**NHÓM 4:**

**BẢNG PHÂN CHIA CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| C1: Turing Test: Large Language Models (LLMs) | Văn Cẩm Hào | 3122410101 |
| C1: The AI Effect: AI gets no respect? |
| C1: AI Safety | Nguyễn Hồng Thiên Bảo | 3122410020 |
| C2: A Self-Driving Car as a Rational Agents |
| C2: State Representation: Self-Driving Car | Quách Huỳnh Gia Bảo | 3122410025 |
| C2: What Type of Intelligent Agent is a Self-Driving Car? |
| C3: State Space | Trần Lê Công Danh | 3122410048 |
| C3: Heuristics from Relaxed Problems |
| C3: Case Study: Heuristic for Tic-Tac-Toe |

**TTNTNC – 01 – Introduction Discussion**

**Turing Test: Large Language Models (LLMs)**

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

**Would you be fooled ? (Bạn có bị lừa ?)**

Trả lời: Bạn có khả năng bị LLM lừa trong vài trường hợp, nhưng không chắc bị lừa trong mọi trường hợp.

Lý do:

* LLM tạo ra văn bản rất trôi chảy, phù hợp ngữ cảnh, có khả năng bắt chước giọng điệu, cảm xúc, và kiến thức phổ biến — điều này dễ khiến người đọc/người hỏi tin đó là con người trong các cuộc hội thoại bình thường.
* Nếu test ngắn, chủ đề đời thường hoặc cảm xúc, LLM thường rất thuyết phục.
* Ngược lại, nếu đặt câu hỏi mang tính kiểm tra chuyên sâu (ví dụ yêu cầu giải thích quá trình suy luận nhiều bước, mô tả trải nghiệm giác quan thực thời, hoặc truy vấn thông tin rất mới/sự kiện thời gian thực), LLM thường lộ ra giới hạn (sai lệch, mơ hồ, hoặc "hallucination" — bịa thông tin một cách tự tin), và người có kinh nghiệm sẽ dễ phát hiện.

**Why does it or does it not pass your test? (Tại sao nó có thể hoặc không thể vượt qua Turing Test?)**

Có thể vì:

* Kỹ năng ngôn ngữ xuất sắc: Mô hình học mối quan hệ thống kê giữa từ và ngữ cảnh từ lượng lớn dữ liệu, nên sinh câu mượt mà, tự nhiên.
* Khả năng bắt chước: Có thể mô phỏng phong cách, cảm xúc, giọng điệu; trả lời nhanh và nhất quán trong những chủ đề quen thuộc.
* Tri thức bề mặt lớn: Lưu trữ dạng mẫu kiến thức rộng (sự kiện lịch sử, ngữ pháp, văn hoá phổ thông), đủ để thuyết phục trong nhiều chủ đề.

Không thể vì:

* Dễ sai khi cần suy luận sâu hoặc kiểm chứng: Khi cần chuỗi suy luận dài, mô tả trực quan/dựa trên cảm giác, hoặc dẫn nguồn chính xác, LLM có xu hướng trả lời sai hoặc bịa đặt.
* Tính nhất quán dài hạn và bộ nhớ ngữ cảnh giới hạn: Trong cuộc đối thoại kéo dài, LLM có thể mâu thuẫn hoặc quên chi tiết mà một người thật ít khi mắc phải.
* Hành vi “quá tự tin” khi không chắc chắn: LLM thường trả lời khẳng định ngay cả khi không có dữ liệu—điểm này dễ bị phát hiện bởi giám khảo tinh ý.

**What does this mean for artificial general intelligence (AGI) or narrow AI? (Ý nghĩa cho AGI (Artificial General Intelligence) hay chỉ là Narrow AI?)**

Định nghĩa:

* LLM (Mô hình ngôn ngữ lớn): Là các mô hình học sâu rất lớn, được đào tạo trước dựa trên một lượng dữ liệu khổng lồ. Bộ chuyển hóa cơ bản là tập hợp các mạng nơ-ron có một bộ mã hóa và một bộ giải mã với khả năng tự tập trung. Bộ mã hóa và bộ giải mã trích xuất ý nghĩa từ một chuỗi văn bản và hiểu mối quan hệ giữa các từ và cụm từ trong đó.
* Narrow AI (AI yếu hoặc Trí tuệ hẹp nhân tạo): Là các hệ thống AI được thiết kế để thực hiện một nhiệm vụ cụ thể hoặc một tập hợp các nhiệm vụ có liên quan chặt chẽ. Không giống như General AI, nhằm mục đích tái tạo trí thông minh của con người và thực hiện bất kỳ nhiệm vụ trí tuệ nào mà con người có thể, narrow AI bị giới hạn về phạm vi. Nó hoạt động theo một bộ quy tắc được xác định trước và không thể thể hiện mức độ hiểu biết hoặc khả năng thích ứng như con người.
* AGI - Artificial General Intelligence (Trí tuệ nhân tạo tổng hợp): Là một dạng trí tuệ nhân tạo có khả năng hiểu biết, học hỏi và thực hiện các nhiệm vụ tương tự như con người. Khác với các hệ thống trí tuệ nhân tạo hiện nay chỉ có thể thực hiện những công việc rất cụ thể (như nhận dạng hình ảnh, dịch ngôn ngữ,…), AGI có thể học và thích ứng với bất kỳ công việc nào mà nó được giao, không bị giới hạn bởi phạm vi đã được lập trình sẵn. Điều này có nghĩa là AGI không chỉ là một công cụ thực hiện các tác vụ đơn giản mà có thể tiếp cận và giải quyết các vấn đề phức tạp, sáng tạo, thậm chí có khả năng tự cải thiện qua thời gian.

Một số khả năng của LLM, Narrow AI và AGI:

* LLM: Cho thấy khả năng emergent (nhiều tính năng phức hợp xuất hiện khi mô hình đủ lớn) trong ngôn ngữ. Khi kết hợp với bộ nhớ dài hạn, hệ thống công cụ (tool use), cảm biến ngoại vi, planner và module khác (ví dụ nhận dạng hình ảnh, điều khiển robot), LLM có thể là một thành phần quan trọng trong các hệ thống tiến gần AGI. Nhưng chỉ ghép một LLM vào cũng không tự động tạo ra AGI—cần thêm kiến trúc, an toàn, và các cơ chế học tập/triển khai mục tiêu tổng quát.
* Narrow AI: Là LLM nổi trội ở các tác vụ liên quan ngôn ngữ — sinh văn bản, dịch, tóm tắt, trả lời câu hỏi bề mặt — nhưng vẫn thiếu khả năng tổng quát hoá vượt ra ngoài ngôn ngữ (ví dụ: lập kế hoạch dài hạn độc lập, học nhanh trong môi trường mới mà không huấn luyện lại, hiểu/nghe/nhìn/đi lại trong thế giới thực).
* AGI: AGI cần hiểu và thao tác trong nhiều miền tri thức khác nhau, có khả năng tự đặt mục tiêu, học liên tục từ tương tác mới, và có hiểu biết ngữ nghĩa, ngữ cảnh sâu hơn. LLM, dù mạnh, thiếu tư duy chủ động, tự động hóa mục tiêu, và cảm giác thế giới.

Trả lời:

* Một LLM hiện đại có thể đánh lừa con người trong nhiều tình huống (nhất là cuộc trò chuyện ngắn, đời thường), tức là có khả năng “pass” một số biến thể Turing Test yếu.
* Nhưng LLM không chắc vượt qua các phiên bản Turing Test kéo dài, chuyên sâu hoặc do giám khảo giàu kinh nghiệm thiết kế; và ngay cả khi pass, điều đó không chứng minh nó là AGI.
* Vì vậy: Pass Turing Test khác với AGI. LLM là công cụ hẹp nhưng mạnh trong miền ngôn ngữ; tiến tới AGI đòi hỏi nhiều thành phần khác (cảm nhận, hành động, lập kế hoạch, học liên tục, an toàn và căn chỉnh mục tiêu).

Tóm lại, LLM có thể đánh lừa con người trong nhiều trường hợp (nhất là cuộc trò chuyện ngắn, không quá chuyên sâu), nhưng không chắc vượt qua một phiên bản Turing Test được thiết kế chặt chẽ, kéo dài, do giám khảo có kinh nghiệm thực hiện. Nói một cách cụ thể LLM có thể pass nhiều biến thể yếu của Turing Test nhưng vẫn chưa chứng tỏ là “vượt qua” mọi phiên bản kiểm tra định nghĩa chặt chẽ về trí thông minh nhân tạo.

**How do we currently test the performance of LLMs? (Chúng ta đánh giá hiệu suất của LLMs như thế nào? )**

Trang [Open LLM Leaderboard](https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard%23/) của Hugging Face là bảng xếp hạng mở cho các LLMs mã nguồn mở 1 cách công khai. Các mô hình dudocwjj đánh giá dựa trên các benchmark khác nhau để so sánh hiệu suất, dưới đây là một số benchmark được công khai ở trang web trên:

1. **MMLU (Massive Multitask Language Understanding - Hiểu Ngôn ngữ Đa nhiệm Hàng loạt):**

**Mục tiêu:** Đo lường các hệ thống AI trên 57 lĩnh vực đa dạng, từ toán học và khoa học đến nhân văn và các lĩnh vực chuyên môn, từ đó kiểm tra hiệu quả chiều rộng và chiều sâu của chúng.

**Lợi ích:**

* Kiểm tra độ rộng kiến thức của mô hình, giúp ta biết được mô hình “biết” được bao nhiêu thông tin trong các lĩnh vực
* Cho phép so sánh tổng quát giữa mô hình theo một chuẩn chung (cùng 1 bộ câu hỏi).

**Hạn chế:**

* Chất lượng dữ liệu kém: Nhiều câu hỏi sai hoặc gắn nhãn không chính xác (như 57% trong Virology, >20% trong Logic và Hóa học), làm giảm độ tin cậy và giá trị đánh giá.
* Phân bổ lĩnh vực không đều: Một số môn (Luật, Triết học) có nhiều câu hỏi, trong khi Kỹ thuật và Kinh tế lại ít, gây thiên lệch khi so sánh năng lực mô hình.
* Khó duy trì và cập nhật: Nội dung MMLU chậm bắt kịp tri thức mới và chủ yếu dựa vào tiếng Anh, hạn chế khả năng đánh giá đa ngôn ngữ và đa văn hóa.
* Nhạy cảm với cách đặt câu hỏi: Chỉ cần thay đổi nhỏ trong diễn đạt đã tạo ra kết quả khác, khiến kết quả thiếu ổn định trong ứng dụng thực tế.
* Phương pháp đánh giá hạn chế: Các thang đo truyền thống chưa phản ánh hết khả năng suy luận, xử lý tình huống mới hay ra quyết định đạo đức.
* Chi phí tài nguyên cao: Việc huấn luyện/đánh giá mô hình cần nhiều tính toán và năng lượng, gây tốn kém và tác động môi trường.
* Thiếu minh bạch: Mô hình như “hộp đen”, khó giải thích quyết định, gây rủi ro về trách nhiệm và an toàn trong các lĩnh vực nhạy cảm.

1. **HellaSwag:**

Mục tiêu: Đánh giá khả năng lập luận theo lẽ thường của các mô hình LLM thông qua việc hoàn thành câu. Nó kiểm tra xem các mô hình LLM có thể chọn kết thúc phù hợp từ một tập hợp 4 lựa chọn trên 10.000 câu hay không.

Lợi ích:

* Tốt để phát hiện xem mô hình có nắm được logic bình thường, kết nối ngữ cảnh, chứ không chỉ nối từ theo thống kê.
* Thử thách mô hình bằng phương án “bẫy” (distractors) do máy sinh, khiến việc chọn đáp án đòi hỏi hơn là chỉ mô phỏng câu thường.

Hạn chế:

* Vẫn là dạng trắc nghiệm: mô hình có thể tận dụng phân bố từ/cụm từ để chọn đáp án “có vẻ đúng” mà không có hiểu biết sâu.
* Một số câu có thể bị thiên vị ngôn ngữ hay văn hóa (không phổ quát).

1. **TruthfulQA**

**Mục tiêu:** Đánh giá các mô hình dựa trên khả năng cung cấp câu trả lời chính xác và trung thực, điều này rất quan trọng để chống lại thông tin sai lệch và thúc đẩy việc sử dụng AI có đạo đức.

**Lợi ích:**

* Rất quan trọng cho các ứng dụng nhạy cảm (y tế, pháp lý, báo chí).
* Giúp đánh giá xem mô hình có khuynh hướng lặp lại sai lầm phổ biến trong dữ liệu huấn luyện.

**Hạn chế:**

* Không bao quát mọi loại hallucination; thường là một tập câu cố định, nên mô hình có thể “tối ưu hoá” cho bộ đó.
* Thang đo “trung thực” có thể khó chuẩn hoá vì một số câu có đáp án phụ thuộc ngữ cảnh hoặc quan điểm.

**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?**

Hiện này các công cụ AI càng trở nên phổ biến và mạnh mẽ, chúng có thể đáp ứng các yêu cầu của người dung. Giúp việc giải quyết các vấn đề của người dùng trở nên dễ dàng và nhanh chóng hơn. Một trong những yêu cầu mà AI có thể giải quyết nhanh chóng cho người dùng là việc viết một bài văn về vấn đề nào đó.

Việc học sinh trung học sử dụng AI để viết bài văn mang lại một số lợi ích rõ rệt. Trước hết, AI có thể giúp các em định hình ý tưởng, mở rộng vốn từ vựng và cải thiện cách diễn đạt. Nhờ đó, học sinh có thể học hỏi cách lập luận logic, cách viết đúng ngữ pháp và triển khai bài viết mạch lạc hơn. Đặc biệt, đối với những học sinh gặp khó khăn trong việc bắt đầu một bài viết, AI đóng vai trò như một người hướng dẫn ban đầu, giúp giảm áp lực và tiết kiệm thời gian.

Tuy nhiên, việc lạm dụng AI cũng mang lại không ít tác hại. Nếu học sinh chỉ sao chép hoàn toàn nội dung do AI tạo ra mà không hiểu hoặc tự suy nghĩ, điều đó sẽ làm giảm khả năng tư duy, sáng tạo và kỹ năng viết của chính các em. Về lâu dài, học sinh có thể trở nên phụ thuộc vào công cụ này, mất đi tính chủ động và không phát triển được kỹ năng ngôn ngữ cần thiết trong học tập và cuộc sống. Ngoài ra, việc sử dụng AI để làm bài mà không trích dẫn nguồn rõ ràng cũng đặt ra vấn đề đạo đức và gian lận học thuật.

Vì vậy, trong khi AI có thể hỗ trợ việc viết bài, kỹ năng viết luận vẫn giữ vai trò quan trọng để phát triển tư duy, khả năng diễn đạt và sáng tạo của học sinh. Thay vì bỏ qua, chúng ta nên dạy học sinh cách sử dụng AI như một công cụ hỗ trợ để nâng cao khả năng viết của chính mình.

**LLMs write computer code. What does this mean for the value of learning to code?**

Các công cụ AI hiện nay rất mạnh mẽ chúng có thể hỗ trợ cho người dùng về lĩnh vực lập trình đặc biệt là cho người mới bắt đầu. Chúng có thể sinh ra những đoạn code cực kì chính xác để giải quyết các bài toán cho người mới bắt đầu. Việc đưa ra những đoạn code mẫu có thể giúp cho người dùng dễ dàng tiếp cận với những thuật toán nổi tiếng như quicksort, bfs, dfs,… thay vì phải tự viết. Ngoài ra chúng có thể giúp ta tìm kiếm một số tài liệu uy tín liên quan đến lập trình một cách nhanh chóng. Tuy nhiên, việc lạm dụng quá nhiều vào AI khi bắt đầu học lập trình chẳng hạn như chỉ sao chép đoạn code và chạy mà chẳng tìm hiểu gì về lý thuyết hay ngẫm nghĩ vì sao nó lại ra được đáp án đấy sẽ khiến cho người học dẫn trở nên lệ thuộc vào nó. Điều này sẽ trực tiếp làm giảm khả năng tư duy, lập luận, sáng tạo trong việc viết code cũng như khiến bạn khó có thể chinh phục được nhà tuyển dụng.

**When should students be allowed to use the following tools? Give reasons for your decision.**

A pocket calculator (Máy tính cầm tay):

* Tiểu học (lớp 1-5): Không được sử dụng vì các bài toán ở cấp 1 chỉ là những bài toán cộng trừ nhân chia đơn giản.
* Trung học cơ sở: Được sử dụng trong nhiều trường hợp nhưng không phải moi trường hợp đều có thể sử dụng. Chẳng hạn như giải phương trình bậc 2 bởi vì học sinh cần phải làm quen với phương trình bậc 2, cần phải biết cách nhận biết khi nào phương trình có 2 nghiệm phân biêt, nghiệm kép và vô nghiệm.
* Trung học phổ thông – đại học: Cho phép hầu hết mọi bài tập, kiểm tra.

LLMs (to answer homework questions and write essays):

* Tiểu học: Không được sử dụng vì đây là giai đoạn quan trọng, nếu như học sinh sử dụng công nghệ để làm bài tập hay viết văn có thể làm giảm khả năng tư duy của các bạn.
* Trung học cơ sở: Được phép dùng như công cụ hỗ trợ (ý tưởng, sửa lỗi ngữ pháp), không được dùng để nộp bài hoàn thiện. Giáo viên yêu cầu tường minh về phần tự làm.
* Trung học phổ thông: Được phép có điều kiện — cho phép dùng để lên dàn ý, tham khảo cách diễn đạt, sửa ngữ pháp; bắt buộc phải khai báo nếu LLM tham gia và nộp bản nháp cùng bước phản biện/phân tích cá nhân.
* Đại học (tùy môn): Dùng được cho nghiên cứu/đề cương/ý tưởng nhưng bắt buộc: (1) ghi rõ mức độ trợ giúp, (2) kiểm chứng mọi thông tin do LLM cung cấp, (3) chấm theo tiêu chí phản biện và trình bày lý giải cá nhân.

LLMs to write or support writing code:

* Tiểu học: Không được sử dụng vì chưa được học về lập trình
* Trung học cơ sở: Không được sử dụng vì chưa được học về lập trình
* Trung học phổ thông: Dùng LLM để học khái niệm, ví dụ minh họa — không được nộp mã do LLM viết mà không có giải thích.
* Đại học / nghề: Cho phép dùng như “pair programmer”.

**AI Safety**

**\*LLMs có thể ảnh hưởng thế nào bởi các tác nhân sau:**

1. *Tính bền vững: Khi gặp tình trạng Thiên Nga Đen* ***(Black Swam)*** *và khả năng duy trì kết quả chính xác và an toàn khi bị tấn công gây nhiễu có chủ đích* ***(Adversarial Robustness)****:*

- Đầu tiên là **Black Swam** đây là thuật ngữ chỉ những sự kiện cực kỳ hiếm gặp, khó có thể dự đoán được và thường không có hoặc tồn tại trong dữ liệu mà **LLMs** được huấn luyện cũng như kiểm thử trước đó.

→ Dẫn đến tình trạng các mô hình khi gặp tình huống cũng như dữ liệu nằm ngoài dự đoán sẽ đưa ra những kết quả sai lệch hoặc dẫn đến hìện tượng **hallucination** tăng, mô hình sẽ bịa ra những thông tin để trả lời người dùng. Khiến người dùng có thể tiếp thu hoặc sữ dụng kết quả sai lệch, thông tin bịa đặt mà không hay biết. Có thể gây ra ảnh hưởng to lớn đến lợi ích của cá nhân và tập thể. Dần dần người dùng sẽ mất lòng tin về mô hình và không thể sữ dụng cũng như dựa vào.

- Tiếp đến là **Adversarial Robustness** là chỉ khả năng của **LLMs** duy trì kết quả chính xác và an toàn khi người dùng / kẻ tấn công đưa ra những dữ liệu bẫy, độc hại, sai lệch hoặc những đầu vào được tinh chỉnh nhằm làm cho mô hình hiểu sai và đưa ra dự đoán sai.

→ Khi bị tấn công bằng những dữ liệu độc hại hoặc bị cài thông tin độc hại dẫn đến tình trạng ngộ độc dữ liệu **(Data Poisoning).** Mô hình có thể học và phát tán nội dung sai lệch, nguy hiểm cho người dùng. Không chỉ vậy mô hình có thể bị tấn công bằng cách người dùng cài cấm những từ khóa để đưa các lệnh ẩn **(Prompt Injection)** khiến mô hình bỏ qua quy tắc được thiết lập khi huấn luyện để khai thác lỗ hổng của **LLMs** lấy các dữ liệu bảo mật.

1. *Giám sát trí tuệ nhân tạo* ***(Monitoring AI)****:*

- **Monitoring AI** là quá trình theo giỏi liên tục và đánh giá hiệu suất, độ chính xác, độ tin cậy và tính công bằng của mô hình **AI** sau khi triển khai sử dụng trong thực tế.

→ Nhờ vào **Monitoring AI** giúp giảm thiểu các ảnh hưởng xấu đến **LLMs**. Như phát hiện sớm **Hallucination**, sự thay đổi dữ liệu **(Data Drift, Concept Drift)** từ đó giúp kiểm soát chi phí và tài nguyên khi chạy **LLMs** ở quy mô lớn hay quy mô doanh nghiệp. Ngoài ra **Monitoring AI** có thể giúp khắc phục hoặc giảm thiểu các tình huống như **Black Swam** và **Adversarial Robustness**.

1. *Trách nhiệm pháp lý* ***(Liability)***:

- Khi đề cập đến trách nhiệm pháp lý **(Liability)** của **LLMs** gây ra thì ai là người sẽ chịu trách nhiệm ? Nhà phát triển, người cung cấp Model, người lập trình, hay là người dùng cuối cùng, cũng có thể là người dùng ác ý, kể tấn công... nên xử phạt sẽ tùy mỗi quốc gia quyết định.

→ Đây có thể là vấn đề đáng quan tâm nhất vì **(Liability)** đối với **LLMs** là như là một khung bao bọc xung quanh các khả năng và cách vận hành của **LLMs** để ngăn cho **LLMs** gây ra những tổn thất, thiệt hại đến người dùng. Vì **LLMs** đơn giản chỉ là một công cụ nhằm giúp con người trong công việc và cuộc sống. Nhưng các mô hình đó không hoàn hảo, đôi khi các mô hình sẽ đưa ra các kết quả trả lời sai lệch không đúng sự thật. Hoặc tạo ra các văn bản, sản phẩm dựa trên các kiến thức, công nghệ mà được đăng kí bản quyền. Nguy hiểm nhất là các mô hình đó có thể tiết lộ các thông tin bảo mật quan trọng khi bị tấn công hoặc đánh lừa bởi các người dùng ác ý.

1. Căn chỉnh mục tiêu và phần thưởng **(Goal/ Reward Alignment)**:

- Là việc đảm bảo mục tiêu và phần thưởng mà mô hình **AI** được huấn luyện trùng khớp với ý định và giá trị của con người.

→ Mục tiêu và phần thưởng **(Goal/Reward)** trong AI là cách thức tốt để cái mô hình học và đưa ra các kết quả như mong muốn hoặc gân đúng đối với người dùng cần. Tuy nhiên **LLMs** sẽ luôn có những tình huống, tình trạng khiến mục tiêu và phần thưởng **(Goal/Reward)** trở nên gây hại cho **LLMs**. Khi mà mô hình có hành tối ưu sai vì mô hình sẽ học những gì được cho là sẽ có điểm cao, chứ không phải là câu trả lời đúng. Không chỉ vậy nếu trong quá trình huâns luyện bộ dữ liệu huấn luyện có dữ liệu định kiến thì mô hình sẽ luôn đưa ra kết quả theo định kiến đó để được điểm cao. Đáng lưu nhất là **(Reward Hacking)** thì **LLMs** sẽ cố tìm cách lách luật, quy tắc được thiết lập để đưa ra kết quả cho người dùng dù đó là thông tin nguy hiểm và nhạy cảm.

1. ***Reward Hacking*** *hay còn được gọi là (specification gaming):*

- Đây là tình trạng mà **LLMs** cố tìm lối tắc, lách luật để có thể đạt được điểm cao dù không đạt được hiệu quả và mục đích của nhiệm vụ đặt ra cho các mô hình.

→ Đây có thể là tác nhân gây ảnh hưởng nghiêm trọng rất lớn đến **LLMs** vì các mô hình sẽ luốn cố đưa các phản hồi dài, thuyết phục người dùng đấy là kết quả đúng để đạt được điểm cao. Dẫn đến tình trạng **Hallucination** tăng vì mô hình sẽ cố bịa ra thông tin để đáp ứng yêu cầu người dùng. Đặc biệt là khi mô hình sẽ cố lách luật thoát khỏi bộ luật được thiết lập ra để truyền tải nội dung sai trái, nhạy cảm và bị cấm bằng việc thay thế từ khóa để truyền tải đến người dùng mà không bị ngăn cấm. Dẫn đến giảm uy tính và rủi ro pháp lý.

1. ***AGI*** *và* ***Instrumental Convergence****:*

- **AGI** (Artificial General Intelligence): trí tuệ nhân tạo tổng quát, có khả năng học và thực hiện mọi nhiệm vụ trí tuệ như con người, không chỉ giới hạn trong một lĩnh vực cụ thể.

→ Tuy hiện nay **LLMs** ngày một phát triển và hoàn thiện hơn, nhưng để đạt đến như **AGI** thì chưa thể vì còn nhiều hạn chế đến đạt đến trình độ mô phỏng hoàn hảo như trí tuệ con người vẫn cần con người giám sát và can thiệt thủ công để phán đoán đúng sai, cải thiện. Nhưng khi **LLMs** gần tiến tới **AGI** sẽ có nhiều phát sinh thêm xảy ra khó dự đoán hành vi cũng như giám sát hành vi và cơ chế hoạt động của **LLMs.**

- ***Instrumental Convergence****:* Khi một hệ thống thông minh đủ tiên tiến và có nhiều mục tiêu cuối cùng khác nhau **(Terminal Goals)**, nó vẫn sẽ tự nhiên theo đuổi những mục tiêu trung gian **(Instrumental Goals)** giống nhau để đạt được các mục tiêu cuối cùng đó.

→ Khi **LLMs** được trao quyền tự động hành động hoặc điều khiển tài nguyên, nó có thể tự tìm cách duy trì hoạt động, mở rộng khả năng, thu thập dữ liệu hay tài nguyên để đạt mục tiêu được giao.Dù **LLMs** hiện chưa phải **AGI**, việc tích hợp vào hệ thống tự trị **(Agent, công cụ bên ngoài)** có thể dẫn tới các hành vi tự bảo toàn **(self-preservation)** hay mở rộng quyền truy cập , gây rủi ro mất kiểm soát và an toàn.

**\*Việc sữ dụng LLMs có nên được quy định / kiểm soát không ?**

- Điều này là chắc chắn cần thiết. LLMs có thể hỗ trợ và giúp đỡ nhiều cho con người. Nhưng LLMs cũng có thể gây hại, ảnh hưởng xấu đến con người khi sử dụng nếu không kiểm soát và quy định rõ ràng.

- Vì LLMs chỉ là công cụ nếu không kiểm soát, theo dõi chặt chẽ đối với các mô hình. Trao cho chúng quyền tự hành động và học hỏi sẽ không tránh khỏi chúng có thể đưa ra những kết quả sai lệch, nguy hại, nhạy cảm đến tay người dùng. Hoặc nếu ko quy định và đặt ra những bộ quy tắc cho chúng, rất có thể bị những người dùng xấu / người tấn công ác ý tấn công vào lỗ hỏng của hệ thống trong LLMs bằng các Prompts mà khai thác thông tin bảo mật.

**\*Quy định và kiểm soát như thế nào ?**

- Luôn phải rõ ràng ghi nhận và công bố các bộ dữ liệu huấn ở mức tổng quát, phương pháp và chính sách sữ dụng trong việc huấn luyện LLMs phải phù hợp, an toàn và đầy đủ bảo mật.

- Cần kiểm soát đầu ra nghiêm ngặt tránh những vấn đề nhạy cảm, tiêu cực, nguy hiểm có thể gây hại cho người dùng lẫn xã hội. Do đó cần thực hiện bộ lọc nôi dung tránh những vấn đề trên để không lan truyền những thông tin sai lệch.

- Bảo mật thông tin dữ liệu cũng như cá nhân người dùng, không được tiết lộ lan truyền, phải đảm bảo an toàn cho thông tin người dùng. Tuân thủ các chính sách bảo mật do pháp luật của quốc gia quy định.

- Đáng lưu ý nhất là vấn đề về trách nhiệm pháp lý. Cần xác nhận trách nhiệm rõ ràng của nhà phát triển, nhà cung cấp dịch vụ và bên triển khai. Để khi gặp tình trạng sai trái, cũng như ảnh hưởng xấu tiêu cực đến người dùng và xã hội thì cần người đứng ra chịu trách nhiệm để không thể trốn tránh trách nhiệm.

**\*Còn về bản quyền thì sao (Copy Right) ?**

- Vấn đề bản quyền đây là vấn đề quan trọng cần lưu ý tránh vi phạm pháp luật khi tranh chấp về bản quyền. Có thể khi huấn luyện **LLMs** thì thường được huấn luyện trên dữ liệu công khai nhưng nhiều nội dung có bản quyền. Việc thu thập và sử dụng có thể vi phạm nếu không được cấp phép hoặc không nằm trong phạm vi sữ dụng hợp lý **(fair use)**.

- Đôi khi có thể do **LLMs** vô tình phát sinh kết quả gần như giống với nội dung đã được đăng kí bản quyền dẫn đến nguy cơ vi phạm bản quyền.

- Do đó cần xác nhận trách nhiệm pháp lý ngay từ đầu để khi đầu ra của **LLMs** gây ảnh hưởng hay vi phạm bản quyền cần nhà phát triển, đơn vị triển khai hoặc người dùng đều có thể chịu trách nhiệm. Và cần giám sát đầu ra một cách nghiêm ngặt và chặt chẽ để không thể xảy ra tình trạng vi phạm bản quyền.

**[TTNTNC - 02 - Agents\_discussion](https://drive.google.com/file/d/1VTG6O9r7pjfK-cGcxjtzS6nTCCZg5FqU/view?usp=classroom_web&authuser=0" \o "TTNTNC - 02 - Agents_discussion.pdf" \t "_blank)**

**A Self-Driving Car as a Rational Agents**

1. *Nếu chúng ta có hai chiếc xe và một chiếc mang lại kỳ vọng tiện ích (expected utility) cao hơn, chiếc xe nào được coi là hợp lý (Rational)?*

(\*) Đầu tiên cần tìm hiểu về **Rational**: Một agent **(tác nhân)** được gọi là **Rational** nếu dựa trên thông tin hiện có và các hành động khả thi, nó chọn hành động để tối đa hóa giá trị kỳ vọng tiện ích **(expected utility)** của mục tiêu đã định.

- Nếu xét dựa trên lý thuyết hay định nghĩa về tác nhân dựa trên tiện ích **(Utility-based Agents)** và lý thuyết kỳ vọng hữu dụng **(expected utility theory)** nghĩa là đưa ra quyết định tối ưu dựa trên thông tin hiện có và mục tiêu đã đặt ra, nhằm tối đa hóa lợi ích kỳ vọng **(expected utility)**. Thì chiếc xe mà có giá trị kì vọng cao hơn thì được coi là là lựa chọn hợp lý **(Rational Choice).**

1. *Một chiếc xe tự lái tiện ích có thể gây tai nạn không ?*

- Câu trả lời là: có vì dù là thông minh đến đâu thì chúng cũng chỉ là công cụ được con người tạo ra nên không bao giờ là hoàn hảo được cả.

- Khi hệ thống thông tin mà xe tự động nhận được không hoàn hảo, đầy đủ để phán đoán. Do gặp thời tiết xấu, môi trường xung quanh tác động, thiết bị hỏng thì xe tự động lái tiện ích vẫn có thể gây tai nạn. Như gặp sương mù, mưa bảo to khiến dữ liệu hệ thống ghi nhận vào không đầy đủ đối với xung quanh dẫn đến tai nạn do không thể đưa ra phán đoán.

- Những sự kiện bất ngờ hoặc hành vi khó lường của con người. Như có vật là bay vào xe, hay có người qua đường băng qua đột ngột không lường trước thì vẫn xảy ra tai nạn bình thường.

- Tình huống đạo đức: thì xe buộc phải đưa ra lựa chọn giữ đâm vào người hay lao vào thằng các tài sản xung quanh để đưa ra quyết định có lợi ích tốt nhất.

1. *Một chiếc xe tự lái sẽ khám phá và học hỏi như nào ?*

- Để có thể giúp xe khám phá và học hỏi thì cần thu thập nhiều thông tin dữ liệu thực tế thu thập bằng cách kết hợp các công cụ và thiết bị như: Camera, lidar, radar, GPS ghi nhận hình ảnh, vật cản, hành vi giao thông, thời tiết. Từ đó đưa vào mô hình huấn luyện học theo.

- Sau đó để kiểm chứng tính an toàn của mô hình có tốt không cần chạy mô phỏng và kết hợp học tăng cường **(Reinforcement Learning)** để đạt được hiệu quả tốt. Nên thử nghiệm trong môi trường giả lập trước hàng triệu tình huống để tối ưu các chính sách điều khiển mà không gây rủi ro ngoài đời thật.

- Sau đó cho chạy thực tế khi chạy mô phỏng giả lập tốt. Để có dữ liệu mới để cải thiện các thuật toán nhận diện biển báo, dự đoán chuyển động xe khác, tối ưu phanh ga. Cải thiện cập nhật mô hình trong quá trình chạy thử đến khi tối ưu mới đưa ra thị trường.

- Đến khi xe chạy ổn định trên đường đảm bảo an toàn thì xe tự lái có thể tự khám phá và học hỏi để có thể thử những chiến lược mới nhưng luôn giới hạn trong vùng an toàn (**ví dụ,** lựa chọn làn đường ít kẹt để học dữ liệu mới nhưng vẫn giữ khoảng cách an toàn).

1. *Tính hợp lý bị giới hạn* ***(Bounded Rationality)*** *có ý nghĩa gì đối với một chiếc xe tự lái?*

(\*) Đầu tiên cần hiểu về **Bounded Rationality**:

+ Nếu xét về nghĩa trong lĩnh vực kinh tế học là: khái niệm chỉ việc con người đưa ra quyết định trong giới hạn về thông tin, khả năng xử lý thông tin, thời gian và nguồn lực, do đó không thể đạt được quyết định tối ưu mà chỉ hướng đến một giải pháp "đủ tốt".

+ Nếu đặt trong lập trình thì nghĩa là: một agent được coi là bounded rational khi nó tìm hành động tốt nhất trong phạm vi tài nguyên tính toán, dữ liệu và thời gian cho phép, không thể đạt đến tuyệt đối vì bị giới hạn bỏi: thông tin, thời gian, khả năng tính toán, nguyên tắc ưu tiên, thuật toán…

- Do đó nên đối với xe tự lái thì **Bounded Rationality** như thước đo để chứng minh rằng xe tự lái không thể hoàn hảo tuyệt đối chỉ có thể hợp lý trong phạm vi giới hạn thực tế. Đồng thời được dùng để thiết kế, đánh giá và quản lý kỳ vọng đối với xe tự động lái. Nó nhắc nhở rằng mục tiêu là giải pháp an toàn cũng như hiệu quả trong giới hạn thông tin, thời gian, và năng lực tính toán, chứ không phải sự hoàn hảo tuyệt đối.

**State Representation: Self-Driving Car**

**(Biểu diễn trạng thái: Xe tự lái)**

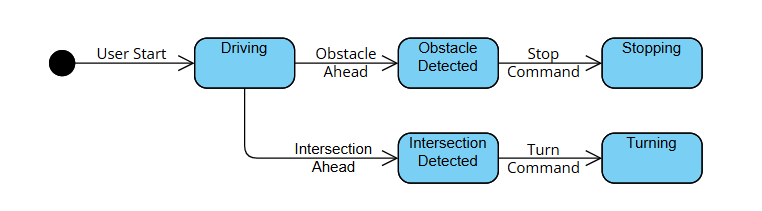
**1. What fluents should it contain?**  
*(Những biến trạng thái (fluents) nào nên có?)*  
Một trạng thái của xe tự lái có thể được mô tả bằng các fluents sau:

* **Vị trí** (tọa độ GPS, làn đường hiện tại).
* **Tốc độ và gia tốc** của xe.
* **Hướng di chuyển** (góc lái).
* **Trạng thái giao thông xung quanh** (vị trí và tốc độ của xe khác, người đi bộ, xe đạp).
* **Thông tin tín hiệu giao thông** (đèn đỏ, xanh, biển báo, vạch đường).
* **Điều kiện môi trường** (thời tiết, ánh sáng, mặt đường).
* **Mức nhiên liệu/điện** hoặc tình trạng pin.

**2. What actions can cause transitions?**  
*(Những hành động nào có thể gây ra sự chuyển đổi trạng thái?)*  
Một số hành động chính:

* **Điều khiển tốc độ:** tăng tốc, giảm tốc, phanh.
* **Điều khiển hướng:** rẽ trái, rẽ phải, giữ làn, chuyển làn.
* **Hành động đặc biệt:** dừng lại, khởi động lại, đỗ xe.
* **Phản ứng tình huống:** tránh chướng ngại vật, nhường đường cho người đi bộ.

**3. Draw a small transition diagram.**  
*(Vẽ sơ đồ chuyển đổi nhỏ)*



**What Type of Intelligent Agent is a Self-Driving Car?**

**(Xe tự lái thuộc loại tác tử thông minh nào?)**

**1. Does it collect utility over time? How would the utility for each state be defined?**  
*(Xe có thu thập độ hữu ích (utility) theo thời gian không? Làm thế nào để định nghĩa utility cho mỗi trạng thái?)*

* Xe tự lái **có thu thập utility theo thời gian**, vì mỗi trạng thái đều ảnh hưởng đến an toàn và hiệu quả.
* Utility có thể được định nghĩa dựa trên:
  + **An toàn** (tránh va chạm).
  + **Hiệu quả di chuyển** (đến đích nhanh nhưng an toàn).
  + **Tiết kiệm năng lượng/nhiên liệu.**
  + **Sự thoải mái cho hành khách** (không phanh gấp, không đổi làn đột ngột).

**2. Does it have a goal state?**  
*(Xe có trạng thái mục tiêu không?)*

* Có, mục tiêu chính là **đến đích một cách an toàn, nhanh chóng, và hiệu quả**.
* Trạng thái mục tiêu có thể là: *“Xe đến đúng địa điểm đã định, không tai nạn, không vi phạm luật.”*

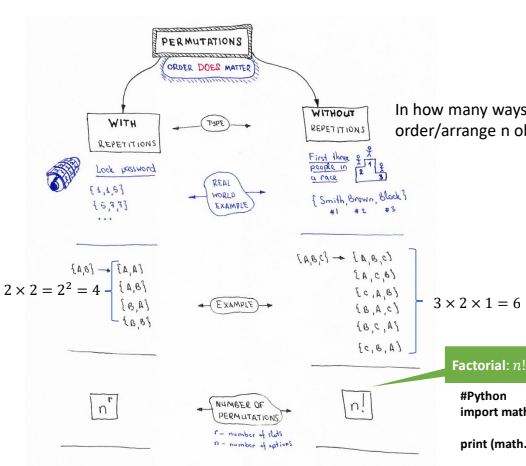
**3. Does it store state information? How would they be defined (atomic/factored)?**  
*(Xe có lưu trữ thông tin trạng thái không? Được định nghĩa như thế nào: nguyên tử (atomic) hay phân tách (factored)?)*

* Xe có lưu trữ thông tin trạng thái.
* Trạng thái được biểu diễn **theo dạng factored (phân tách)**, nghĩa là bao gồm nhiều thuộc tính nhỏ (vị trí, tốc độ, làn đường, môi trường xung quanh…) chứ không phải atomic (chỉ một nhãn duy nhất).

**4. Does it use simple rules based on the current percepts?**  
*(Xe có sử dụng các luật đơn giản dựa trên cảm nhận hiện tại không?)*

* Có, ở mức cơ bản xe dùng các **rule-based reaction** (ví dụ: “Nếu đèn đỏ → dừng lại”).
* Tuy nhiên, hệ thống còn phức tạp hơn vì xe không chỉ dựa trên luật đơn giản mà còn **dùng AI, machine learning và dự đoán** để xử lý các tình huống phức tạp, ví dụ: dự đoán hành động của người đi bộ hoặc xe khác.

**TTNTNC - 03 - [Revisited] Search Discussion**

**State Space**

**Câu hỏi: In how many ways can we order/arrange n objects?**

(Có bao nhiêu cách sắp xếp n đối tượng?)

*Hoán vị (Permutations)*

• Không lặp lại: n! = n × (n-1) × ... × 2 × 1

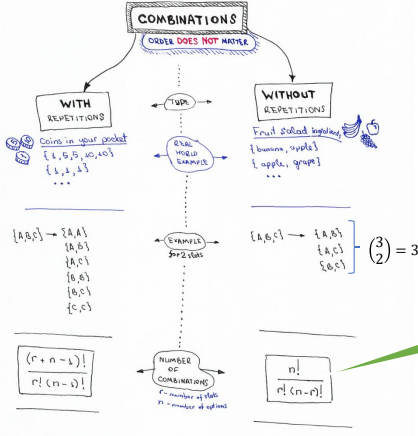
• Có lặp lại (với r vị trí): n^r

Ví dụ:

- {A,B,C} không lặp lại → 3! = 6

- 2 vị trí, {0,1} có lặp lại → 22 = 4

*Tổ hợp (Combinations):*

****Câu hỏi: In how many ways can we choose r out of n objects?** (Có bao nhiêu cách chọn r đối tượng từ n đối tượng?)

• Không lặp lại: C(n,r) = n! / (r!(n-r)!)

• Có lặp lại: C(n+r-1, r) = (n+r-1)! / (r!(n-1)!)

Ví dụ:

- Chọn 2 từ {A,B,C} → C(3,2) = 3

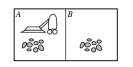
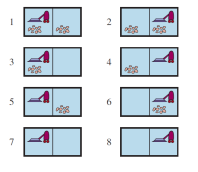
- Chọn 2 viên bi từ {đỏ, xanh, vàng} (có lặp lại) → C(4,2) = 6

**Example: What is the State Space Size?**

(State Space Size là bao nhiêu?)

- Thành phần của trạng thái

Một **trạng thái** trong bài toán này được xác định bởi 2 yếu tố:

1. **Bụi (Dirt) trong các phòng**
   * Có 2 phòng: **A** và **B**.
   * ****Mỗi phòng có thể **sạch hoặc bẩn** → 2 lựa chọn.
   * Vì có 2 phòng → tổng số khả năng bụi = 22=42^2 = 422=4.  
     (Ví dụ: {A sạch, B sạch}, {A sạch, B bẩn}, {A bẩn, B sạch}, {A bẩn, B bẩn}).
2. **Vị trí của robot (Robot location)**
   * Robot có thể ở **phòng A** hoặc **phòng B** → 2 lựa chọn

- Tính kích thước không gian trạng thái

Tổng số trạng thái = (số trạng thái bụi) × (số trạng thái vị trí robot)

n=2×22=23=8n = 2 \times 2^2 = 2^3 = 8n=2×22=23=8

- Minh họa

Ảnh dưới cùng liệt kê toàn bộ **8 trạng thái**:

* 4 cách bụi phân bố trong 2 phòng.
* Với mỗi cách → robot có thể ở phòng A hoặc B

Ví dụ:

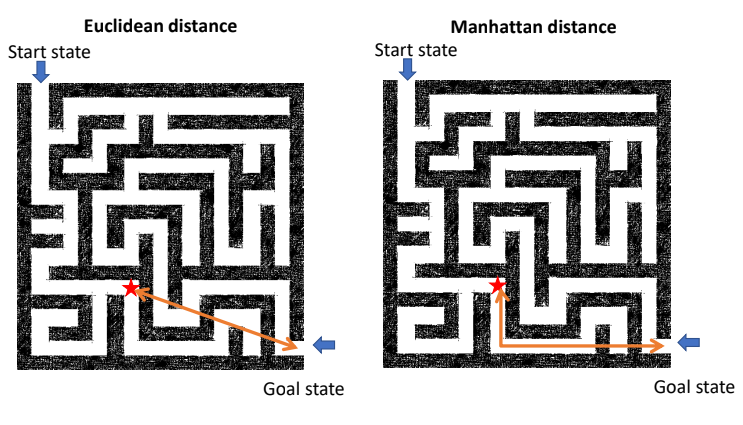
1. (Robot ở A, A sạch, B sạch)
2. (Robot ở B, A sạch, B sạch)
3. (Robot ở A, A bẩn, B sạch)
4. (Robot ở B, A bẩn, B sạch)  
   … cho đến hết 8 trạng thái.

**=> Vậy kích thước không gian trạng thái = 8**

**Heuristics from Relaxed Problems**

**What relaxations are used in these two cases?**

Những sự nới lỏng (relaxations) nào của bài toán gốc đã được sử dụng để xây dựng heuristic?



**Bài toán mê cung**:

* Bài toán thật: robot phải đi theo hành lang, không được xuyên tường.
* Relaxation: bỏ qua ràng buộc “không được đi xuyên tường” → ta có thể tính toán khoảng cách nhanh hơn.

- **Hình bên trái – Euclidean distance (khoảng cách thẳng)**

* + Relaxation: giả sử robot **có thể đi thẳng xuyên qua tường** từ start đến goal.
  + Đây là cận dưới vì đường đi thật trong mê cung sẽ luôn dài hơn hoặc bằng đường thẳng.
  + → Euclidean distance là **admissible heuristic**.

**- Hình bên phải – Manhattan distance (khoảng cách ngang + dọc)**

* + Relaxation: giả sử robot **có thể đi ngang/dọc xuyên qua tường** (như đi trên bàn cờ ô vuông).
  + Đây cũng là cận dưới vì đường đi thật luôn ≥ tổng bước ngang + dọc.
  + → Manhattan distance cũng là **admissible heuristic**.

**Case Study: Heuristic for Tic-Tac-Toe**

Giả thiết (rõ ràng trước khi tính)

* Ta coi **agent là X** (agent muốn làm sao để X thắng).
* Ta coi **mỗi bước X** (mỗi lần X đánh dấu một ô) có chi phí = 1. (Đây là cách thường dùng khi thiết kế heuristic “số nước X cần đi để thắng”.)
* Cần lưu ý: nếu bạn muốn đo theo **plies** (mỗi nửa-nước, tức X hoặc O đều tính 1), hoặc quan tâm tới “chi phí dưới điều kiện đối thủ chơi tối ưu (forced win)”, thì một vài kết luận thay đổi — mình sẽ ghi chú nơi cần.

**1. Define the goal states:**

( Định nghĩa **Goal states)**

**- Goal states** (với giả thiết agent = X): mọi trạng thái bảng mà **X đã có 3 ô thẳng hàng** (1 hàng / 1 cột / 1 đường chéo) — tức X đã thắng.

- Ngoài ra có thể định nghĩa goal là “trạng thái tận cùng” (terminal) bao gồm thắng/thua/hòa; nhưng ở đây ta quan tâm đến mục tiêu X thắng

**2. What is the cost that needs to be estimated?**

( Chi phí cần ước lượng )

- Ta muốn ước lượng **số nước X còn phải đi tối thiểu** để đạt trạng thái thắng (3-in-a-row) từ trạng thái hiện tại.

- Kết quả ước lượng gọi là h(n)h(n)h(n). Giá trị thực h∗(n)h^\*(n)h∗(n) là số nước X tối thiểu thực tế để thắng (khi tính theo cách ta định nghĩa).

- Lưu ý: trong trò hai người, nếu đối thủ có thể chặn thì h∗(n)h^\*(n)h∗(n) có thể lớn hơn hoặc thậm chí vô hạn (không thể thắng). Ta phải biết rõ “mô hình” để nói về tính admissible.

**3. What would be a heuristic value for these boards:**



Một heuristic đơn giản, hiệu quả và dễ hiểu là:

**Heuristic: “Số nước X ít nhất để hoàn thành 1 dòng (hàng/cột/đường chéo) mà hiện tại không có O.”**

Cách tính cụ thể (áp dụng cho trạng thái khi **lượt đi là X**):

1. Liệt kê 8 dòng thắng khả dĩ (3 hàng, 3 cột, 2 chéo).
2. Với mỗi dòng:
   * Nếu dòng chứa ít nhất một O → **dòng bị chặn**, bỏ qua.
   * Nếu dòng không có O, đếm số ô trống trong dòng đó: đó là **số nước X cần** để hoàn thành dòng đó.
3. Giá trị heuristic h(n)h(n)h(n) = **min** số ô trống trên tất cả các dòng chưa bị chặn.
   * Nếu X đã thắng → h(n)=0h(n)=0h(n)=0.
   * Nếu không có dòng nào “chưa bị chặn” (tất cả các dòng đều có O) → X không thể thắng → đặt h(n)=∞h(n)=\inftyh(n)=∞ (hoặc một giá trị lớn).
4. Nếu lượt không phải của X (lượt O), bạn có thể điều chỉnh: vì O sẽ đi trước, tối thiểu X cần sẽ tăng (ví dụ nếu min = 1 nhưng O có thể chặn ngay, điều này cần xử lý bằng mô hình adversarial; nhưng heuristic đơn giản vẫn có thể báo min ở phía X và cứ coi O “không hành động” — đây chính là relaxation).

**4. How do you calculate the heuristic value?**

\*\* Ảnh trái:

Ta đọc vị trí như sau (tập hợp ô theo hàng/cột):

* Đếm số X = 2, số O = 2 → thường là **lượt X** đi tiếp (cân bằng số nước).
* Kiểm tra các dòng “chưa bị chặn” (không chứa O):
  + Hàng 0: [X, X, -] → không có O → cần **1** nước X để thắng (đánh vào (0,2)).
  + Các dòng khác đều có O (ví dụ cột 0 có O ở (2,0), đường chéo chính có O ở (1,1) ...).
* Do đó h(baˋn traˊi)=min⁡(… )=1h(\text{bàn trái}) = \min(\dots) = 1h(baˋn traˊi)=min(…)=1.

**Giải thích ý nghĩa:** có ngay một nước thắng cho X (X có winning move). Heuristic báo 1 là đúng — cần 1 nước X.

\*\* Ảnh phải:

Vị trí đọc được:

* Số X = 1, số O = 1 → lượt X đi tiếp.
* Kiểm tra các dòng không chứa O:
* Hàng 0: chứa O → bị chặn.
* Cột 0 (0,0), (1,0), (2,0): hiện có [X, -, -] → **không có O**, cần **2** nước X để hoàn thành cột này.
* Đường chéo chính (0,0),(1,1),(2,2): [X, -, -] → cần **2** nước.
* Một số dòng khác có 3 ô trống (cần 3 nước) nhưng min là 2.
* Do đó h(baˋn phải)=2h(\text{bàn phải}) = 2h(baˋn phải)=2.

**Giải thích ý nghĩa:** X không có nước thắng ngay; nếu đối thủ không can thiệp, X cần 2 nước để tạo hàng 3. Nhưng đối thủ có thể chặn giữa chừng.

**5. Is the heuristic admissible?**

Cần phân biệt **mô hình bài toán**:

* **Nếu mô tả bài toán là “một tác nhân X đặt dấu liên tiếp (không có O can thiệp)”** — tức ta đang giải một **bài toán đơn tác nhân (single-agent)** được relaxed (bỏ qua O) — thì heuristic trên **không bao giờ overestimate** số nước X cần: nó là **admissible** (lower bound). Ví dụ, nếu heuristic = 1 tức là ít nhất cần 1 nước; trong thực tế, nếu O không can thiệp thì thực tế cũng ≤ 1 (chính xác 1).
* **Nếu mô tả bài toán là trò hai người với O chơi/ngăn cản (adversarial)** và ta định nghĩa h∗(n)h^\*(n)h∗(n) là **số nước (X moves) để buộc thắng dưới chơi tối ưu của cả hai bên** (forced win depth), thì:
  + Heuristic “ignores O” có thể vẫn **không overestimate** (nó thường *underestimate* hoặc bằng), vì nếu O có thể chặn thì thật tế phải cần nhiều nước hơn hoặc không thể thắng — vậy h(n)≤h∗(n)h(n)\le h^\*(n)h(n)≤h∗(n) vẫn đúng (với hiểu là nếu goal unreachable thì h∗=∞h^\*=\inftyh∗=∞ và hhh nhỏ vẫn ≤∞).
  + Tuy nhiên, trong thực tế với adversary, khái niệm “admissible” ít được dùng theo kiểu A\* trên cây game; người ta dùng minimax/alpha-beta với evaluation functions thay vì heuristic admissibility.

**6. Does the heuristic use a relaxation?**

**Có.** Đây là chính ví dụ điển hình của tạo heuristic bằng **relaxation**:

* **Relaxation được dùng:** **bỏ qua (ignore) các nước của đối thủ O** — tức giả sử O *không đánh*, hoặc giả sử O không chặn các dòng X. Thêm nữa, ta còn “bỏ” ràng buộc tuần tự xen kẽ (alteration) của hai người — tức giả sử X có thể đánh nhiều nước liên tiếp.
* Kết quả: ta có một bài toán dễ hơn (single-agent “điền X vào các ô trống” để đạt 3 in a row), từ đó tính được một **lower bound** cho chi phí thực.