để lưu lại các đối tượng),numpy(lưu dữ liệu training),random(xáo trộn dữ liệu training),tkinter(tạo giao diện đồ họa người dùng), speechRecognition (chuyển giọng nói thành văn bản), playsound(chạy file âm thanh), gTTS (chuyển văn bản thành giọng nói bằng Google Translate)

+ Giải thuật gì: Mạng nơ-ron nhân tạo là một chuỗi các thuật toán được đưa ra để nỗ lực tìm kiếm các mối quan hệ cơ bản trong một tập hợp dữ liệu, thông qua quá trình bắt chước cách thức hoạt động của bộ não con người. Mạng nơ-ron nhân tạo có thể thích ứng với các thay đổi trong đầu vào, do đó, nó đưa ra các kết quả tốt nhất có thể mà không cần phải thiết kế lại các tiêu chí đầu ra.

+ Sản phẩm đề tài: Chatbot nghe và phản hồi bằng tiếng Anh

## Ngôn ngữ lập trình và công cụ sử dụng

+ Ngôn ngữ lập trình: Python

+ Công cụ sử dụng Spyder 4 , Python 3.8

## Sản phẩm của đề tài

Chatbot nghe và đưa ra câu trả lời (bằng chữ và giọng nói) tùy theo nội dung người dùng nhập vào.

## Bố cục của báo cáo

Chương 1: Giới thiệu tổng quan về đề tài

Chương 2: Cơ sở lý thuyết của học phần: gồm những lý thuyết cơ bản về AI liên quan đến đề tài.

Chương 3: Giới thiệu về thư viện và giải thuật AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo.

Chương 4: Mô hình bài toán AI và cài đặt giải thuật tìm kiếm lời giải

Chương 5: Giới thiệu về sản phẩm hệ thống AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo.

Chương 6: Kết luận về kết quả đạt được và những tồn tại, trên cơ cở đó đề xuất các giải pháp khắc phục tồn và hướng mở rộng đề tài.

# CƠ SỞ LÝ THUYẾT VỀ TRÍ TUỆ NHÂN TẠO

## Tổng quan về Trí tuệ nhân tạo (AI)

### Các khái niệm

* + - 1. Trí tuệ nhân tạo = AI (Artificial Intelligence)

Là ngành khoa học nghiên cứu và ứng dụng các nên tảng kỹ thuật làm cơ sở cho việc thiết kế và sản xuất ra các hệ thống, đặc biệt là các hệ thống CNTT, có khả năng thực hiện các hành vi “khôn” như con người

+ Biểu hiện “khôn” của các hệ thống AI: “hành vi” và “nhận thức”/”suy luận”

+ Đặc trưng của các hệ thống AI:

* Kết quả không nhất thiết chính xác 100%, tỷ lệ sai số là tùy thuộc vào bài toán AI
* Không nhất thiết phải chứng minh, giải thích kết quả, mục tiêu là hiệu quả
* Có khả năng “học” và “tự học”
* “Khôn” dần theo thời gian
* Do “chuyên môn” và “lập trình” cùng xây dựng và phát triển
* Sử dụng ngôn ngữ và công cụ lập trình riêng
  + - 1. **Phân loại hệ thống Trí tuệ nhân tạo**

- Dựa vào “nền tảng”: Hueristics và ES

- Biểu hiện: “hành vi” và “nhận thức”/”suy luận”

- Dựa vào nền tảng kỹ thuật: ML, NLP, ANN, Hệ thống mờ, …

* + - 1. **Turing Test kiểm tra khả năng khôn của các hệ thống AI**
      2. **Tác tử (Agent: Intelligent Agent) trong hệ thống AI**
      3. **Tri thức là gì (Knowledge)?**

- Tri thức là hệ thống sự hiểu biết bằng lý thuyết hay thực tế về một chủ đề hay lĩnh vực

- Tri thức tự nhiên hoặc tri thức hệ thống [sản phẩm AI là tri thức hệ thống]

- Phân biệt phần mềm truyền thống và phần mềm trí tuệ nhân tạo

+ chương trình truyền thống

= cấu trúc dữ liệu + giải thuật

+ chương trình trí tuệ nhân tạo=

cơ sở tri thức (knowledge base) + qui tắc suy diễn (inference engine)

* + - 1. **Cơ sở tri thức (Knowledge Base: KB)**

- Cơ sở tri thức là tập hợp các tri thức liên quan đến vấn đề quan tâm được hệ thống và lưu trữ trên các thiết bị số (máy tính) dưới một dạng nào đó

- Cơ sở tri thức chứa các kiến thức được sử dụng để giải quyết các vấn đề (bài toán) trong trí tuệ nhân tạo.

|  |  |
| --- | --- |
| Database | Knowledge base |
| Record = Bản ghi = Tuple = Bộ dữ liệu  01 KennyThe 2000 1 | Rules = pattern = luật/mẫu  If ………. then………… |
| Nhiều Record -> Data Warehouse -> Big Data | Đúc kết từ Data Warehouse, Big Data để tạo nên các Rules -> Data Mining |
|  |  |

* + - 1. **Hệ quản trị cơ sở tri thức (KBMS)**

- Hệ quản trị cơ sở tri thức thực chất là một sản phẩm phần mềm mà trong đó có 2 nhóm công cụ và phương tiện cơ bản

+ Các công cụ, phương tiện hỗ trợ cho việc hình thành, duy trì và phát triển các hệ thống cơ sở tri thức

+ Các công cụ, phương tiện hỗ trợ việc khai thác, sử dụng hệ thống cơ sở tri thức đã có vào trong thực tiễn một cách hiệu quả

|  |  |
| --- | --- |
| DBMS | KBMS |
| Quản lý DB (New, Use) | Quản lý KB (New,Use) |
|  |  |

- Cơ chế suy diễn: là phương pháp vận dụng tri thức trong cơ sở tri thức trong cơ sở tri thức để giải quyết vấn đề

-> Cơ chế suy diễn là các công cụ, phương thức thuộc nhóm khai thác sử dụng cơ sở tri thức trong KBMS

### Vai trò AI

- Là nhân tố trung tâm và gần như không thể thiếu trong các hệ thống hiện nay và tương lai

- AI là yếu tố nền tảng mang tính bắt buộc của các hệ thống trong thời đại cách mạng công nghiệp 4.0 ( cách mạng công nghiệp lần thứ 4: IR 4.0 = The Fourth Industrial Revolution)

### Nền tảng kỹ thuật của AI

- Cơ sở tri thức (KB)

- Cơ chế: suy diễn|| suy luận và lập luận

- Xử lý ngôn ngữ tự nhiên (Natural Language Processing = NLP) Automata

- ML = Machine Learning

- Các lý thuyết cơ sở

+ Lý thuyết giải bài toán và suy diễn thông minh

+ Lý thuyết tìm kiếm may rủi (ngẫu nhiên)

+ Các ngôn ngữ công cụ về Trí tuệ nhân tạo

+ Lý thuyết biểu diễn tri thức và hệ chuyên gia

+ Lý thuyết nhận dạng và xử lý tiếng nói: Computer Vision (CV)

+ Người máy (ROBOT)

+ Tâm lý học xử lý thông tin (Bio-inspired Computing)

### Các lĩnh vực nghiên cứu và ứng dụng cơ bản của AI

- Game

- Tối ưu

- Dự đoán, dự báo

## Giới thiệu về các bài toán (vấn đề) AI

### Một số khái niệm

* + - 1. **AI Problem (Bài toán AI = Vấn đề AI)**

##### - Bài toán AI:

Là tình huống thực tế mà con người cần giải quyết thuộc một lĩnh vực nào đó, thông thường phải dựa vào “trí khôn” con người.

VD: Game cờ tướng, bài toán tìm đường đi tốt nhất, Tư vấn pháp luật, dự đoán hành vi khách hàng,…

* + - 1. **Các yếu tố cơ bản xác định bài toán AI**

##### - Các yếu tố cơ bản xác định bài toán AI:

Một bài toán thông thường được xác định bởi 4 yếu tố cơ bản sau đây:

+ Tập hợp các trạng thái của bài toán: Mỗi trạng thái trong tập hợp đó là 1 tình huống của bài toán, nó có thể là một giải pháp của bài toán, không nhất thiết là giải pháp đúng và thậm chí nếu là giải pháp đúng cũng không nhất thiết là giải pháp đúng nhất.

VD: Trạng thái của bài toán cờ tướng có thể là trạng thái của toàn bộ bàn cờ hoặc là trạng thái của 1 nước đi tùy theo cách ta biểu diễn bài toán.

+ Trạng thái bắt đầu của bài toán: Trong số các trạng thái nêu trên, cần xác định 1 trạng thái bắt đầu hoặc một số trạng thái có thể bắt đầu.

VD: Có bao nhiêu quân cờ thì có bấy nhiêu trạng thái bắt đầu của bàn cờ tướng

+ Mục tiêu đặt ra ( thông qua hàm mục tiêu) đối với bài toán AI, thường được xác định qua hàm mục tiêu mà kỹ thuật AI phải giải quyết cho bài toán đó.

VD

Bài toán cờ tướng => Mục tiêu: “Chiếu tướng bí”

Rubik => Mục tiêu: 6 mặt cùng màu

Tìm đường đi => Mục tiêu: Tìm được 1 hoặc 1 số lộ trình đường đi mà người đi họ “hài lòng” nhất

+ Có cơ chế chuyển trạng thái: Quy tắc để chuyển từ “trạng thái” này sang “trạng thái” khác trong tập không gian trạng thái, việc chuyển “trạng thái” phải đảm bảo các quy tắc đó.

VD

Cờ tướng: Nước đi của 1 quân cờ của 1 trong 2 bên trên bàn cờ tướng xuất phát từ trạng thái ban đầu

Tháp Hà Nội: Quy tắc chuyển các đĩa từ cột này sang cột khác

Như vậy, việc cài đặt 1 bài toán trí tuệ nhân tạo thông thường phải đầy đủ 4 yếu tố trên.

* + - 1. **(Hàm) Mục tiêu của Bài toán AI**

##### - Hàm mục tiêu:

Là 1 cơ chế, 1 công thức hoặc 1 quy tắc,… để xác định 1 trạng thái nào đó trong không gian trạng thái của bài toán được xem là mục tiêu đạt được của bài toán đó; Tức là nếu 1 trạng thái nào đó trong không gian trạng thái mà thỏa mãn các điều kiện của mục tiêu thì được xem là 1 lời giải (giải pháp) của bài toán. Hàm mục tiêu của bài toán trí tuệ nhân tạo rất đa dạng và tùy thuộc vào từng bài toán cụ thể. Và thường có 2 dạng mục tiêu: mục tiêu “định lượng[[1]](#footnote-1)” và mục tiêu “định tính[[2]](#footnote-2)”, và khi mục tiêu đặt ra là mục tiêu định lượng thì có thể xác định bằng hàm và hàm đó được gọi là hàm mục tiêu.

VD: Với bài toán “xác định đường đi tốt nhất từ A đến B”

Mục tiêu để xác định lộ trình đường đi tốt nhất phụ thuộc vào nhiều tham số

(x) khoảng cách phải ngắn nhất (km)

(y) thời gian phải nhanh nhất (min)

(z) chi phí thấp nhất (VNĐ)

-> Các đối số của hàm mục tiêu nêu trên rất đa dạng tùy thuộc vào bài toán và được đo lường, tính toán thông qua lộ trình cụ thể của đường đi

VD

Để tính khoảng cách có thể tính tổng độ dài của các đoạn đường trong lộ trình đó dựa vào tham số trên bản đồ; Với bài toán này, ta phải xác định khoảng cách trước khi đi, tức là tại điểm A chứ không phải điểm B.

Đối với yếu tố thời gian và yếu tố chi phí thì thông thường được xác định dựa vào lịch sử sử dụng phần mềm AI này nhờ vào khả năng “tự học” của nó.

->Ngoài các tiêu chí nêu trên, còn có thể có nhiều tiêu chí khác như: đường đi có nhiều cây xanh, phong cảnh đẹp, đường đi ít “ổ gà”, …

-.> Mỗi một đối số (tham số) nêu trên đều có 1 mức độ quan trọng khác nhau tùy thuộc vào từng tình huống cụ thể của bài toán và mỗi mức độ quan trọng này được đánh giá bởi một hệ số kèm theo và ta thường gọi là “trọng số”.

-> Hàm mục tiêu cũng rất đa dạng tùy vào từng trường hợp cụ thể và phổ biến nhất là hàm mục tiêu tuyến tính, và khi đó hàm mục tiêu tuyến tính của bài toán nêu trên có thể biểu diễn như sau:

f= a1\*x + a2\*y + a3\*z

x,y,z là các tham số như trên

a1, a2, a3 : là các trọng số

-> Ngoài hàm dạng mục tiêu tuyến tính như trên, ta còn có dạng hàm mục tiêu phi tuyến, đặc biệt là hàm mục tiêu bậc 2 với hệ số a âm dạng parabol. Khi đó, trạng thái mục tiêu chính là trạng thái đỉnh của parabol

(Hình parabol)

-> Các dạng hàm mục tiêu nêu trên là 1 trong những dạng tiêu chí cơ bản để xác định trạng thái mục tiêu của bài toán AI

### Tìm kiếm lời giải (searching) cho AI Problem

* + - 1. **Một số dạng bài toán (vấn đề) AI**

Có 4 dạng bài toán AI cơ bản

- Bài toán trạng thái đơn giản

- Bài toán đa trạng thái

- Bài toán ngẫu nhiên

- Bài toán thăm dò: “thử - sai”

* + - 1. **Các cấp độ tìm kiếm lời giải cho bài toán (vấn đề) AI**

Việc tìm kiếm lời giả cho bài toán AI có thể xác định bởi 3 cấp độ:

- Tìm ra lời giải

- Tìm ra lời giải tốt nhất trong số các lời giải đã tìm ra

- Chi phí, công sức tìm lời giải tốt nhất là thấp nhất

## Biểu diễn một số bài toán AI

### Các bài toán Trò chơi

VD1: Trò chơi m quân cờ (Cờ ta canh) = tictac-toe(có nhiều dạng)/n cột\_dòng

+ Trò chơi có m = n2 – 1 quân cờ; VD n = 3

Ban đầu: Đích:

1 4 3 1 2 3

7 6 8 4

5 8 2 7 6 5

- Gồm một bảng kích thước n = 3 (3x3) với n2-1 = 8 quân cờ được đánh số từ 1 đến 8 và 1 ô trống.

- Một quân cờ đứng cạnh ô trống có thể tiến vào ô trống

- Mục tiêu là đưa bàn cờ từ trạng thái bên trái sang trạng thái bên phải (Các quân cờ được sắp xếp tăng dần theo chiều kim đồng hồ)

VD2: Game Caro:

- Cho một bàn cờ kích cỡ m x n, mục tiêu là người chơi phải tạo được một hàng dọc, ngang hoặc chéo gồm 5 quân cờ của mình.

### Cây không gian trạng thái của Bài toán AI

* + - 1. **Định nghĩa**

- Định nghĩa: không gian trạng thái của bài toán AI: là tập các trạng thái (=nút trên cây)

* + - 1. **Phương pháp giải quyết bài toán AI**
         1. **Phương pháp gia tăng**

Đặt từng đối tượng vào sao cho vẫn đảm bảo các yêu cầu của bài toán

* + - * 1. **Phương pháp trạng thái**

Đặt tất cả các đối tượng vào một lần, sau đó tìm cách để đảm bảo yêu cầu của bài toán

* + - 1. **Sự bùng nổ không gian trạng thái**

## Các phương pháp biểu diễn tri thức

### Phân loại tri thức theo phương pháp biểu diễn

* + - 1. **Tri thức sự kiện**

Tri thức sự kiện là 1 khẳng định về một sự kiện, hiện tượng hay một khái niệm nào đó trong một hoàn cảnh không gian hoặc thời gian nhất định.

VD: Khẳng định về hiện tượng: “Mặt trời phát sáng”.

Khái niệm về: “hình vuông: là hình tứ giác có 1 góc vuông và có 4 cạnh bằng nhau”.

* + - 1. **Tri thức mô tả**

Tri thức mô tả đối tượng, hiện tượng, vấn đề trong tự nhiên

VD: ngoại hình của một người có thể dùng để nhận diện một người một cách tương đối.

* + - 1. **Tri thức thủ tục**

Là tri thức biểu diễn các giải quyết một vấn đề, quy trình xử lý các công việc, lịch trình tiến hành các thao tác…

VD: Các bước để giải quyết bài toán phương trình bậc nhất ax + b = 0

+ Nhập vào các hệ số a và b

+ Nếu a==0:

- Nếu b==0 => Vô số nghiệm (hay vô định)

- Nếu b!=0 => Vô nghiệm

+ Nếu a!=0:

Phương trình có nghiệm là x = -b/a

* + - 1. **Tri thức mêta**

Là 1 dạng “tri thức thứ cấp” = tri thức tham chiếu đến các tri thức khác (dựa trên tri thức sơ cấp)

VD: SOP = tri thức thứ cấp = vị từ cấp 2 ; FOP = tri thức sơ cấp = vị từ cấp 1

VD: Phần mềm hỗ trợ xây dựng phần mềm mới

* + - 1. **Tri thức có cấu trúc**

Tri thức được hình thành dựa trên cơ sở định nghĩa cấu trúc Agent (Token) một cách rõ ràng bao gồn những thành phần gì = mỗi một phần tử tri thức mới hình thành phải tuân thủ cấu trúc đã định nghĩa (tương tự Cấu trúc dữ liệu).

### Phân loại tri thức theo nền tảng hình thành Cơ sở tri thức

* + - 1. **Tri thức hệ Heuristic**

Là tri thức dựa vào các quy luật Xác suất thống kê = quy luật số đông (số lớn)

[2 yêu cầu cơ bản: đủ lớn + đủ phổ biến/dàn trải]

Tri thức heuristic thường được coi là một “mẹo”(tip) nhằm dẫn dắt tiến trình lập luận. Vì vậy có nhiều rủi ro, đơn giản, …=> có tỷ lể sai (phải dùng Ràng buộc đảm bảo các suy luận của hệ thống AI có giá trị sử dụng => VD sử dụng thư viện Python-constraint).

* + - 1. **Tri thức hệ chuyên gia (ES = Expert System)**

**Chuyên gia (Expert)** là tri thức AI được xác định và hình thành dựa trên cơ sở chuyên gia:

+ người có đầy đủ kỹ năng, kiến thức sâu (cả về luật và các sự kiện) về một lĩnh vực nào đó

+ người có thể làm những việc mà người khác ít khả năng làm được

VD: Bác sĩ giỏi = có chuyên môn + kinh nghiệm

**Hệ chuyên gia** là chương trình máy tính có thể thực hiện các công việc, vấn đề trong thuộc lĩnh vực hẹp ở mức tương tự như một chuyên gia.

+ Hầu hết các hệ chuyên gia là các hệ dựa “luật”.

+ Hiện nay một số các hệ chuyên gia thành công trong các lĩnh vực: bán hàng, kỹ nghệ, y học và khai khoáng (tìm kiếm mỏ; GIS), các hệ điện lực.

* + - 1. **So sánh Tri thức hệ Heuristic với tri thức hệ chuyên gia (ES)**

Chú ý: Phân biệt {tương đối VD đông y dược học trong điều trị y khoa (ES, H)}

So sánh tri thức heuristic <-> ES = Expert System

\*Giống: Cơ sở tri thức

\*Khác:

|  |  |
| --- | --- |
| Heuristic | Expert System |
| Số đông | Kinh nghiệm |
| Nhiều đối tượng [đủ lớn + đủ phổ biến] | 1 số ít chuyên gia |
| Phổ thông | Gắn chuyên môn |
|  |  |

\*Ứng dụng: Tùy trường hợp có thể ứng dụng cả 2

Nhận xét: Khi cần phát triển các phần mềm hoặc hệ thống

- phần mềm liên quan đến yếu tố chuyên môn, đặc biệt chuyên môn dựa trên trình độ cao: ES

- phần mềm dựa trên cơ sở nhận xét + đánh giá -> rút kinh nghiệm, thống kê, tổng hợp: Heuristic

VD: Một số giải thuật tìm đường đi ngắn nhất, giải thuật A\* có thể được coi là lời giải của một vấn đề tốt nhưng chưa hẳn tối ưu.

### Logic mệnh đề và Logic vị từ

Logic mệnh đề và logic vị từ là cơ sơ xây dựng các cơ chế suy luận logic/ cơ sở tri thức

* + - 1. **Logic Mệnh đề**

Mệnh đề = là một khẳng định có thể nhận giá trị Đúng | Sai (chân trị)

VD:

“Hà Nội của thủ đô của Việt Nam”

“Mặt trời mọc ở hướng đông”

“hai nhân hai bằng bốn”

…..

+ Đa số các mệnh đề đúng | sai tùy thuộc vào nhiều yếu tố ngoại cảnh như:

Theo thời điểm

Chủ đề liên quan

+ Các bài toán AI chỉ xem xét mệnh đề (chân trị mệnh đề) một cách tương đối

VD: Dựa vào số đông cho là đúng

CÁC KÝ HIỆU

- Mệnh đề: a,b,p,q,…

- Các phép kết nối mệnh đề:

+ Hội (và): ^

+ Tuyển (hoặc): v

+ phủ định(đảo):

+ kéo theo(suy ra): a -> b hoặc -a v b

- Luật suy diễn:

A: Đúng và A->B đúng => B đúng

A->B Đúng và B sai => A sai

- Tất cả các sự kiện trong cơ sở tri thức (CSTT) có thể biểu diễn bởi một số hữu hạn các mệnh đề và các phép nối trên

MỘT SỐ CÁC KÝ HIỆU KHÁC

+ Các biến mệnh đề

+ Các hàm mệnh đề

* + - 1. **Logic vị từ** 
         1. **Vị từ và lượng từ**

Logic vị từ = là cách mở rộng logic mệnh đề bằng cách bổ sung biểu diễn vị từ

\* Một số lượng từ

+ phổ dụng (∀)

+ tồn tại (∃) hoặc tồn tại duy nhất (∃!)

\* Vị từ là dạng hàm mà các biến || tham số của nó là các biến mệnh đề

Kết luận:

+ Logic vị từ có thể biểu diễn gần như toàn bộ các khái niệm và nguyên lý của các khoa học cơ bản

+ Logic mệnh đề và logic vị từ là cơ sở xây dựng các luật dẫn xuất nhằm giúp cho việc thực hiện tìm kiếm lời giải bài toán AI thông qua khai thác CSTT

* + - * 1. **Biểu diễn Logic vị từ và chân trị của biểu diễn tri thức vị từ**

VD1: vị từ p(x): số chẵn; với x = “Số Nguyên”

Tùy vào x là số nào mà p(x) có thể đúng/sai

VD2: biểu diễn phát biểu: “Chẳng có vật nào lớn nhất và chẳng có vật nào nhỏ nhất”

Biểu diễn: Quy ước kí hiệu như sau

p(y, x): “y lớn hơn x”

q(z, x): “ z bé hơn x”

Câu (pb) trên:

(Ɐx ⱻy : p (y,x)) /\ (Ɐz ⱻy : q(z, x))

##### Mở rộng Logic vị từ

VD: P(x,y) 🡪 Q(P1(x), P2(y)) : vị từ bậc/ cấp 2

P1(x) và P2(y): vị từ bậc/ cấp 1

+ Vị từ cấp 1 (FOP: First Order Predicate)

+ Vị từ cấp 2 (SOP: Second Order Predicate)

VD1:

P1 = kết hôn => FOP x = nam? y = nữ? P1(x,y)

P2 = bà con (Cùng huyết thống) => FOP x,y = người? P2(x,y)

Q = phạm luật => SOP Q(P1(x, y), P2(x, y))

VD2:

P1 = kết hôn x = nam? y = nữ? P1(x,y)

P2 = đã có/lập gia đình x = người?

Q = phạm luật => SOP Q(P1(x,y), P2(x))

**Chú ý**: Với các tri thức dạng “luật” (Rule) có thể biểu diễn toàn bộ dưới dạng **FOP** (vị từ cấp 1 =Lượng từ, Vị từ, Phép nối) HOẶC cũng có thể biểu diễn dạng SOP (vị từ cấp 2) hoặc cấp cao hơn

Với các tri thức dạng “luật” (Rule) có thể biểu diễn FOP với nhiều kết quả biểu diễn khác nhau. XOR(x,y) <> XOR(y,x) OR(x,y) = OR(y,x)

##### Logic vị từ bậc||cấp cao

Chú ý:

+ Với các tri thức dạng “luật” (Rule) có thể biểu diễn toàn bộ Rule đó dưới dạng FOP(Vị từ cấp 1 = lượng từ [∀,∃,∃!], vị từ[MAN(…), PASS(…)], phép nối [∧,∨,¬]) hoặc cũng có thể biểu diễn dạng SOP(vị từ cấp 2) hoặc cao hơn.

+ Với các tri thức dạng “luật” (Rule) có thể biểu diễn FOP với nhiều kết quả khác nhau.

##### Logic tình huống

Logic vị từ có bổ sung thêm các biến tình huống

“Tàu XYZ đến cảng lúc 11g”

“Bốc dỡ hàng kết thúc lúc 18g”

“Anh A chỉ huy bốc dỡ hàng”

“Tàu XYZ rời cảng lúc 19g”

Hỏi: “17g A đang ở đâu?” => Trả lời: A ở bến cảng

Biến tình huống = biến thời gian = lượng hóa = biểu diễn định lượng

**KL: Cài đặt Logic vị từ dạng “tình huống” thường sử dụng “biến tình huống” tương tự như trên**

##### Logic trạng thái (modal)

Liên quan đến Fuzzy logic (Logic : “mờ”)

VD: Wiki: “Anh ABC ở phòng khách và phòng bếp ” = “Mờ”

Với phát biểu trên có 3 trạng thái xảy ra: [Giả định 2 phòng trên là 2 phòng riêng biệt]

- Anh ABC ở trong phòng khách (không liên quan phòng Bếp)

- Anh ABC ở trong phòng bếp (không liên quan phòng Khách)

- Anh ABC ở giữa cửa nối 2 phòng bếp và phòng khách = Logic trạng thái

=> “Mờ” nêu trên là chấp nhận được = cài đặt được vào AI Sys dưới dạng Logic trạng thái.

->Suy luận(lập luận) Logic cho bài toán AI

##### Logic xác suất và Logic khả xuất

Liên quan đến Fuzzy logic (Logic “mờ”)

VD: Tiếp VD trên 2.4.3.3.3.

- Anh ABC ở giữa cửa nối 2 phòng bếp và phòng khách = Logic trạng thái

Trên thực tế: ABC sẽ có 30% cơ thể ở bên phòng khách

ABC sẽ có 40% cơ thể ở bên phòng bếp

ABC sẽ có phần còn lại (30%) cơ thể ở giữa cửa nối 2 phòng

= Logic xác suất(ướm tương đối -> thực tế KHÔNG đo lường)

= Logic diễn đạt một sự kiện, có kèm theo xác suất xảy ra là ?%

**Logic khả xuất** = là trường hợp cá biệt(đặc biệt) của Logic xác suất = thay vì “xác xuất” thông thường thì xét “xác suất xảy ra” = ?% xảy ra sự kiện đó. Thường biểu diễn

VD p 🡪 0.8 q (độ tin cậy 0.8): 80% xảy ra

Tức là, khi có sự kiện p thì 80% xảy ra sự kiện q

VD cụ thể: “Trời nhiều mây thì 80% chiều nay mưa” = biểu diễn = Logic vị từ

“Mờ” = không rõ “nhiều mây” là như thế nào

=> minh họa 1 cách giải quyết: chụp ảnh bầu trời => phân tích và xử lý ảnh

=> xác định tỉ lệ % mây trên bầu trời/ảnh (biểu diễn logic vị từ: dạng logic xác suất)

=> Xác định “ngưỡng” (dưới = tối thiểu) [%mây] = “nhiều mây”

### Biểu diễn tri thức bằng Frame (Cấu trúc = "Khung")

- Frame là phương pháp biểu diễn tri thức có cấu trúc dữ liệu, chứa tất cả tri thức liên quan đến một đối tượng cụ thể nào đó = hình thức biểu diễn tri thức dựa trên cơ sở lập trình truyền thống (dùng CTDL = định nghĩa Agent = Token)

- Frame tương tự KDL “đối tượng” [Class] trong lập trình hướng đối tượng

- Frame thường được dùng để biểu diễn những

+ tri thức “chuẩn” hoặc

+ những tri thức dựa trên kinh nghiệm hoặc

+ các tri thức đã được hiểu biết cặn kẽ

### Suy diễn tri thức bằng luật dẫn xuất

#### Khái niệm

\* Suy diễn = lập luận =

Suy luận: là dựa trên: CSTT ban đầu (đã có || đã biết ) = tri thức cơ sở + Luật (Rules) dẫn xuất (Tập luật = nhiều luật)

=> Xác định (Suy diễn ra || lập luận để xác định ra) các “sự kiện mới” (bổ sung vào CSTT)

+ Sự kiện mới = là dạng logic “vị từ” [chân trị] = tri thức “dẫn xuất” || “suy luận” || “thứ cấp”

\* Nền tảng luật [quy tắc] suy diễn:

+ suy diễn tiến = cho các tiền đề -xác định-> kết qua có được từ các tiền đề đó.

+ suy diễn lùi|| ngược = cho [đặt ra] kết quả cần đạt -xác định-> các tiền đề cần thiết để có được kết quả đó

#### Luật dẫn xuất trong Cơ sở tri thức: Suy diễn tiến

\* Là các quy tắc || quy luật làm cơ sở để “suy diễn tri thức” thường là dạng biểu diễn hình thức hóa tri thức tự nhiên = các ký hiệu (toán) / lưu trữ trên máy tính phục vụ hệ thống phần mềm AI

\* Phổ biến là luật dẫn xuất (kéo theo) dạng tam đoạn luận (logic học)

Cho 1 CSTT biểu diễn (giả thuyết|| tiền đề) => Kết luận

\* CSTT dùng luật dẫn xuất để suy luận ra các KL bổ sung các sự kiện vào tập sự kiện biểu diễn (tri thức)

VD:

+ Sự kiện: Q, S

+ Tập dẫn xuất:

Text

Description automatically generated

Yêu cầu tìm tất cả các sự kiện có thể có được từ {Q,S} = “bao đóng” của tập {Q,S}, ký hiệu {Q,S}\* hoặc nếu kí hiệu P = {Q,S} thì P\* là bao đóng của {Q,S}

R3: S 🡪 P {P,Q,S}

R1: P 🡪 V {P,Q,S,V}

R4: V ∧ Q 🡪 Y {P,Q,S,V,Y}

R2: Y 🡪 X {P,Q,S,V,X,Y}

R5: X ∧ V ∧ Q 🡪 U {P,Q,S,U,V,X,Y}

Kết quả : {P,Q,S,U,V,X,Y} = {Q,S}\*

### Biểu diễn suy luận bằng đồ thị AND/OR

\* Dùng để suy diễn lùi || ngược

\* Là đồ thị có hướng AND/OR, trong đó

Nút = sự kiện

Cung/cạnh: suy luận 1 chiều từ sự kiện này -> sự kiện khác (tiến)

Cung/cạnh từ 1 nút: biểu diễn cho các quan hệ AND/OR -> dẫn xuất điều kiện

Gốc = sự kiện mục tiêu

Mục đích: tìm tất cả sự kiện dẫn xuất từ sự kiện mục tiêu

VD: Có 2 luận đề cho kết quả Z: K ∧ H -> Z và Y -> Z

Diagram

Description automatically generated

### Các biểu diễn tương đương trong Logic mệnh đề và Logic vị từ

CHÚ Ý & KÝ HIỆU:

Biểu thức vị từ (có chân trị) = {phép nối} & {lượng từ[Mọi, tồn tại,tồn tại duy nhất]} & {vị từ[hàm](biến mệnh đề)} = MR MĐ

Ký hiệu … (vị từ || mệnh đề ) 1 … <=> .. (vị từ || mệnh đề) 2…

∴ ⬄

* + - 1. **Biến đổi tương đương**
* Các tương đương được sử dụng thường xuyên trong quá trình biến đổi một biểu thức vị từ từ dạng này sang dạng khác
* Khả năng biến đổi tương đương trên máy tính có thể được làm tự động

// CƠ CHẾ SUY DIỄN

* **Các tương đương:**

Trong các tương đương sau A,B,C là các mệnh đề (có thể dạng vị từ) bất kỳ.

+ Dạng phủ định kép

¬¬ A <=> A

**CHÚ Ý: Các phép toán:**

¬ ~ Phủ định một ngôi (KHÔNG)

∧ Hội hai ngôi (VÀ)

∨ Tuyển hai ngôi (HOẶC)

⇒ Suy ra hai ngôi (cũng là: 🡪)

<=> = Tương đương hai ngôi

\* Dạng tuyển (HOẶC)

A ∨ TRUE ⬄ TRUE

A ∨ FALSE ⬄ A

A ∨ A ⬄ A

A ∨ ¬ A ⬄ TRUE

\* Dạng hội (VÀ)

A ∧ TRUE ⬄ A

A ∧ FALSE ⬄ FALSE

A ∧ A ⬄ A

A ∧ ¬ A ⬄ FALSE

\* Dạng suy ra (dẫn xuất)

A ⇒ TRUE ⬄ TRUE chú ý: A ⇒ TRUE ⬄ ¬ A ∨ TRUE

A ⇒ FALSE ⬄ ¬ A chú ý: A ⇒ FALSE ⬄ ¬ A ∨ FALSE

TRUE ⇒ A ⬄ A chú ý: TRUE ⇒ A ⬄ ¬ TRUE ∨ A

FALSE ⇒ A ⬄ TRUE chú ý: FALSE ⇒ A ⬄ ¬ FALSE ∨ A

A ⇒ A ⬄ TRUE chú ý: A ⇒ A ⬄ ¬ A ∨ A

\* Dạng hấp thu

A ∧ (A ∨ B ) ⬄ A A “hấp thụ” (…): mất tác dụng -> nếu có A bên trong & đảo phép nối

A ∨ (A ∧ B ) ⬄ A

A ∧ (¬ A ∨ B ) ⬄ A ∧ B Bỏ ~ A trong (…) = B “hấp thụ” A -> nếu có ~A bên trong & đảo phép nối

A ∨ (¬ A ∧ B ) ⬄ A ∨ B

\* Dạng De Morgan

¬ (A ∧ B) ⬄ ¬ A ∨ ¬ B

¬ (A ∨ B) ⬄ ¬ A ∧ ¬ B

A => B ⬄ ¬ A ∨ B

¬ (A => B) ⬄ A ∨ ¬ B

¬ (A => A) ⬄ A ∨ ¬ A => FALSE

\* Phép ∧ và ∨ có khả năng kết hợp

A ∧ (B ∧ C) ⬄ (A ∧ B) ∧ C (A ∨ B) ∨ C ⬄ A ∨ (B ∨ C)

\* Phép ∧ và ∨ có khả năng hoán vị (giao hoán)

A ∧ B ⬄ B ∧ A A ∨ B ⬄ B ∨ A

\* Phép ∧ có khả năng phân phối trên ∨

A ∨ (B ∧ C) ⬄ (A ∨ B) ∧ (A ∨ C)

\* Phép ∨ có khả năng phân phối trên ∧

A ∧ (B ∨ C) ⬄ (A ∧ B) ∨ (A ∧ C)

* + - 1. **Các dạng chuẩn CNF & DNF**

Còn gọi là “Dạng chuẩn tắc”

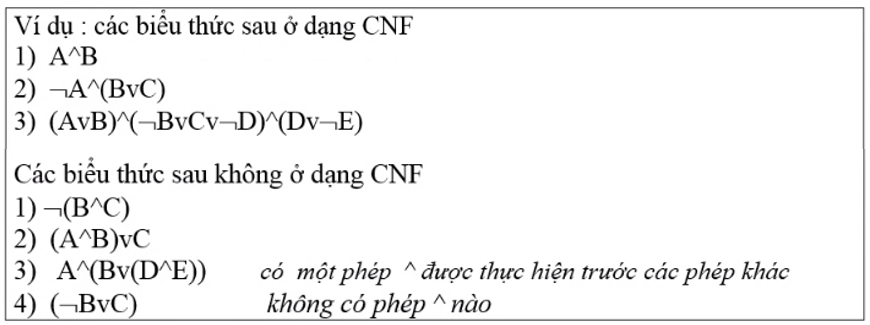
* **Dạng chuẩn**

Là kết xuất || dẫn xuất chuẩn của các giải thuật với | trên phép toán mệnh đề (vị từ)

* Dạng chuẩn hội (VÀ) – CNF(Conjunctive Normal Form)

Là thành phần **hội**(VÀ) **cơ bản** [hay các **hội cơ bản**] kết hợp với phép **TUYỂN (HOẶC)**

CHÚ Ý: Tất cả các phép hội (và = ∧) thực hiện **sau cùng**

****

* **Dạng chuẩn tuyển (HOẶC) – DNF (**Disjunctive Normal Form**)**

Là thành phần **tuyển**(HOẶC) **cơ bản** [hay các **tuyển cơ bản**] kết hợp với phép **HỘI (VÀ)**

CHÚ Ý: Tất cả các phép tuyển (hoặc = ∨) thực hiện **sau cùng**

**Text

Description automatically generated with medium confidence**

**Kết luận: Nếu một lập luận (suy luận) của phần mềm AI xuất phát từ CNF hoặc DNF thường cho kết quả chắc chắn = độ chính xác cao**

* + - 1. **Suy diễn chứng minh**

\* Luật suy diễn được áp dụng để phát triển các ứng dụng có khả năng suy luậ. Suy luận là hoạt động thường xuyên của con người để hiểu các lý lẽ, kiểm chứng, phán đoán các vấn đề.

\* Ứng dụng trong nhiều bài toán AI dạng:

+ “Tư vấn” như: Tư vấn pháp luật,…

+ “Phán đoán” như: thời tiết, chuẩn đoán y khoa, tử vi(game), game,…

+ “Chăm sóc khách hàng” như: tổng đài trả tự động trả lời câu hỏi, Robot, …

MỘT SỐ LUẬT THƯỜNG DÙNG (+ CÁC LUẬT CƠ BẢN NÊU TRÊN)

* Luật Modus Ponens(MP)

A, A => B ⬄ B

“Nếu A đúng và A => B đúng thì B đúng”

* Luật Modus Tollens(MT)

A => B, ¬ B ⬄ ¬ A

“Nếu A =>B đúng và B sai thì A sai”

* Luật Hội || Giao

A, B ⬄ A ∧ B

“Nếu A đúng và B đúng thì A ∧ B đúng”

* Luật đơn giản

A ∧ B ⬄ A, B

“Nếu A ∧ B đúng thì A đúng và tất nhiên B cũng đúng”

* Luật Cộng || Tuyển || Hợp

A ⬄ A ∨ B

“Nếu A đúng thì A ∨ B đúng”

* Luật tam đoạn luận tuyển (phản chứng)

A ∨ B, ¬ A ⬄ B

“Nếu A ∨ B đúng và A sai thì B đúng ”

* Luật tam đoạn luận giả thuyết (kéo theo = bắc cầu)

A => B, B => C ⬄ A => C

“Nếu A => B đúng và B => C cũng đúng thì A => C đúng”

KL:

* Thường dùng luật dẫn xuất nêu trên để biểu diễn các tri thức dạng “ký tự”||Text dạng “định tính” vào CSTT để lưu trữ trên máy tính =phục vụ => cho các phần mềm AI lập luận và giải quyết các vấn đề trong thế giới tự nhiên
* Khi PM AI nhận (Input) các dữ liệu đầu vào -truy vấn tri thức theo phương pháp sao trùng mẫu (Pattern) 🡪tìm ra các Pattern có độ sao trùng gần nhất (tính sai số min) các tiền đề 🡪 nhằm -AI-> đề xuất (đưa ra) các KL tương ứng

== Đòi hỏi phải có phương pháp biểu diễn các tri thức dạng text

=>Sử dụng logic Vị từ(Mệnh đề) + luật dẫn xuất

**AI dạng: Association Rules = Luật kết hợp**

= AI dựa trên cơ sở kết hợp các luật trong tập luật suy diễn (dẫn xuất) nhằm giải quyết các tình huống thực tiễn + tìm ra các quy luật (Luật suy diễn) mới bổ sung vào tập luật suy diễn (KB)

### Xử lý ngôn ngữ tự nhiên (Natural Language Processing = NLP): Voice Assistant

## Các phương pháp tìm kiếm lời giải của bài toán AI

### Giới thiệu

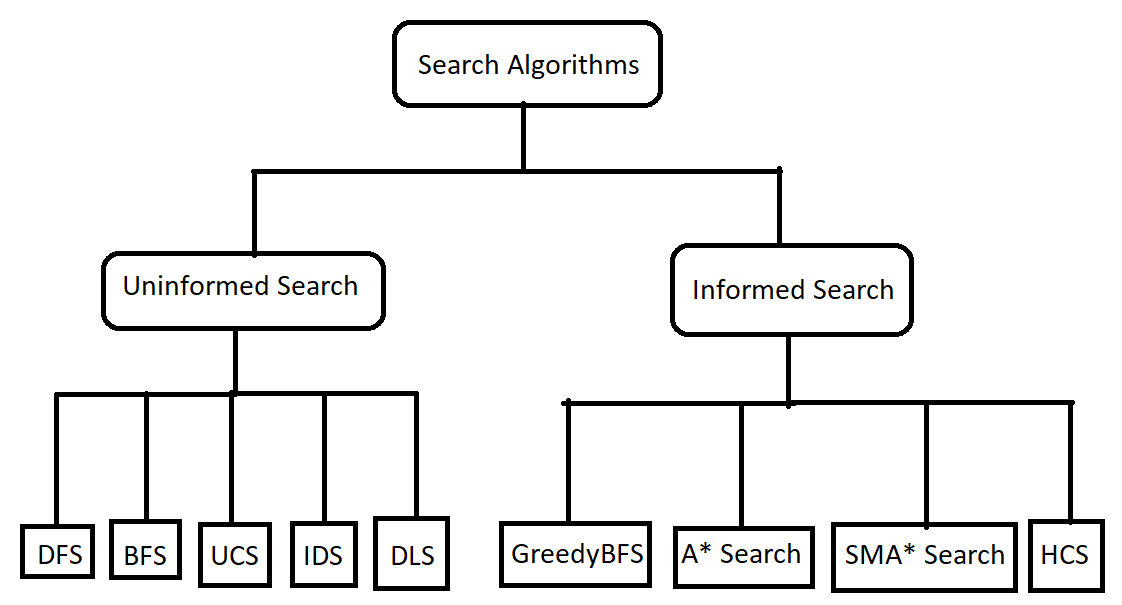
Còn gọi là "Các chiến lược tìm kiếm lời giải " bài toán AI

* + - 1. **Các tiêu chí đánh giá**
* Tính hoàn thành: chiến lược có bảo đảm tìm thấy giải pháp cho bài toán ?
* Độ phức tạp thời gian: chiến lược mất bao lâu để tìm ra một giải pháp?
* Độ phức tạp không gian (dung lượng bộ nhớ): chiến lược đó cần bao nhiêu dung lượng bộ nhớ để thực hiện việc tìm kiếm.
* Tính tối ưu: chiến lựơc có tìm được giải pháp có chất lượng cao nhất khi có một số các  giải pháp khác nhau?
  + - 1. **Phân loại các phương pháp (chiến lược) tìm kiếm lời giải**

Có nhiều cách phân loại các chiến lược tìm kiếm, phổ biến là chia làm 2 loại:

+ Các phương pháp tìm kiếm mù (Blind Search) = tìm kiếm không có thêm thông tin  (uninformed search)

+ Các phương pháp tìm kiếm theo kinh nghiệm (Heuristic Search) = tìm kiếm với thông tin  bổ sung (Informed Search) = tìm kiếm rõ



Chú ý: **Phương pháp** tìm kiếm … = còn gọi là **Chiến lược** tìm kiếm ….

### Các phương pháp tìm kiếm mù (Blind Search) = tìm kiếm không có thêm thông tin (uninformed search)

* + - 1. **Depth-First Search (DFS): Tìm kiếm theo chiều sâu**

\* Là giải thuật để duyệt cấu trúc cây [đồ thị] không gian trạng thái để tìm lời giải của bài toán AI, trong  đó nó lần lượt tìm theo chiều sâu của từng nhánh cây [đồ thị] trước, cho đến khi KHÔNG tìm được lời  giải [đáp án] trên nhánh đó [còn gọi là “bế tắc” || “bí” || hết đường || cùng đường] thì nó quay lui  [back\_tracking] để tìm trên nhánh kế tiếp: cho đến khi:

+ Tìm được lời giải [đáp án] OR

+ Duyệt hết toàn bộ (OR giới hạn phạm vi) không gian cây [đồ thị] được xem xét mà vẫn không tim  được lời giải [đáp án]

Chart, radar chart

Description automatically generated

*Hình -1. Cây minh họa giải thuật DFS*

A picture containing telephone, electronics, calculator, pool ball

Description automatically generated

*Hình -2. Cây minh họa cho thủ tục DFS*

**\* Mô tả DFS:**

+ Tìm kiếm theo chiều sâu luôn luôn mở rộng một trong các nút ở mức sâu nhất của cây.

+ Chỉ khi phép tìm kiếm đi tới một "điểm cụt" : "nút lá" (một nút không phải đích mà không có nút con), việc  tìm kiếm sẽ quay lại và xét những nút nông/cạn hơn: BACK TRACKING

+ Giải pháp này có thể được thực hiện bởi 1 phép tìm kiếm với sự hỗ trợ of một stack: các trạng thái mới  được sinh ra đc push vào đỉnh stack (thông thường: phải -> trái)

**\* Nhận xét DFS:**

[1] Do nút được xét là sâu nhất, các nút kế tiếp của nó sẽ sâu hơn và khi đó sẽ trở thành sâu nhất. => Phép tìm kiếm theo chiều sâu yêu cầu dung lượng bộ nhớ rất thấp / nhanh / rủi ro

=> Tìm kiếm: nó chỉ cần phải lưu một đường duy nhất từ gốc tới nút lá, cùng với các nút anh em với các nút trên đường đi  chưa được mở rộng còn lại.

=> Đối với một không gian trạng thái với hệ số rẽ nhánh b và độ sâu tối đa m, phép tìm kiếm theo nhiều sâu chỉ yêu cầu lưu  trữ b\*m nút.

 Độ phức tạp thời gian của phép tìm kiếm sâu là O(b\*m).

=> Đối với những vấn đề (bài toán) mà có rất nhiều giải pháp (đáp số), phép tìm kiếm sâu có thể nhanh hơn tìm kiếm rộng, bởi vì nó có một  cơ hội tốt tìm ra một giải pháp chỉ sau khi khám phá một phần nhỏ của toàn bộ không gian.

=> Mặt hạn chế của phép tìm kiếm sâu là nó có thể bị tắc khi đi theo một đường sai: rủi ro lớn (bị “chìm”)

[2] Rất nhiều bài toán có các cây tìm kiếm rất sâu, thậm chí vô hạn, vì vậy tìm kiếm sâu có thể sẽ không quay lại  được (ko back track được) NẾU sự lựa chọn sai (chọn nhầm nhánh cây quá “sâu”) = Phép tìm kiếm sẽ luôn luôn  tiếp tục đi xuống mà không quay trở lại, thậm chí trong khi có một giải pháp ở mức rất nông tồn tại ở các nhánh  khác. Như vậy đối với những bài toán này, phép tìm kiếm sâu sẽ bị sa lầy trong một vòng lặp vô hạn hoặc không  bao giờ đưa ra một giải pháp, hoặc là cuối cùng nó có thể đưa ra một đường đi giải pháp ko tốt so với phương án  tối ưu = RỦI RO DFS rất cao.

* + - 1. **Breadth-First Search (BFS): Tìm kiếm theo chiều rộng || ngang**

\* Là giải thuật để duyệt cấu trúc cây [đồ thị] không gian trạng thái để tìm lời giải của bài toán AI, trong  đó nó lần lượt tìm theo chiều ngang [rộng] của từng mức [cấp] cây [đồ thị] trước, cho đến khi KHÔNG  tìm được lời giải [đáp án] trên mức [cấp] đó [còn gọi là “bế tắc” || “bí” || hết đường || cùng đường] thì  nó quay lui [back\_tracking] để tìm trên mức [cấp] kế tiếp: cho đến khi:

+ Tìm được lời giải [đáp án] OR

+ Duyệt hết toàn bộ (OR giới hạn phạm vi) không gian cây [đồ thị] được xem xét mà vẫn không tim  được lời giải [đáp án]

A picture containing electronics, calculator, black, dark

Description automatically generated

*Hình-3. Cây minh họa BFS*

**\* MÔ TẢ BFS**

Trong chiến lược BFS này,

+ nút gốc được xét trước tiên,

+ sau đó đến lượt tất cả các nút con trực tiếp của nút gốc: được xét

+ tiếp đến là những nút con kế tiếp của chúng và cứ lặp lại như vậy

=> cho đến khi tìm ra nút đích (hoặc hết phần không gian trạng thái mà ko tím thấy lời giải)

**\* Nhận xét BFS**

+ Tất cả các nút ở độ sâu d trên cây tìm kiếm được xét trước các nút ở độ sâu d+1.

+ Tìm kiếm theo chiều rộng có thể được thực hiện bằng cách gọi giải thuật general-search (dưới) với một hàng đợi (queue =  FIFO = First In First Out = vào trước ra trước)

+ Tìm kiếm theo chiều rộng là một chiến lược tìm kiếm lời giải có hệ thống bởi vì nó xem xét tất cả các đường đi có độ dài  bằng 1 trước, sau đó đến tất cả những đường đi có độ dài bằng 2, và lặp lại như vậy => nên :

. tìm kiếm theo chiều rộng sẽ luôn tìm ra trạng thái đích cạn/nông nhất trước tiên.

. tìm kiếm theo chiều rộng: trạng thái đích tim đc là trạng thái có chi phí tối ưu.

 (nếu chi phí = một hàm tăng của độ sâu các nút).

\* kết luận: BFS phù hợp các bài toán:

+ cây không gian trạng thái quá sâu; ko có nút "lá"

+ số lượng nút con của mỗi nút trên cây: có giới hạn

* + - 1. **Uniform Cost Search (UCS): Tìm kiếm với chi phí ĐỀU (không thay đổi) = chi phí cực tiểu [Áp dụng thuật toán Dijkstra]**
      2. **Interative Deepening Search (IDS) = Tìm kiếm sâu dần = cải tiến DFS**

d = 2 : 1\_2\_3\_4\_5\_7\_8\_9

d +=2 ⬄ d =4 : 1\_2\_3\_4\_5\_7\_8\_9 : 6 => tăng dần độ "sâu" từ 2 -> 4 -> …(Δd=2)  (Δd : tương đối nhỏ)

A picture containing electronics

Description automatically generated

* + - 1. **Depth-Limited Search (DLS) = Tìm kiếm giới hạn độ sâu = cải tiến DFS**

d = 2: const : 1\_2\_3\_4\_5\_7\_8\_9

 => KHÔNG tăng thêm "sâu" (thường d = const = tương đối lớn, hàng ngàn,…)

A picture containing electronics

Description automatically generated

### Các phương pháp tìm kiếm theo kinh nghiệm (Heuristic Search) = tìm kiếm với thông tin bổ sung (Informed Search) = Cải tiến BFS

= có tham số bổ sung,

 thông thường là trọng số (hệ số đánh gía mức độ quang trọng), hàm chi phí (Cost). .

* + - 1. **Best first: BestBFS (Greedy\_BFS)**

A picture containing shape

Description automatically generated

BFS : Giả sử có hàm ước lượng chi phí để đi từ nút này đến nút khác : g  g(1, 2) = 5 g(1,3) =3 g(1,4) = 7

 g(2,5) = 9 g(3,6) =4 g(3,7) = 1 g(4,8) = 9

 g(6,9) = 2

BFS thông thường trên 1 layer nó duyệt từ L->R (trái -> phải): ko quan tâm hệ số (cơ bản)

G\_BestBFS = từ Layer đến layer kế tiếp : nó ưu tiên chọn duyệt theo thứ tự  các nhánh có chi phí thấp hơn là duyệt trước, cao hơn duyệt sau

**G\_BestBFS** 1\_3[3]\_2[5]\_4[7] || 7[1]\_6[4]\_5[9]\_8[9] || 9[2]

* + - 1. **A\* và SMA\* (A\* với bộ nhớ giới hạn): tìm kiếm trên đồi thị (Graph Search)**

A picture containing shape

Description automatically generated

Sử dụng 2 hàm & luôn tính lại tại mỗi nút duyệt (n):

+ Hàm đánh giá: f(n)= g(n) + h(n) => ước tính (chưa xác định chính xác) g(n) – Chi phí từ node hiện tại tới node n (tính hàm g)

*->mục tiêu: xác định từ node hiện tại sẽ duyệt đến Node nào trong số các node n tiếp theo  (node hiện tại <> node n)*

h(n) – ước lượng khoảng cách từ node n -> đích

+ Giả sử các hàm g và h tính số cung, mỗi *cung phải đi qua có chi phí = 1* và đích 9 . Xét 1 : xét xem duyệt 2 3 4 nút nào duyệt trước

n=2 g(2) = c(1, 2) = 1 h(2) = c(2,9) = ∞ f(2) = ∞

n=3 g(3) = c(1, 3) = 1 h(3) = c(3,9) = 2 f(3) = 3

n=4 g(4) = c(1, 4) = 1 h(4) = c(4,9) = ∞ f(4) = ∞

=> 1\_3 (bỏ 2 và 4) || …..

Kết quả : 1\_3\_6\_9

**Nhận xét** : Giải pháp rất hiệu quả, nhưng việc tính toán (ước tính) các hàm h và g quá tốn kém  và một số trường hợp không khả thi, đặc biệt là h.

* + - 1. **Hill-Climbing Search (Tìm kiếm tối ưu cục bộ || leo đồi|| lep núi) = HCS**

A picture containing shape

Description automatically generated

+ Từ Node hiện tại --tìm đỉnh tốt nhất lân cận-chuyển node hiện tại đến đó .CHỈ là tốt nhất trong lân cận, chưa chắt toàn cục => lời giải chưa chắc tốt nhất .Từ Node hiện tại (chưa tốt = thấp hơn) –"trèo lên" nút tốt hơn ("cao hơn") trong local. VD: 9 là đích: từ 1 chuyển đến đỉnh tiếp theo [có độ đo đến (ướm) gần đích {9}nhất] + Giả sử hàm h tính số cung, mỗi *cung phải đi qua có chi phí = 1* và đích 9 . Xét 1 : xét xem duyệt (lân cận) 2 3 4 nút nào là nút mà nó sẽ chuyển đến  h(2) = c(2,9) = ∞

 h(3) = c(3,9) = 2

 h(4) = c(4,9) = ∞

=> 1 -> 3 : Nút 3 trở thành nút hiện tại

. Xét 3: xét xem duyệt (lân cận) 6 7 nút nào là nút mà nó sẽ chuyển đến  h(6) = c(6,9) = 1

 h(7) = c(7,9) = ∞

=> 3 -> 6 : Nút 6 trở thành nút hiện tại

 . . .

Kết quả : 1\_3\_6\_9

**Nhận xét** :

+ Giải pháp phù hợp dạng KG TT dạng đồ thị (Graph)

+ Rất hiệu quả, nhưng việc tính toán (ước tính) các hàm h quá tốn kém và một số trường hợp  không khả thi

+ Giải pháp Kết quả chưa chắc tốt tốt nhất, vì chỉ tối ưu cục bộ ; KO Tối ưu toàn cục (global optimum)

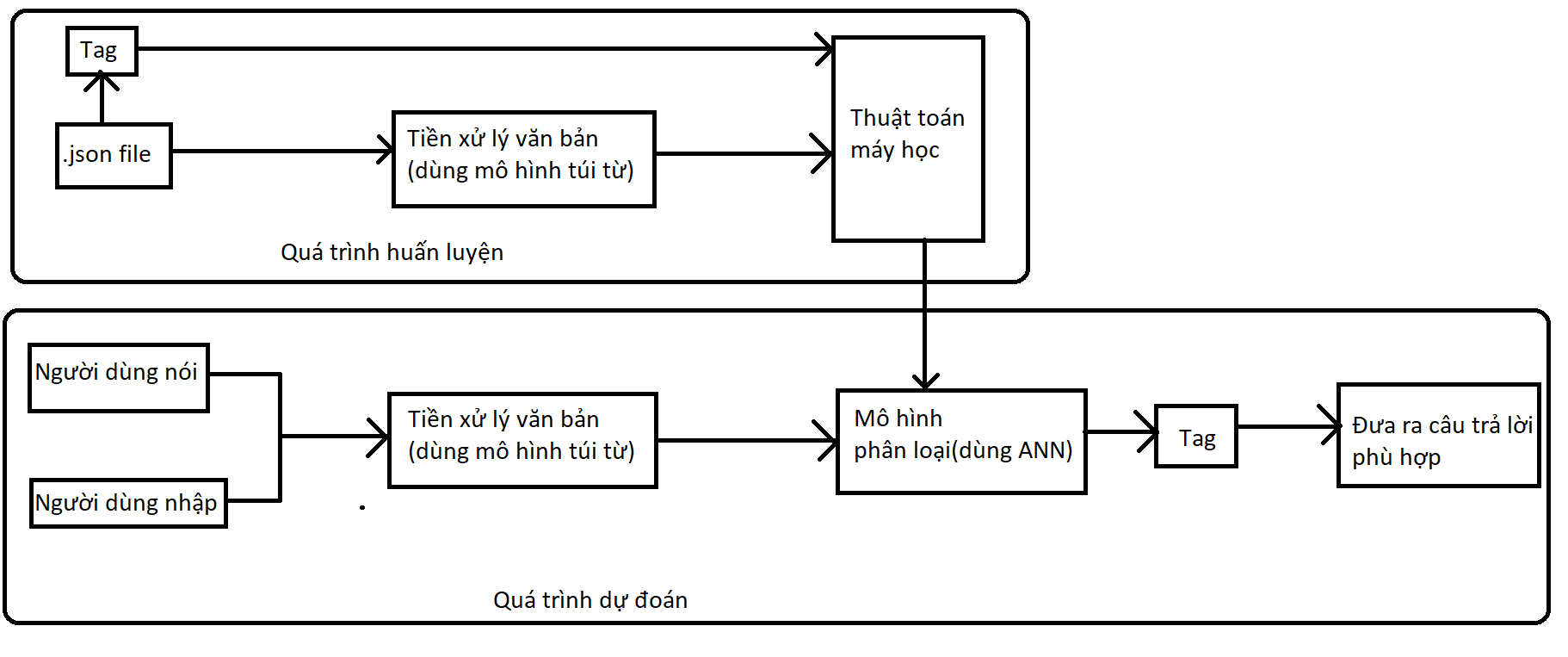
* + - 1. **Chiến lựơc tìm kiếm đối kháng (Board Game) với Giải thuật MiniMax||MinMax**

### Lập trình Logic (Logic Programming) phát triển cơ chế lập luận logic

# GIỚI THIỆU VỀ MÔ HÌNH BÀI TOÁN AI: NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO

## Mô tả bài toán AI

Chatbot sẽ đưa ra câu trả lời phù hợp cho nội dung mà người dùng nhập hoặc nói vào, có thể tiếp nối câu chuyện



## Các yếu tố xác định bài toán AI

Bài toán xây dựng hệ thống AI chatbot giao tiếp bằng tiếng Anh

được xác định bởi 4 yếu tố cơ bản sau đây:

+ Không gian trạng thái (tập trạng thái) của bài toán: Các câu trả lời có sẵn trong KB của AI

+ Trạng thái bắt đầu của bài toán: Câu trả lời phù hợp với nội dung mà người dùng nhập vào đầu tiên (thường là chào hỏi)

+ Mục tiêu của bài toán: Đưa ra câu trả lời tùy theo nội dung người dùng nhập vào

+ Một số ràng buộc của bài toán: Câu trả lời phải có nghĩa và phù hợp với nội dung người dùng nhập vào

## Giới thiệu nguồn dữ liệu gốc dùng thực nghiệm bài toán AI

Tập dữ liệu gốc chủ yếu dựa theo nội dung các file trong thư mục chatterbot-corpus/chatterbot\_corpus/data/english/ của link: <https://github.com/gunthercox/chatterbot-corpus> và tệp intents.json trong <https://bom.to/hRF5GBRrXKtYq> . Nội dung của file dữ liệu gốc còn khá là ít nhưng cũng có thể đáp ứng được việc chào hỏi bình thường. Trong file chứa một mảng Class, với mỗi phần từ của mảng có 4 key là: tag, patterns, responses, context. Hiện tại thì số phần tử của mảng là 9.

## Một số tiền xử lý dữ liệu nguồn dùng thực nghiệm bài toán AI

Đầu tiên ta sẽ chia câu ra thành các từ, sau đó đưa những từ đó và tag của nó vào class

Text

Description automatically generated

Tiếp theo, ta sẽ biến đổi các từ về dạng gốc của nó sau đó sẽ lưu lại để sử dụng, vì trong tiếng Anh có các từ với nghĩa gần giống nhau

Text

Description automatically generated

Kết quả ta được:

Text, letter

Description automatically generated

Tiếp theo, ta sẽ số hóa dữ liệu và tạo tập dữ liệu để huấn luyện mô hình ANN, ở đây ta sẽ xét các từ trong words ở trên, nếu từ nào có trong câu thì sẽ đánh số 1, nếu không có thì đánh số 0, tương tự với tag

Text

Description automatically generated

Kết quả ta tạo được được tập huấn luyện như sau:

Background pattern

Description automatically generated

A picture containing electronics, keyboard

Description automatically generated

# BIỂU DIỄN BÀI TOÁN AI VÀ CƠ SỞ TRI THỨC CỦA AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO

# Giới thiệu

+ Phương pháp giải quyết bài toán: là phương pháp gia tăng vì ta không biết là sau câu nói này thì người dùng sẽ nhập câu nói thuộc loại nào, nên không thể dùng phương pháp trạng thái để đưa sẵn các câu trả lời.

+ Biểu diễn cơ sở tri thức: bằng Frame , mỗi Frame là một loại câu nói (chào hỏi, cảm ơn,tạm biệt, … ) và cách trả lời của chúng. Vậy nên chúng ta chỉ cần phân loại xem câu nói người dùng nhập vào thuộc loại nào là có thể có ngay câu phản hồi.

## Biểu diễn bài toán AI

Bài toán AI ở đây là: người dùng sẽ nhập vào một câu nói, ta phải dùng các giải thuật để xác định xem câu nói đó thuộc loại nào (chào hỏi, cảm ơn,tạm biệt, … ) để đưa ra câu trả lời hợp lý. Phương pháp giải quyết bài toán là phương pháp gia tăng vì ta không biết là sau câu nói này thì người dùng sẽ nhập câu nói thuộc loại nào, nên không thể dùng phương pháp trạng thái để đưa sẵn các câu trả lời, vậy nên phải dùng phương pháp gia tăng để xác định từng câu nói.

## Xây dựng cơ sở tri thức (KBS) cho bài toán AI

Cơ sở tri thức là được biểu diễn bằng Frame được định nghĩa sẵn, mỗi Frame có:

- tag: Xác định xem Frame này là loại câu nói nào

- patterns: Một số mẫu câu nói thuộc loại này (Dựa vào đây để tạo mô hình ANN phân loại )

- responses: Một số mẫu câu phản hồi

- context: Xác định ngữ cảnh để tiếp tục câu chuyện

## Lập trình xử lý ngôn ngữ tự nhiên (Natural Language Processing = NLP): Voice Assistant (trợ lý ảo)

+ Thư viện sử dụng: Pyaudio, gTTS, playsound,datetime

Hàm để chuyển giọng nói thành văn bản:

Text

Description automatically generated

Hàm chuyển văn bản thành giọng nói:

Text

Description automatically generated

# PHƯƠNG PHÁP TÌM KIẾM LỜI GIẢI CHO BÀI TOÁN AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO

## Giới thiệu

Các phương pháp tìm kiếm lời giải cho bài toán AI

+ Tìm kiếm mù: DFS,BFS, IDS và DLS: Không phù hợp với bài toán này, vì đầu vào là một câu nói, đầu ra cũng là một câu nói để trả lời cho câu nói nhập vào nên mỗi nút của cây không gian trạng thái là một từ trong câu => Cây không gian trạng thái sẽ rất lớn nên DFS và BFS có thể sẽ hoạt động không hiệu quả trên cây này, IDS và DLS cũng không phù hợp vì là một câu nói nên chưa chắc là số từ của câu nói sẽ chia hết cho bước nhảy của thuật toán

+ Tìm kiếm Heuristic: GreedyBFS, A\*, leo đồi: Cũng không phù hợp vì mục tiêu của bài toán là mục tiêu “định tính” nên việc xác định được giá trị của hàm đánh giá tại mỗi nút là không khả thi

+ Mạng nơron nhân tạo: Có thể dùng các thuật toán máy học để xây dựng mạng nơron nhân tạo nhận diện câu được nhập vào thuộc loại nào, từ đó đưa ra câu phản hồi phù hợp. Nhưng phải có bộ dữ liệu đủ phong phú và đa dạng để huấn luyện mô hình

## Mô tả sơ bộ về mô hình cây không gian trạng thái của bài toán AI

Vì áp dụng mạng nơron nhân tạo nên cây không gian trạng thái có độ cao là 2, các nút ở mức 1 là các “tag” của Frame trong cơ sở tri thức, các lá của nó là các “responses” của “tag” đó.

## Xác định phương pháp tìm kiếm lời giải cho bài toán AInghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo

Phương pháp tìm kiếm lời giải cho bài toán AI của đề tài này là mạng nơron nhân tạo. Dựa vào mạng nơron nhân tạo, ta xác định được “tag” của câu mà người dùng nhập vào, sau đó chọn một lá ngẫu nhiên trên nhánh “tag” đó để làm câu trả lời.

## Cài đặt (lập trình) phương pháp tìm kiếm lới giải cho bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo

Dựa vào tập dữ liệu huấn luyện đã được tạo ở trên, ta tạo mạng nơron nhân tạo với lớp thứ nhất có 128 nơron, lớp thứ hai có 64 nơron và lớp cuối cùng có số nơron bằng với số “tag” của dữ liệu đầu vào, tức là số nút mức 1 của cây không gian trạng thái.

Text

Description automatically generated

Tiếp theo, ta huấn luyện mô hình 500 lần bằng thuật toán tối ưu Stochastic Gradient Descent (SGD). Sau đó ta lưu mô hình lại để sử dụng.

Text

Description automatically generated

# GIỚI THIỆU SẢN PHẨM CỦA ĐỀ TÀI AI NGHE VÀ PHẢN HỒI BẰNG GIỌNG NÓI DỰA VÀO MÔ HÌNH TÚI TỪ VÀ MẠNG NƠRON NHÂN TẠO

# Giới thiệu sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo

Thông tin sản phẩm

+ Tên gọi: Chatbot nghe và phản hồi bằng tiếng Anh

+ Công cụ & ngôn ngữ & phiên bản & thư viện: cài đặt Anaconda3 và Spyder4, cài đặt các thư viện : NLTK, Keras, json, pyaudio, numpy, tkinter, speechRecognition, playsound, gTTS

+ Nền tảng công nghệ: Mô hình túi từ và mạng nơron nhân tạo

Full codes:

Nội dung file dùng để tạo mô hình mạng nơron nhân tạo:

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated with medium confidence

A picture containing text

Description automatically generated

Nội dung file dùng để tạo giao diện người dùng và giao tiếp với người dùng:

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Kết quả thực nghiệm sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo

Graphical user interface, text, application

Description automatically generated

- Nút Send dùng để gửi tin nhắn trong ô nhập cho chatbot

- Nút Voice dùng để nhập văn bản bằng giọng nói, khi nhập xong thì nội dung sẽ hiện lên trong ô nhập

## Nhận xét và đánh giá về sản phẩm bài toán AI nghe và phản hồi bằng giọng nói dựa vào mô hình túi từ và mạng nơron nhân tạo.

- Ưu điểm: Sản phẩm đã thực hiện được yêu cầu đặt ra của bài toán là nghe và phản hồi bằng giọng nói

- Nhược điểm:

+ Cần bộ dữ liệu nguồn lớn và đa dạng thì mô hình mới có độ chính xác cao

+ Chưa liên kết được đoạn hội thoại

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## Kết luận

### Những kết quả đạt được

- Đã xây dựng được chatbot có thể nghe và phản hồi bằng giọng nói cho người dùng

### Hạn chế

- Chưa thể liên kết được câu chuyện

## Hướng phát triển

### Hướng khắc phục các hạn chế

Nghiên cứu cách sử dụng mục context trong Frame của Cơ sở tri thức

### Hướng mở rộng đề tài

- Thêm khuôn mặt cho chatbot để thể hiện cảm xúc

- Thêm ngữ điệu cho chatbot để thể hiện cảm xúc

- Thêm khả năng “tự học”

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Xuân Thể (2021), *Bài giảng học phần Trí tuệ nhân tạo*, TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM (lưu hành nội bộ).
2. Giang Trann(2020), *Xây Dựng Chatbot Bằng NLTK & Keras – Python Chatbot Project,* tek4.vn, <https://tek4.vn/xay-dung-chatbot-bang-nltk-keras-python-chatbot-project/> ,truy cập ngày 1/6/2021

# CÁC PHỤ LỤC

Nội dung file intents.json

Text

Description automatically generated with medium confidence

# ĐÁNH GIÁ KẾT QUẢ HỌC TẬP HỌC PHẦN

ĐỒ ÁN HP THAY CHO BÀI THI KẾT THÚC HỌC PHẦN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Hình thức KT** | **Nội dung** | **Chuẩn đầu ra đánh giá** | **Trình độ năng lực** | **Phương pháp đánh giá** | **Công cụ đánh giá** | **Tỉ lệ (%)** |
| **BÀI TẬP LẬP TRÌNH** | | | | | |  |
| BL#1 | Bài tập lập trình theo từng Buổi thực hành = tính bình quân  Kiểm tra viết | G1.1 | 3 | Bài tập  Kiểm tra | Bài tập  Kiểm tra | 50 |
| G2.1 | 3 |
| G3.1  G3.2 | 4  3 |
| G4.1 | 5 |
| **ĐỒ ÁN HỌC PHẦN (Project)** | | | | | |  |
| ĐA#1 | Phát triển một hệ thống AI đơn giản:  + Đề tài cá nhân  + Xây dựng và thực nghiệm giải thuật AI theo bài toán cá nhân  => báo cáo vào ngày thi. | G1.1 | 3 | Báo cáo, Demo và thuyết trình | Rubric | 50 |
| G2.1 | 3 |
| G3.1 | 4 |
| G3.2  G4.1 | 3  5 |
| G2.1 | 4 |
| G3.1 | 4 |
| G3.2  G4.1 | 3  5 |

1. Định lượng là phải xác định đối tượng đó dựa vào các giá trị số cụ thể, đo lường được, tính toán được. [↑](#footnote-ref-1)
2. Định tính là xác định dựa vào các tính chất, đặc trưng của đối tượng (không nhất thiết là giá trị số). [↑](#footnote-ref-2)