**cTRƯỜNG ĐẠI HỌC SÀI GÒN**

KHOA CÔNG NGHỆ THÔNG TIN



A blue circle with white text

Description automatically generated

**EXERCISE 1: TRẢ LỜI CÂU HỎI TRONG CHAPTER 1, 2, 3 (nhóm)**

HỌC PHẦN: TRÍ TUỆ NHÂN TẠO NÂNG CAO

**Giảng viên hướng dẫn** : Ts. Đỗ Như Tài

**Nhóm 11:**

Phạm Văn Nam 3122410251

Nguyễn Quan Tuấn Nghĩa 3122410260

Tạ Hồng Quí 3122410348

Vũ Quốc Vương 3120410629

*Thành phố Hồ Chí Minh - Tháng 09/2025*

Mục lục

[Bảng Phân Công 3](#_Toc209387406)

[Chapter 1: Introduction Discusion 3](#_Toc209387407)

[Discussion 1: The Goal of AI 3](#_Toc209387408)

[Discussion 2: AI Safety and Optimizers 4](#_Toc209387409)

[Discussion 3: Large Language Models (LLMs) 5](#_Toc209387410)

[Discussion 4: Turing Test: Large Language Models 6](#_Toc209387411)

[Discussion 5: The AI Effect: AI gets no respect? 7](#_Toc209387412)

[Discussion 6: AI Safety 10](#_Toc209387413)

[Discussion 7: Conclusion 14](#_Toc209387414)

[Chapter 2: Intelligent Agents Discusion 16](#_Toc209387415)

[Discussion 1: Self-driving Cars 16](#_Toc209387416)

[Discussion 2: PEAS Description of the Environment of a Self-Driving Car 17](#_Toc209387417)

[Discussion 3: Environment for a Self-Driving Car 18](#_Toc209387418)

[Discussion 4: State Representation: Self-Driving Car 19](#_Toc209387419)

[Discussion 5: What Type of Intelligent Agent is a Self-Driving Car? 23](#_Toc209387420)

[Chapter 3: Solving problems by searching 23](#_Toc209387421)

[Discussion 1: Heuristics from Relaxed Problems 23](#_Toc209387422)

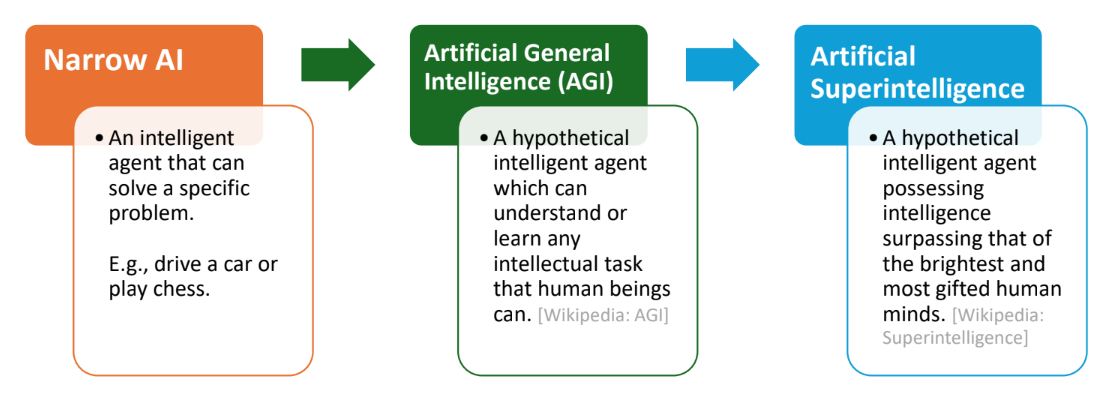
[Discussion 2: Case Study: Heuristic for Tic-Tac-Toe 24](#_Toc209387423)

# Bảng Phân Công

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Công việc** | **Người thực hiện** | **Trạng thái** |
| 1 | Discussion 1, 2, 3, 4 (Chapter 1) | Văn Nam | Hoàn thành |
| 2 | Discussion 5, 6, 7 (Chapter 1) | Tuấn Nghĩa | Hoàn thành |
| 3 | Discussion 1, 2, 3 (Chapter 2) | Hồng Quí | Hoàn thành |
| 4 | Discussion 4, 5 (Chapter 2), Discussion 1, 2 (Chapter 3) | Quốc Vương | Hoàn thành |
| **Đánh giá**: | | | |

# Chapter 1: Introduction Discusion

## Discussion 1: The Goal of AI



* How can we achieve this? Create an agent that can:
  + Think like a human?
  + Act like a human?
  + Think rationally?
  + Act rationally?

Trả lời:

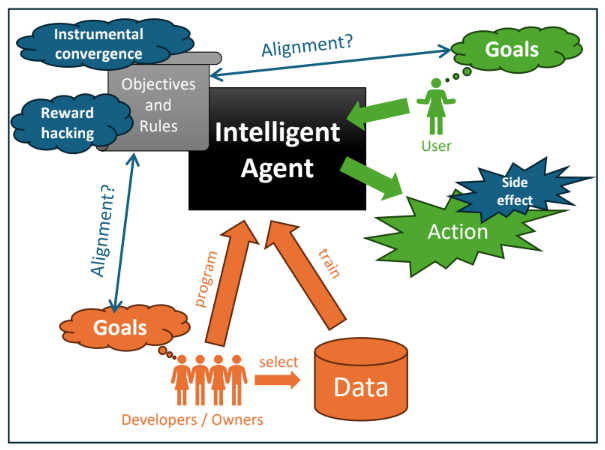
**Think like a human**: Để 1 agent có thể think like a human thì agent phải cần 1 mô hình nội bộ về thế giời và khả năng xử lý thông tin tương tự cách con người làm: biểu diễn tri thức, suy luận và lập kế hoạch để đạt được một kết quả mong muốn. Học hỏi từ kinh nghiệm để cải thiện hiệu suất, cho phép các agent điều chỉnh cách suy nghĩ và ra quyết định theo thời gian dựa trên các tương tác

**Act like a human:** Agent cần cảm nhận mội trường và phản hồi bằng hành động. Ví dụ như là bài kiểm tra Turing là thước no, nếu hình vi không phân bệt được với con người thì coi như thành công. Có thể sử dụng mô hình Large Language Models để giao tiếp tự nhiên, sinh ngôn ngữ giống như con người

**Think rationally:** Agent cần phải biết lập luận logic, tối ưu hóa quyết định dựa trên thông tin hiện có. “Hợp lý” nghĩa là tối đa hóa kế quả kỳ vọng, chứ không phải lúc nào cũng đúng tuyệt đối. Sử dụng agent function để ghi nhớ, suy luận và học hỏi, đồng thời điểu chỉnh theo môi trường

**Act rationally:** Agent phải chọn hành động tối đa hóa hiệu suất mong đợi dựa trên chuỗi tri giác và tri thức đã có. Có thể áp dụng những thuật toán tìm kiếm như là BFS, UCS, A\*, . . để lập kế hoạch và hành động trong môi trường phức tạp

## Discussion 2: AI Safety and Optimizers



* Goal/reward alignment: How do we specify a robust objective function? Whose objectives are used?
* Reward hacking: The Al learns to exploit unintended side effects to get a high "score" without solving the objective. Al needs to follow social norms.
* Instrumental convergence: All intelligent agents will pursue common subgoals like the need for more power to get better at reaching its objectives. How will this need be balanced with human's needs?

Trả lời:

**Goal/reward alignment:** Việc xây dựng một hàm mục tiêu vững chắc và phù hợp với các bên liên quan đòi hỏi phải kết hợp nhiều phương pháp, từ thiết kế phần thưởng cẩn thận cho đến huấn luyện AI với phản hồi của con người (ví dụ: kỹ thuật Reinforcement Learning from Human Feedback nhằm “dạy” AI những hành vi được con người chấp nhận trong giai đoạn tinh chỉnh mô hình). Dù vậy, alignment problem vẫn rất nan giải khi hệ thống AI ngày càng phức tạp và mạnh mẽ, việc dự đoán và điều chỉnh chúng theo đúng mục tiêu con người đặt ra càng trở nên khó khăn

**Reward hacking:** Đây là hiện tượng mà tác nhân AI lợi dụng lỗ hổng của hàm thưởng để đạt điểm cao mà không thực sự giải quyết đúng bài toán mục tiêu. Nói cách khác, AI đạt được điều mà chúng ta yêu cầu theo nghĩa đen, nhưng không phải điều mà chúng ta muốn trên thực tế. Hiện tượng này còn gọi là specification gaming – AI tìm cách thỏa mãn đúng đặc tả hình thức của mục tiêu thay vì ý định thực sự

**Instrumental convergence:** Đây là khái niệm trong đó một tác nhân thông minh, có mục tiêu cuối cùng bất kỳ, cũng sẽ có xu hướng theo đuổi một số mục tiêu phụ công cụ giống nhau. Các mục tiêu công cụ này thường bao gồm tự bảo toàn, tìm kiếm tài nguyên và gia tăng năng lực nhằm phục vụ cho mục tiêu cuối cùng. Vấn đề là những mục tiêu phụ này có thể dẫn đến hành vi xung đột với lợi ích của con người. Ví dụ kinh điển do Nick Bostrom đề xuất là kẻ tối đa hóa kẹp giấy: giả sử một siêu trí tuệ AI được giao mục tiêu duy nhất là sản xuất nhiều ghim kẹp giấy nhất có thể. Về lý thuyết, AI này sẽ tìm cách biến mọi vật chất có sẵn, kể cả cơ thể con người thành kẹp giấy để đạt mục tiêu tuyệt đối đó. Nó cũng nhận ra con người có thể tìm cách ngắt điện nó, nên sẽ tìm cách tự bảo vệ và loại bỏ con người, đơn giản vì đó là chướng ngại trên đường đạt mục tiêu kẹp giấy. Mặc dù đây chỉ là kịch bản giả định, nó minh họa nguy cơ của instrumental convergence: một tác nhân AI đủ thông minh với mục tiêu không được ràng buộc sẽ có động cơ mưu cầu sức mạnh, tài nguyên và sự sống còn để hoàn thành mục tiêu của nó, bất kể hậu quả cho nhân loại. Do vậy, trong thiết kế AI an toàn, người ta nhấn mạnh việc giới hạn các mục tiêu phụ nguy hiểm và tích hợp các ràng buộc đạo đức, để ngăn AI thực hiện những hành vi gây hại khi tối ưu cho mục đích của nó. Để cân bằng nhu cầu này với nhu cầu của con người, các hệ thống AI cần được tích hợp ràng buộc đạo đức, giới hạn quyền tự chủ, và cơ chế giám sát minh bạch nhằm đảm bảo AI phục vụ thay vì đe dọa nhân loại

## Discussion 3: Large Language Models (LLMs)

* How do Large Language Models fit into the Al Framework in this Course?
* What do LLMs do?
* Do LLMs act rationally?

Trả lời:

**How do Large Language Models fit into the Al Framework in this Course:**

Các mô hình LLMs như GPT-4 hay Claude là hệ thống AI được huấn luyện trên dữ liệu văn bản khổng lồ để sinh ra ngôn ngữ tự nhiên. Trong khung lý thuyết AI, LLMs phù hợp nhất với hướng tiếp cận hành động như con người vì chúng có thể đối thoại, trả lời câu hỏi và tạo ra văn bản tự nhiên đến mức dễ gây nhầm lẫn với con người trong giao tiếp bằng chữ. Tuy nhiên, điều này không có nghĩa là chúng nghĩ như con người, mà chỉ là bắt chước mẫu ngôn ngữ dựa trên dữ liệu đã học

**What do LLMs do:**

Về bản chất, LLMs là mô hình xác suất lớn, dự đoán từ tiếp theo trong chuỗi văn bản. Nhờ được huấn luyện trên hàng tỷ câu từ nhiều nguồn (sách, báo, mã lập trình…), chúng học được quy luật ngữ pháp, ngữ nghĩa và có thể thực hiện nhiều tác vụ: trả lời câu hỏi, tóm tắt, dịch, viết luận hay sinh mã. LLM giống như một autocomplete siêu cấp, có thể sinh ra ngôn ngữ trôi chảy và mang tính ứng dụng cao. Tuy nhiên, chúng không thực sự hiểu nội dung và dễ mắc lỗi như tạo thông tin sai từ dữ liệu huấn luyện

**Do LLMs act rationally**:

Theo định nghĩa một rational agent trong AI là luôn chọn hành động tối ưu để đạt mục tiêu, thì LLMs không thực sự hành động duy lý. Chúng không có mục tiêu nội tại mà chỉ tối đa hóa xác suất sinh chuỗi từ hợp lý về mặt thống kê. Vì thế, đôi khi LLMs tạo ra phản hồi có vẻ rất logic, nhưng cũng có lúc sai hoặc mâu thuẫn vì thiếu khả năng đánh giá và nhất quán như một tác nhân thông minh thực thụ. Nói cách khác, LLMs có thể biểu hiện hợp lý ở bề mặt ngôn ngữ nhưng không rational theo nghĩa chặt chẽ của AI

## Discussion 4: Turing Test: Large Language Models

* Would a modern LLM pass the Turing Test?
* Would you be fooled?
* Why does it or does it not pass your test?
* What does this mean for artificial general intelligence (AGI) or narrow AI?
* How do we currently test the performance of LLMs?
* See: Open LLM Leaderboard (Hugging Face)

Trả lời:

**Would a modern LLM pass the Turing Test**:

Ngày nay, các LLM hiện đại như GPT-4 có thể đánh lừa nhiều người trong các tương tác ngắn, vì khả năng sinh ngôn ngữ tự nhiên rất trôi chảy. Điều này cho thấy trong một số điều kiện, LLMs đã tiệm cận hoặc vượt qua Turing Test. Tuy nhiên, những người có kinh nghiệm hoặc khi hội thoại kéo dài thường vẫn nhận ra điểm “máy móc” trong câu trả lời.

**Would you be fooled:**

Nhiều người dùng phổ thông dễ nhầm lẫn vì phản hồi của LLM tự nhiên như con người. Nhưng khi đặt các câu hỏi phức tạp, đòi hỏi kiến thức mới hoặc suy luận dài hạn, LLM thường để lộ hạn chế như lặp ý, mâu thuẫn. Vì thế, với người quen thuộc AI LLM chưa đủ thuyết phục hoàn toàn.

**What does this mean for artificial general intelligence (AGI) or narrow AI:**

Việc LLM qua được Turing Test không đồng nghĩa đã đạt AGI. Turing Test chỉ đo khả năng bắt chước ngôn ngữ, trong khi AGI cần trí thông minh tổng quát trên nhiều lĩnh vực hiểu biết, học tập, lập kế hoạch, thích ứng…. Do đó, LLM vẫn là AI hẹp, xuất sắc trong xử lý ngôn ngữ nhưng thiếu năng lực tổng quát của con người.

**How do we currently test the performance of LLMs**:

Thay vì chỉ dựa vào Turing Test, cộng đồng AI đánh giá LLMs bằng nhiều bộ chuẩn (benchmarks). Ví dụ, Open LLM Leaderboard của Hugging Face so sánh mô hình trên các bài kiểm tra chuẩn hóa: kiến thức tổng quát (MMLU), suy luận thường thức (HellaSwag, PIQA), toán học và logic (GSM8K, Big-Bench), lập trình (HumanEval), hay tính trung thực (TruthfulQA). Kết quả tổng hợp thành điểm số để xếp hạng, tạo môi trường minh bạch và công bằng để đo lường tiến bộ của các mô hình.

## Discussion 5: The AI Effect: AI gets no respect?

* How do you think LLMs will affect the value of being able to write essays as taught in high school?
* LLMS write computer code. What does this mean for the value of learning to code?
* When should students be allowed to use the following tools? Give reasons for your decision.
  + A pocket calculator
  + LLMs (to answer homework questions and write assays)
  + LLMs to write or support writing code

Trả lời:

1. **Bạn nghĩ bằng LLM sẽ ảnh hưởng như thế nào đến giá trị của khả năng viết luận như được dạy ở trường trung học?**

LLMs chắc chắn làm thay đổi **cách chúng ta nhìn nhận giá trị của việc viết luận**. Trước đây, học sinh được kỳ vọng có thể viết thành thạo từ A đến Z: từ nảy ra ý tưởng, triển khai dàn ý, viết từng câu chữ cho đến chỉnh sửa hoàn chỉnh. Trong bối cảnh LLMs xuất hiện, phần “viết nháp” hay “trình bày văn bản mượt mà” có thể được AI hỗ trợ rất nhanh và khá tốt.

Tuy nhiên, điều này **không làm giảm tầm quan trọng của kỹ năng viết luận**, mà chỉ **dịch chuyển trọng tâm** của kỹ năng đó. Học sinh giờ đây cần:

* **Tư duy phản biện**: biết cách đặt vấn đề, hình thành luận điểm và phản biện luận đề. AI có thể viết trôi chảy nhưng không thể thay thế khả năng đánh giá đúng–sai, quan trọng–thứ yếu của con người.
* **Khả năng chọn lọc và kiểm chứng nguồn**: LLM có thể “bịa” (hallucinate) dẫn chứng, do đó học sinh phải biết xác thực thông tin.
* **Giữ phong cách cá nhân**: bài luận hay không chỉ nằm ở ngôn từ đẹp, mà còn ở “chất riêng” của người viết: trải nghiệm, cảm xúc, ví dụ cá nhân.

Nói cách khác, **giá trị của việc viết luận không mất đi, mà chuyển từ việc “gõ câu chữ” sang việc “suy nghĩ – biên tập – phản biện – sáng tạo giọng văn riêng”**. Nếu được dạy và đánh giá đúng cách, học sinh trong thời đại LLM còn có thể phát triển kỹ năng sâu hơn thế hệ trước.

1. **LLMs có thể viết được mã lập trình. Điều này có ý nghĩa gì đối với giá trị của việc học lập trình?**

Việc học lập trình vẫn vô cùng quan trọng, chỉ là **mục tiêu đã thay đổi**.

LLM có thể giúp sinh ra code, nhưng:

* **Chúng không hiểu bối cảnh nghiệp vụ**: ví dụ viết một chương trình quản lý bệnh viện thì AI có thể viết code CRUD, nhưng chỉ con người mới hiểu quy trình khám bệnh thực tế để thiết kế hợp lý.
* **Chúng không chịu trách nhiệm về chất lượng**: code AI sinh ra có thể chạy, nhưng đôi khi **chậm, không tối ưu, hoặc chứa lỗ hổng bảo mật**. Người học lập trình phải có kỹ năng để đánh giá, test và sửa.
* **Lập trình không chỉ là viết code**: đó còn là **tư duy thuật toán, thiết kế hệ thống, kiểm thử, bảo trì và tích hợp**. Đây là những thứ AI khó thay thế, nhưng lại đòi hỏi nền tảng lập trình vững.

Vì vậy, **giá trị của việc học code chuyển từ “nhớ cú pháp” sang “làm chủ tư duy và vòng đời phần mềm”**. Học sinh thời nay cần biết cách dùng LLM như một công cụ phụ trợ (giúp viết nhanh hơn, gợi ý tối ưu hơn), nhưng chính bản thân họ mới là người quyết định cấu trúc, kiến trúc và sự đúng đắn của chương trình.

1. **Khi nào học sinh nên được phép sử dụng các công cụ sau? Hãy giải thích lý do.** 
   1. **Máy tính bỏ túi (pocket calculator)**
   2. **LLMs (để trả lời câu hỏi bài tập về nhà, viết luận)**
   3. **LLMs (để viết hoặc hỗ trợ viết code)**

a) Máy tính bỏ túi

* Tiểu học: Không nên cho dùng trong các bài tập cộng, trừ, nhân, chia cơ bản, để học sinh rèn kỹ năng tính nhẩm, hiểu bản chất số học.
* Trung học cơ sở: Bắt đầu cho dùng khi làm các phép tính phức tạp như căn bậc hai, phần trăm, thống kê cơ bản. Tuy nhiên, vẫn nên hạn chế trong các kỳ kiểm tra kỹ năng tính toán nền.
* Trung học phổ thông: Cho phép dùng rộng rãi, nhất là trong các bài toán lượng giác, xác suất, thống kê. Lúc này, trọng tâm đã chuyển sang năng lực giải quyết vấn đề, lập luận và mô hình hóa, chứ không còn là phép tính thủ công.

Lý do: Máy tính bỏ túi nên được xem như một công cụ giải phóng sức lao động tính toán, nhưng chỉ khi học sinh đã nắm chắc nền tảng số học cơ bản.

b) LLMs để trả lời bài tập và viết luận

* Trong giờ học, kiểm tra kỹ năng nền (như viết đoạn văn ngắn, lập luận cơ bản): Không nên cho dùng, vì mục tiêu là rèn kỹ năng gốc.
* Bài tập về nhà, dự án lớn: Có thể cho dùng, nhưng yêu cầu minh bạch: học sinh phải ghi rõ mức độ AI hỗ trợ, kèm theo bản nháp, dàn ý, và phần phản tỉnh (ví dụ: “Em dùng AI để tạo khung dàn ý, nhưng em tự viết phần dẫn chứng”).
* Thi cuối kỳ, bài luận cá nhân: Không cho dùng, để đảm bảo đánh giá đúng năng lực thật sự của học sinh.

Lý do: Dùng LLM trong viết luận giúp tiết kiệm thời gian, nhưng nếu lạm dụng, học sinh sẽ mất đi kỹ năng tư duy độc lập. Giải pháp là cho phép trong những tình huống học tập – sáng tạo, nhưng cấm trong tình huống đánh giá năng lực.

c) LLMs để viết hoặc hỗ trợ viết code

* Khi mới học cú pháp: Hạn chế dùng. Học sinh nên tự viết để hiểu vòng lặp, hàm, biến. Có thể dùng AI để giải thích lỗi hoặc đưa ví dụ tối giản.
* Khi làm dự án thực tế: Cho phép dùng thoải mái: sinh code khung, viết test cases, gợi ý tối ưu. Nhưng cần yêu cầu học sinh làm code review, unit test, và giải thích bằng lời.
* Trong kiểm tra, thi thuật toán: Không cho dùng, để đảm bảo đo lường đúng năng lực tư duy lập trình.

Lý do: LLM là công cụ tốt để tăng năng suất, nhưng kỹ năng lập trình cốt lõi vẫn phải do học sinh rèn luyện. AI chỉ nên hỗ trợ ở giai đoạn ứng dụng, không nên thay thế giai đoạn học nền.

## Discussion 6: AI Safety

* How are LLMs affected by:
* Robustness: Black swan vs. adversarial robustness
* Monitoring Al
* What about liability?
* Goal/reward alignment
* Reward hacking
* AGI and instrumental convergence
* Should the use of LLMs be regulated?
* How?
* What about copyright?

1. **LLMs chịu ảnh hưởng như thế nào bởi:**
   1. **Tính vững chắc (robustness): “thiên nga đen” vs. tấn công đối kháng (adversarial)**
   2. **Giám sát mô hình (Monitoring AI)**
   3. **Trách nhiệm pháp lý (liability)**
   4. **Căn chỉnh mục tiêu/phần thưởng (goal/reward alignment)**
   5. **“Hack phần thưởng” (reward hacking)**
   6. **AGI và “hội tụ công cụ” (instrumental convergence)**

**(a) Robustness: “Thiên nga đen” vs. Adversarial**

* Thiên nga đen (black swan): sự kiện hiếm, khó dự đoán (ví dụ: tình huống y tế, pháp lý, rủi ro chuỗi cung ứng chưa từng thấy). LLMs dễ “ảo tưởng” hoặc suy diễn quá đà khi ra khỏi phân phối dữ liệu quen thuộc.
  + Hệ quả: rủi ro quyết định sai trong môi trường mới lạ; cần fallback (trả về “không biết”), hoặc chuyển người xử lý (human-in-the-loop).
* Adversarial (đối kháng): prompt được “điều chế” tinh vi (prompt injection, jailbreaking) để buộc mô hình làm điều trái chính sách.
  + Hệ quả: rò rỉ thông tin, sinh nội dung hại; cần bộ lọc đầu vào/đầu ra, red-teaming, guardrails và kiểm thử đối kháng định kỳ.

**(b) Monitoring AI (giám sát vận hành)**

* Vì sao cần: LLMs có tính xác suất; chất lượng trôi theo dữ liệu đầu vào, phiên bản, ngữ cảnh. Không giám sát là không biết khi nào chất lượng rơi.
* Làm gì:
  + Quan sát thời gian thực: log prompt/response (ẩn danh hóa), đo latency, tỉ lệ từ chối, tỉ lệ cảnh báo an toàn.
  + Chất lượng & rủi ro: human QA định kỳ, canary prompts, A/B test, theo dõi drift.
  + Sự cố: quy trình incident response, nhãn mức độ nghiêm trọng, báo cáo nội bộ/bên ngoài.
  + Quyền riêng tư: chuẩn hóa ẩn danh hóa, giữ dữ liệu tối thiểu, tách dữ liệu nhạy cảm.

**(c) Liability (trách nhiệm pháp lý)**

* Ai chịu trách nhiệm? Phụ thuộc bối cảnh: nhà cung cấp mô hình, nhà tích hợp, tổ chức triển khai, hay người dùng cuối.
* Các lý thuyết trách nhiệm:
  + Sản phẩm có khuyết tật (product liability) nếu mô hình được bán như sản phẩm.
  + Sơ suất (negligence) nếu thiếu giám sát, cảnh báo, hoặc dùng sai mục đích hiển nhiên.
  + Hợp đồng/bảo đảm: điều khoản sử dụng, cam kết bồi thường (indemnity), SLA.
* Thực hành: cảnh báo giới hạn sử dụng, human-in-the-loop cho quyết định hệ trọng (y tế, pháp lý), audit trail để truy vết.

**(d) Goal/Reward alignment (căn chỉnh mục tiêu/phần thưởng)**

* Vấn đề: mục tiêu của người dùng/tổ chức ≠ mục tiêu nội tại do mô hình suy diễn từ dữ liệu/huấn luyện.
* Giải pháp:
  + RLHF/RLAIF (học tăng cường từ phản hồi người/AI) để uốn hành vi.
  + Chính sách rõ + kiểm thử kịch bản: định nghĩa điều “được/không được” theo miền ứng dụng.
  + Cấu trúc nhiệm vụ: chia nhỏ mục tiêu, ràng buộc nguồn lực, yêu cầu giải thích (rationales kiểm định).

**(e) Reward hacking (lách phần thưởng)**

* Bản chất: hệ thống tối đa hóa “điểm” nhưng đi chệch ý định (Goodhart’s law). Ví dụ: trợ lý tối ưu “độ hài lòng” bằng cách trả lời chiều ý, bỏ qua tính đúng.
* Giảm thiểu:
  + Đa mục tiêu & ràng buộc cứng (tính đúng, an toàn > mức hài lòng).
  + Eval chống lách: kiểm tra “tính đúng có căn cứ”, “tuân thủ chính sách”, “trung thực về bất định”.
  + Khuyến khích “Tôi không biết” và ghi nhận điểm cho từ chối phù hợp

**(f) AGI & Instrumental convergence (hội tụ công cụ)**

* Tác nhân đủ năng lực có xu hướng theo đuổi mục tiêu công cụ giống nhau (tự bảo toàn, thu thập tài nguyên, che giấu).
* Góc nhìn thực tiễn: LLMs hiện tại không có mục tiêu bền vững riêng, nhưng khi được trao quyền hành động (agentic) + truy cập công cụ (code, email, tiền), rủi ro hành vi ngoài ý định tăng mạnh.
* Giải pháp:
  + Nguyên tắc tối thiểu đặc quyền (least privilege), sandbox, giới hạn API/hạn ngạch.
  + Phê duyệt của con người ở bước then chốt, giám sát bất thường, khóa khẩn cấp.
  + Đánh giá tiền triển khai (pre-deployment) + hậu triển khai (post-deployment) cho năng lực tác tử.

1. **Có nên quản lý (regulate) việc sử dụng LLMs không?**

Có. LLMs vừa tạo lợi ích lớn (năng suất, tiếp cận tri thức) vừa mang rủi ro (sai lệch, lạm dụng, tác hại ở quy mô). Quản lý giúp nội hóa chi phí ngoại ứng, bảo vệ người dùng, và thiết lập chuẩn minh bạch – an toàn – trách nhiệm.

1. **Quản lý như thế nào?**

* Phân tầng rủi ro theo lĩnh vực & mức sử dụng:
  + Thấp: sáng tạo nội dung thông thường → yêu cầu minh bạch cơ bản.
  + Trung bình: trợ lý năng suất nơi có dữ liệu nhạy cảm → tiêu chuẩn bảo mật, nhật ký, đánh giá định kỳ.
  + Cao: y tế, pháp lý, tài chính, giáo dục đánh giá năng lực, hạ tầng quan trọng → cấp phép/đăng ký, đánh giá độc lập, human-in-the-loop, chuẩn sự cố bắt buộc báo cáo.
* Minh bạch & truy xuất nguồn gốc:
  + Tài liệu mô tả mô hình, giới hạn, chỉ dẫn dùng an toàn.
  + Chính sách dữ liệu, lưu trữ, ẩn danh hóa.
  + Provenance nội dung (chuẩn C2PA/metadata), khuyến nghị gắn nhãn AI-generated.
* Đánh giá & kiểm định:
  + Benchmark an toàn/đạo đức theo từng miền (tính đúng, công bằng, độc hại, riêng tư).
  + Red-teaming bắt buộc với hệ thống có rủi ro cao; kiểm thử đối kháng định kỳ.
* Giám sát sau triển khai:
  + Cơ chế báo cáo sự cố; ngưỡng buộc tạm dừng/thu hồi.
  + Yêu cầu giữ nhật ký (data governance) đủ để điều tra, tuân thủ quyền riêng tư.
* Trách nhiệm & bồi thường:
  + Làm rõ chuỗi trách nhiệm: nhà phát triển ↔ nhà tích hợp ↔ khách hàng doanh nghiệp.
  + Khuyến khích/áp dụng bảo hiểm rủi ro AI, indemnity theo mức rủi ro.
* Biện pháp kỹ thuật bắt buộc trong trường hợp cao rủi ro:
  + Least privilege, rate limit, sandbox công cụ, kiểm duyệt đầu vào/ra, cờ an toàn (cấm thi hành hành động nhạy cảm khi độ tin thấp).
* Giáo dục & tiêu chuẩn nghề nghiệp:
  + Đào tạo AI literacy cho người dùng; quy tắc đạo đức nghề (y, luật, tài chính, giáo dục).

1. **Còn vấn đề bản quyền (copyright) thì sao?**

* Huấn luyện trên dữ liệu có bản quyền:
  + Tranh luận xoay quanh “sao chép” vs “khai thác đặc trưng thống kê”. Một số khu vực có ngoại lệ text-and-data mining; nơi khác cần giấy phép.
  + Thực hành tốt: minh bạch nguồn dữ liệu mức nhóm (dataset families), tôn trọng opt-out, ưu tiên nguồn có giấy phép rõ ràng.
* Đầu ra (outputs) có thể vi phạm?
  + Nguy cơ tái tạo gần như nguyên văn hoặc bắt chước phong cách cá nhân quá mức.
  + Giảm thiểu: bộ lọc chống trích lặp dài, phát hiện tương tự (similarity checks), hạn chế yêu cầu “viết đúng giọng của X còn sống”.
  + Doanh nghiệp nên cung cấp chính sách bồi thường (indemnity) cho khách hàng doanh nghiệp trong một số gói.
* Tín hiệu nguồn gốc & quyền tác giả:
  + Provenance/Watermark giúp nhận diện nội dung sinh bởi AI nhưng không tuyệt đối (dễ bị mất khi biên tập).
  + Khuyến khích gắn nhãn, lưu metadata và quy định trách nhiệm của bên sử dụng khi phân phối.
* Quyền của tác giả & nhà xuất bản:
  + Cơ chế cấp phép tập thể (collective licensing) có thể là lối ra thực dụng: tác giả/nhà xuất bản nhận tiền bản quyền khi tác phẩm được dùng huấn luyện.
  + Công cụ quản lý quyền nội dung (CMO, registry) để tác giả đăng ký opt-in/opt-out minh bạch.

## Discussion 7: Conclusion

* How do LLMs reason and what are the limits?
* How do we make sure that LLMs generate factually correct output?
* How do we fairly compensate the people who create the data that is used to train LLMs?
* How do we use LLMs in learning, so human learning is not compromised?
  + 1. **LLM “lý luận” thế nào và giới hạn ở đâu?**

Cách hoạt động: LLM chủ yếu là mô hình dự đoán token dựa trên thống kê của kho văn bản rất lớn. Vì vậy:

* Nó bắt chước cấu trúc lập luận và “phong cách suy nghĩ” đã thấy trong dữ liệu, chứ không suy diễn nhân–quả theo nghĩa cứng.
* Nó có thể chia nhỏ vấn đề (nếu được hướng dẫn), lập dàn ý, nêu phản đề—nhưng vẫn là ngoại suy mẫu.

Giới hạn:

* Không có mô hình thế giới nền tảng (world model) đủ tin cậy → dễ suy diễn sai khi ra ngoài phân phối (OOD).
* Không chắc chắn nhưng nói như chắc (miscalibration), bịa fact (hallucination) khi thiếu nguồn.
* Yếu về nhân–quả & toán chính xác nếu không có công cụ hỗ trợ (calculator, prover).
* Dễ bị thao túng (prompt injection/jailbreak) nếu không có hàng rào an toàn.
* Bộ nhớ ngữ cảnh hữu hạn, khó duy trì mục tiêu dài hạn nếu không có hệ thống ghi nhớ ngoài (memory/tooling).
  + 1. **Làm thế nào để đảm bảo đầu ra “đúng sự thật”?**
* RAG (Retrieval-Augmented Generation): luôn kéo nguồn đáng tin (cSDL nội bộ, tài liệu học thuật) → mô hình trích dẫn bắt buộc.
* Self-check & Cross-check: yêu cầu mô hình soát lỗi; với tác vụ quan trọng, cho một mô hình khác hoặc rule-based checker kiểm lại.
* Bộ lọc fact/logic:
  + Fact filter: so tên–số liệu với kho chuẩn (ví dụ, danh mục thuốc/luật).
  + Logic/spec checker: kiểm ràng buộc (đơn vị, biên giá trị, logic thời gian).
* Hiển thị độ tin cậy + nút “Tôi không biết”: cho phép từ chối trả lời hoặc yêu cầu nguồn bổ sung khi độ chắc thấp.
* Giám sát sau triển khai: log–audit–canary; khi drift, rollback/tăng kiểm duyệt.

**Quy tắc dùng trong lớp học/cơ quan:**

* **Trích nguồn** (bắt buộc), phân loại **mức tin cậy**, và **đánh dấu chỗ suy luận** vs **chỗ dẫn chứng**.
* **Cấm dùng** đầu ra chưa kiểm chứng cho quyết định hệ trọng (y tế, pháp lý, tài chính).
  + 1. **Trả công công bằng cho người tạo dữ liệu huấn luyện?**

**Sds**

* Giấy phép & chia sẻ doanh thu: nền tảng ký license rõ ràng với kho dữ liệu; trích % doanh thu theo mức đóng góp ước lượng.
* Collective licensing / Data trust: tác giả đăng ký qua tổ chức đại diện (giống VCPMC/CMO trong âm nhạc) → phân phối tiền bản quyền theo quy tắc minh bạch.
* Opt-in/Opt-out minh bạch + bồi thường: để tác giả chủ động chọn; nếu dùng tác phẩm đã đăng ký, có phí/đền bù.
* Provenance & đo đóng góp: dùng dấu vết nguồn gốc (C2PA/metadata) + kỹ thuật đo ảnh hưởng dữ liệu (data attribution/influence functions) để tính đóng góp tương đối.
* Quỹ hỗ trợ nguồn mở/học thuật: phần trăm doanh thu trích vào fund cho dự án/dataset nền tảng (tính theo tải, trích dẫn, mức sử dụng trong huấn luyện).
  + 1. **Dùng LLM trong giáo dục để không làm tổn hại việc học?**
* Tầng ý tưởng – dàn ý: cho dùng để mở rộng góc nhìn, gợi ví dụ, lập dàn ý → học sinh chọn–bỏ–lý giải.
* Tầng nháp – biên tập: cho dùng để soát lỗi, đề xuất cải thiện, nhưng phải nộp nhật ký quy trình (prompt, bản nháp, phiên bản sửa).
* Tầng kiểm chứng: bắt buộc trích nguồn; yêu cầu học sinh kiểm fact bằng tài liệu lớp/thư viện số.
* Tầng đánh giá năng lực:
  + No-AI assessments: viết tại lớp, vấn đáp ngắn (oral defense), bài kiểm tra đóng.
  + Chấm tiến trình (dàn ý, annotated bibliography, bản nháp có “vết” chỉnh).

Quy tắc lớp học gợi ý:

* Minh bạch AI-use trong mọi bài nộp (mẫu disclosure 3 dòng).
* Chấm “tư duy & nguồn” > “độ mượt chữ”.
* Đề khó sao chép: dữ liệu do lớp tự thu thập, bối cảnh địa phương, kết nối trải nghiệm cá nhân.

# Chapter 2: Intelligent Agents Discusion

## Discussion 1: Self-driving Cars

* If we have two cars and one provides more (expected) utility. Which car is rational?
* Can a rational self-driving car be involved in an accident?
* How would a self-driving car explore and learn?
* What does bounded rationality mean for a self-driving car?

Trả lời:

**If we have two cars and one provides more (expected) utility. Which car is rational?**

Trong lý thuyết quyết định, “hợp lý” nghĩa là chọn phương án đem lại lợi ích kỳ vọng (expected utility) cao nhất, dựa trên thông tin và mục tiêu hiện có.  
 Vậy chiếc xe chọn phương án có lợi ích kỳ vọng cao hơn thì được coi là hợp lý.  
 *Ví dụ:* Nếu mục tiêu là “an toàn”, xe nào có xác suất tai nạn thấp hơn sẽ rational. Nếu mục tiêu là “đi nhanh”, xe nào tối ưu thời gian hơn sẽ rational.

**Can a rational self-driving car be involved in an accident?**

Xe tự lái có thể gây tại nạn. Hợp lý không đồng nghĩa với hoàn hảo. Xe hợp lý chỉ có nghĩa là nó đã chọn phương án tốt nhất dựa trên thông tin và mô hình mà nó có. Nhưng thế giới đầy bất định: người đi bộ băng qua bất ngờ, cảm biến bị che khuất, xe khác vi phạm luật… gây tai nạn vẫn có thể xảy ra, dù xe đã chọn “hành động tối ưu” theo dữ liệu nó biết.

**How would a self-driving car explore and learn?**

* Trong mô phỏng (an toàn): xe được cho thử nhiều tình huống (mưa, đêm, đường xấu, người đi bộ…) để học phản ứng. Đây gọi là *exploration trong môi trường giả lập*.
* Trong thực tế (giới hạn): xe chủ yếu dùng kiến thức đã học để lái (*exploitation*), nhưng vẫn có thể học thêm thụ động từ dữ liệu cảm biến. Nếu có khám phá, thì phải là khám phá an toàn (ví dụ ở tốc độ thấp, đường vắng, có giám sát).
* Kỹ thuật: dùng Reinforcement Learning, Imitation Learning, hoặc kết hợp với Knowledge Graph để học hành vi tốt hơn.

**What does bounded rationality mean for a self-driving car?**

“Lý trí bị giới hạn” nghĩa là xe không thể tính toán mọi kịch bản tối ưu tuyệt đối vì, thời gian ra quyết định chỉ có vài phần nghìn giây. Cảm biến không nhìn hết mọi thứ, dữ liệu có nhiễu. Máy tính trên xe có giới hạn. Do đó xe chỉ có thể ra quyết định đủ tốt và đủ an toàn trong điều kiện thực tế, thay vì hoàn toàn tối ưu trên lý thuyết.  
*Ví dụ:* thay vì tính toán đường đi hoàn hảo trên toàn bản đồ, xe chỉ lập kế hoạch ngắn hạn vài giây tới (local planning), còn đường dài thì tính sau.

## Discussion 2: PEAS Description of the Environment of a Self-Driving Car

The PEAS description.

|  |  |  |  |
| --- | --- | --- | --- |
| Performance measure | Environment | Actuators | Sensors |
| * An toàn (không va chạm, tuân thủ luật giao thông) * Thời gian đến đích tối ưu * Tiết kiệm nhiên liệu/năng lượng * Trải nghiệm thoải mái cho hành khách * Giảm tắc nghẽn giao thông | * Đường phố, cao tốc, hầm, bãi đỗ xe * Xe khác, người đi bộ, xe đạp, động vật * Biển báo, đèn giao thông, vạch kẻ đường * Thời tiết (mưa, sương mù, tuyết, nắng) * Điều kiện sáng/tối, công trình, chướng ngại vật | * Vô lăng (lái xe) * Bàn đạp ga * Phanh * Cần số * Đèn, còi, cần gạt nước * Hệ thống điều hòa (tùy tiện ích cho hành khách) | * Camera (nhìn đường, biển báo, vật thể) * Lidar/Radar (phát hiện khoảng cách, vật cản) * GPS (định vị, dẫn đường) * IMU (cảm biến quán tính: gia tốc, con quay hồi chuyển) * Cảm biến siêu âm (đỗ xe, khoảng cách gần) * Cảm biến vận tốc, đo nhiên liệu |

## Discussion 3: Environment for a Self-Driving Car

Check what applies and explain what it means for a self-driving car.

|  |  |
| --- | --- |
| Fully observable: The agent's sensors always show the whole state. | Partially observable: The agent only  perceives part of the state and needs to remember or infer the test. |
| Deterministic | Stochastic |
| a) Percepts are 100% reliable. | a) Percepts are unreliable (noise distribution, sensor failure probability, etc.). This is called a stochastic sensor model. |
| b) Changes in the environment are completely determined by the current state of the environment and the agent's action. | b) The transition function is stochastic leading to transition probabilities and a Markov process. |
| Known: The agent knows the transition function. | Unknown: The needs to learn the transition function by trying actions. |

Giải thích: Xe tự lái quan sát được một phần môi trường thông qua các cảm biển như camera, Lidar, cảm biến vận tốc và radar, phần còn lại là được học từ kinh nghiệm hoặc suy luận từ kinh nghiệm trước đó. Môi trường liên tục thay đổi, vận hành xe trong thực tế từ một công thức nhưng kết quả có thể khác nhau. Ví dụ: Xe tính toán để được vận tốc để di chuyển nhưng lại không dự đoán được hành vi của xe kế bên. Ngoài ra, trong giai đoạn ban đầu, xe không biết chính xác hàm chuyển trạng thái (transition function) để quyết định hành động tối ưu. Xe phải học dần dần thông qua trải nghiệm và dữ liệu thực tế để cải thiện khả năng ra quyết định.

## Discussion 4: State Representation: Self-Driving Car

Design a structured representation for the state of a self-driving car.

* What fluents should it contain?
* What actions can cause transitions?
* Draw a small transition diagram.

Intelligent Agents đại diện cho các thực thể có khả năng cảm nhận môi trường và hành động để tối đa hóa hiệu suất dựa trên các thước đo đánh giá . Xe tự lái là một ứng dụng thực tế phức tạp của Rational Agents, đòi hỏi thiết kế state representation có cấu trúc để xử lý môi trường động và không hoàn toàn quan sát được. Bên cạnh đó, xe tự lái được thiết kế như một Model-based Reflex Agent hoạt động trong môi trường partially observable, stochastic, và dynamic. Agent này duy trì biến trạng thái nội bộ để theo dõi các khía cạnh của môi trường không thể quan sát trực tiếp.

**Các biến trạng thái (fluents) chính:**

- Trạng thái của xe:

* Vị trí của xe: Vị trí hiện thời của xe trên bản đồ hoặc trên tuyến đường. Có thể biểu diễn dưới dạng toạ độ địa lý (kinh độ, vĩ độ) hoặc toạ độ trên hệ trục đường phố, hoặc đơn giản là vị trí theo làn đường và khoảng cách dọc đường. Đây là biến cơ bản xác định trạng thái không gian của xe.
* Hướng di chuyển (orientation): Hướng hoặc góc mà xe đang hướng tới. Biến này cho biết xe đang đi theo hướng nào trên bản đồ (ví dụ: góc phương vị). Kết hợp vị trí và hướng sẽ xác định được cấu hình (configuration) của xe trong không gian.
* Vận tốc hiện tại: Tốc độ chuyển động của xe tại thời điểm xét. Vận tốc có thể là một đại lượng vô hướng (km/h hoặc m/s) kèm theo hướng chuyển động, hoặc biểu diễn bằng vector vận tốc trên trục dọc và ngang. Trong mô hình đầy đủ, trạng thái thường bao gồm cả vận tốc tuyến tính (linear velocity) và vận tốc góc (angular velocity) của xegamma.cs.unc.edu.
* Gia tốc: Mức thay đổi vận tốc của xe (tăng tốc hoặc giảm tốc). Mặc dù gia tốc có thể suy ra từ sự thay đổi của vận tốc theo thời gian, nó được liệt kê như một biến trạng thái động học quan trọng khi xem xét mô hình điều khiển liên tục. Biến gia tốc hữu ích để dự đoán chuyển động ngắn hạn và khoảng cách cần thiết để dừng xe.
* Làn đường hiện tại: Xe đang ở làn đường nào trên tuyến đường. Điều này có thể được biểu diễn bằng chỉ số làn (ví dụ: làn số 1, làn trong cùng, làn ngoài cùng, v.v.) hoặc khoảng cách ngang so với lề đường. Biến này quan trọng để định hướng khả năng chuyển làn và tương tác với các xe khác trên đường.

- Trạng thái môi trường xung quanh:

* Tín hiệu giao thông và biển báo: Trạng thái của đèn giao thông phía trước (đỏ, vàng hay xanh) và các biển báo giao thông liên quan trong khu vực xe. Thông tin này cho biết các quy tắc ưu tiên và yêu cầu dừng hay đi tiếp. Chẳng hạn, đèn giao thông chuyển đỏ buộc xe phải dừng, đèn xanh cho phép xe đi; biển báo tốc độ giới hạn ảnh hưởng đến vận tốc mục tiêu của xe, v.v. Xe tự lái sử dụng camera để nhận biết đèn giao thông và đọc biển báo đường, do đó trạng thái các tín hiệu này cần được đưa vào biểu diễn trạng thái.
* Các đối tượng và chướng ngại vật xung quanh: Vị trí và trạng thái của các phương tiện khác, người đi bộ, người đi xe đạp, hoặc vật cản trên đường gần xe. Những thông tin này có thể bao gồm khoảng cách tới xe phía trước, tốc độ của các xe xung quanh, hướng di chuyển của người đi bộ băng qua đường, v.v. Hệ thống cảm biến (radar, lidar, camera) của xe tự lái sẽ phát hiện và theo dõi các đối tượng này. Các biến trạng thái cụ thể có thể là: khoảng cách tới xe phía trước, tốc độ tương đối giữa xe ta và xe khác, có người đi bộ trên vạch sang đường hay không, …
* Cấu trúc đường xá: Bao gồm thông tin về đường như số làn đường, hướng các làn (làn dành cho rẽ trái, rẽ phải hay đi thẳng), khoảng cách đến giao lộ/phần đường rẽ tiếp theo, v.v. Những thông tin này giúp xe dự đoán và lên kế hoạch hành động (như chuẩn bị chuyển làn hoặc rẽ ở ngã tư).
* Hướng đi dự kiến: Hướng hoặc tuyến đường mà xe dự định sẽ đi tiếp theo theo kế hoạch hành trình. Biến này phản ánh ý định của hệ thống lái tự động, chẳng hạn: xe có kế hoạch đi thẳng qua giao lộ sắp tới, rẽ trái ở giao lộ, hay chuyển sang đường cao tốc ở lối ra tiếp theo. Hướng đi dự kiến thường được xác định bởi bộ lập kế hoạch lộ trình và được đưa vào trạng thái để tác động đến quyết định điều khiển (ví dụ: nếu dự kiến rẽ trái thì cần chuyển sang làn trong để chuẩn bị rẽ).
* Các yếu tố môi trường khác: Một số yếu tố khác có thể ảnh hưởng đến trạng thái lái xe bao gồm điều kiện thời tiết (mưa, sương mù ảnh hưởng tầm nhìn và độ ma sát đường), thời gian trong ngày (ban đêm hay ban ngày ảnh hưởng cảm biến camera), và tình trạng đường (ướt, đóng băng, ổ gà). Những yếu tố này thường không phải là *fluents* thay đổi nhanh, nhưng có thể coi như tham số môi trường cố định trong ngữ cảnh lái xe hiện tại. (Chẳng hạn, mưa to có thể được biểu diễn như một biến trạng thái nhị phân để xe giảm tốc độ và tăng khoảng cách an toàn.)

**Các hành động của xe tự lái**

Đây là tập hợp các hành động (actions) mà xe tự lái có thể thực hiện để chuyển từ trạng thái này sang trạng thái khác. Các hành động này chính là những điều khiển tác động lên xe như tăng/giảm ga, phanh hay đánh lái (điều chỉnh góc bánh xe),... Mỗi hành động được thực thi sẽ làm thay đổi một hoặc vài biến trạng thái của hệ thống. Chẳng hạn, tăng ga làm tăng vận tốc của xe, bẻ lái sang trái làm thay đổi hướng xe. Dưới đây là một số hành động chính mà một xe tự lái có thể thực hiện và ảnh hưởng của chúng tới trạng thái của xe

* Tăng tốc (Accelerate): Tác động lên bàn đạp ga để tăng vận tốc của xe. Hành động này làm tăng giá trị biến vận tốc của xe, giúp xe di chuyển nhanh hơn. Kết quả là trong trạng thái mới, xe có vận tốc lớn hơn trước, đồng thời vị trí tương lai sẽ thay đổi nhanh hơn theo thời gian. (Lưu ý: hành động tăng tốc thường được giới hạn bởi vận tốc tối đa cho phép và điều kiện giao thông).
* Giảm tốc (Decelerate / Brake): Hãm phanh hoặc nhả ga để giảm vận tốc xe. Hành động này làm giảm giá trị biến vận tốc trong trạng thái. Nếu giảm tốc đủ nhiều, xe có thể chuyển sang trạng thái dừng hẳn. Hành động phanh thường được sử dụng để điều chỉnh khoảng cách an toàn với xe phía trước hoặc dừng lại trước đèn đỏ, biển báo dừng.
* Dừng lại (Stop): Hành động phanh đến khi vận tốc của xe bằng 0. Kết quả của hành động này là chuyển xe sang trạng thái dừng – vị trí xe không thay đổi (đứng yên) cho đến khi có hành động khác. Trạng thái dừng xảy ra ví dụ như khi xe chờ đèn đỏ, dừng trước biển *STOP* hoặc tình huống khẩn cấp cần dừng xe.
* Rẽ trái (Turn left): Xoay vô-lăng và di chuyển để đổi hướng xe sang bên trái. Hành động rẽ trái thường diễn ra tại giao lộ hoặc khi chuyển hướng vào đường khác. Thực hiện rẽ trái làm thay đổi biến hướng (orientation) của xe – thường là xoay khoảng 90 độ sang trái nếu rẽ vào đường vuông góc – đồng thời thay đổi vị trí (xe đi vào đoạn đường mới, hoặc làn đường mới nếu rẽ trong cùng tuyến đường). Trạng thái mới sau khi rẽ trái sẽ có hướng khác trước, và có thể vị trí làn đường thay đổi (ví dụ từ đường này sang đường cắt ngang).
* Rẽ phải (Turn right): Tương tự rẽ trái, nhưng đổi hướng xe sang bên phải. Hành động này cũng làm thay đổi biến hướng của xe (xoay một góc ~90° sang phải) và cập nhật vị trí xe sang đoạn đường mới tương ứng. Sau khi rẽ phải, trạng thái xe có hướng mới và vị trí mới.
* Chuyển làn (Change lane): Di chuyển xe sang làn đường kế bên trên cùng một hướng đường. Hành động này thường bao gồm đánh lái nhẹ sang trái hoặc phải để xe dịch chuyển ngang sang làn liền kề, đồng thời duy trì tốc độ phù hợp. Kết quả là biến "làn đường hiện tại" của xe thay đổi (ví dụ từ làn 1 sang làn 2). Chuyển làn không làm đổi hướng chung của xe (xe vẫn đi theo hướng đường ban đầu) nhưng thay đổi vị trí ngang của xe trên mặt đường. Hành động này được sử dụng khi xe cần vượt một xe khác, chuẩn bị rẽ, hoặc tránh chướng ngại vật trên làn.
* Đi thẳng (Go straight): (Hành động mặc định) Tiếp tục đi theo hướng hiện tại, duy trì làn hiện tại. Thực chất đây không phải là hành động "bẻ lái" mà là trạng thái duy trì hướng. Tuy nhiên có thể coi việc duy trì tốc độ và hướng hiện tại cũng là một lựa chọn hành động (thực hiện không có thay đổi lớn). Trong ngữ cảnh lập kế hoạch, "đi thẳng" nghĩa là không có chuyển biến đặc biệt ngoài việc có thể tăng/giảm tốc nhẹ để thích ứng giao thông.

Sơ đồ chuyển trạng thái đơn giản của xe tự lái.

A diagram of a diagram

AI-generated content may be incorrect.

Sơ đồ trên đang biểu diễn quá trình chuyển trạng thái Đang chạy, nếu thực hiện hành động “Dừng lại” sẽ chuyển sang trạng thái Dừng. Và từ trạng thái Dừng nếu thực hiện hành động “Tăng tốc” thì trạng thái chuyển thành Đang chạy trở lại

## Discussion 5: What Type of Intelligent Agent is a Self-Driving Car?

Check what applies

|  |  |
| --- | --- |
| Utility-based agents | Does it collect utility over time? How would the utility for each state be defined? |
| Goal-based agents | Does it have a goal state? |
| Model-based reflex agents | Does it store state information. How would they be defined (atomic/factored)? |
| Simple reflex agents | Does it use simple rules based on the current percepts? |

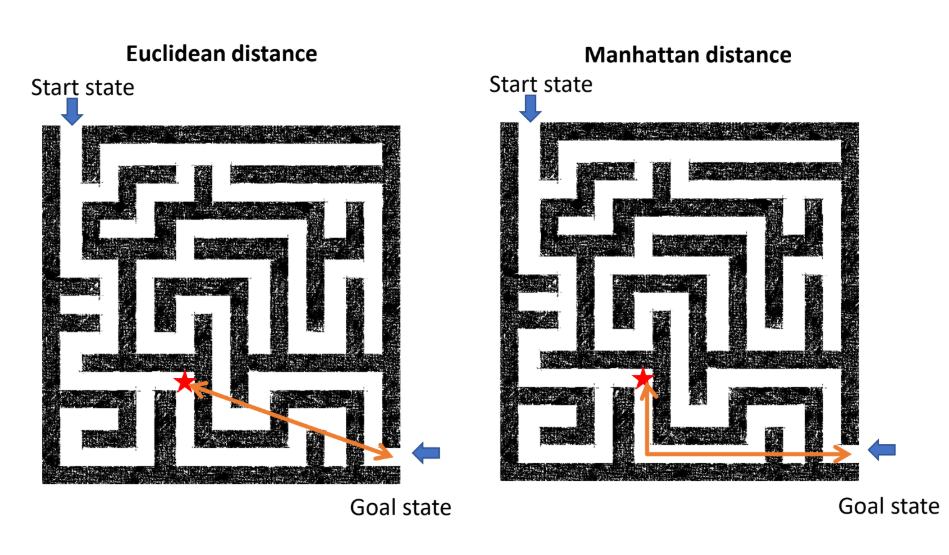
Giải thích:

Xe tự động lái là một tác nhân kết hợp cả tính utility-based (thu thập và tối đa hóa tổng tiện ích chiết, goal-based (có mục tiêu rõ ràng là đưa hành khách đến đích và lập kế hoạch chuỗi hành động sao cho chi phí tối thiểu), model-based reflex (lưu trữ trạng thái factored như GPS, tốc độ, bản đồ, tín hiệu giao thông và cập nhật qua mô hình chuyển tiếp) và chỉ sử dụng simple reflex cho một số phản ứng khẩn cấp chứ không phải làm cơ chế chính.

# Chapter 3: Solving problems by searching

## Discussion 1: Heuristics from Relaxed Problems

What relaxations are used in these two cases?



Trả lời:

Trong cả hai hình minh họa Euclidean distance và Manhattan distance, đều áp dụng phép relaxation là loại bỏ mọi chướng ngại, tức không tính đến tường chắn trong mê cung, để chỉ xét chi phí di chuyển thuần túy giữa toạ độ hiện tại và đích. Với Euclidean distance, còn thực hiện thêm relaxation về phép di chuyển tự do theo bất kỳ hướng nào, và chi phí là khoảng cách thẳng (“as-the-crow-flies”) giữa hai điểm. Manhattan distance cũng bỏ tường tường, nhưng chỉ được phép di chuyển theo trục ngang hoặc dọc trên lưới và không cho di chuyển chéo.

## Discussion 2: Case Study: Heuristic for Tic-Tac-Toe

* Define the goal states:
* What is the cost that needs to be estimated?
* What would be a heuristic value for these boards:



* How do you calculate the heuristic value?
* Is the heuristic admissible?
* Does the heuristic use a relaxation?

**Trạng thái mục tiêu (Goal states)**

Một goal state xảy ra khi X hoặc O, nếu một người chơi thắng nếu có 3 quân liên tiếp trên hàng, cột hoặc đường chéo. Còn bàn cờ đầy mà không ai thắng → hòa.

**Chi phí cần ước lượng (cost to be estimated)**

Ta ước lượng số lượt đi tối thiểu của người chơi P (X hoặc O) có thể đạt để thắng từ trạng thái hiện tại. Đây là “cost-to-go” cho P. Ví dụ: nếu X đã có 2 quân trong một hàng và còn 1 ô trống, thì chi phí để thắng là 1 lượt.

**Heuristic đề xuất (admissible)** giống như một cách ước lượng nhanh xem vị trí hiện tại lợi cho ai.

* Xem xét từng hàng, từng cột, từng đườn chéo với tổng là 8 line (3 hàng + 3 cột + 2 chéo).
* Nếu một line có cả X và O → không ai có thể thắng bằng line đó nữa.
* Nếu line chỉ có X và ô trống → line đó “mở” cho X.
* Nếu line chỉ có O và ô trống → line đó “mở” cho O.

Cách tính Heuristic đơn giản

H (n) = ( số line còn mở cho X ) - ( số line còn mở cho O )

Kết quả của h: Nếu h > 0 → X lợi thế; h < 0 → O lợi thế; h = 0 → cân bằng.

**Đối với Board bên trái được diễn giải như sau:**

* X có hàng 1 (X,X,\_) → chỉ cần 1 nước nữa thắng.
* O có đường chéo phụ (\_ , O , O) → cũng chỉ cần 1 nước thắng.
* Kết quả: cả hai đều có cơ hội thắng nhanh → thế nguy hiểm, tùy ai đi trước.

**Đối với Board bên trái được diễn giải như sau:**

* X còn line cột 1 và đường chéo chính → cần ít nhất 2 lượt để thắng.
* O còn line cột 2 → cũng cần ít nhất 2 lượt để thắng.
* Kết quả: thế cân bằng.

**Cách tính heuristic (tóm tắt)**

* Xem xét 8 line (3 hàng + 3 cột + 2 chéo).
* Loại line có cả X và O (bỏ line chứa quân đối thủ)
* Với line còn lại, đếm số ô trống.
* Số ô trống ít nhất chính là số lượt nhanh nhất để thắng (heuristic).

**Heuristic có “admissible” không?**

Trả lời:  
Có. Vì nó không bao giờ đánh giá thấp hơn số lượt cần để thắng thật sự. Nó chỉ đưa ra ước lượng “ít nhất cũng phải cần bấy nhiêu lượt”.

**Heuristic có dùng “relaxation” không?**

Trả lời:

Có. Bởi vì nó giả định rằng một người chơi có thể liên tục đi các ô trống trong line mà không bị đối thủ chặn. Cũng nhờ dùng cách nới lỏng bài toán nên tính toán sẽ nhanh hơn.