HỆ THỐNG LOGIC
(LOGICAL AGENT)

Dẫn nhập

Các hệ thống tìm kiếm đã học

– Đều có dạng lập luận từ tri thức đã biết

- Vd, lập luận để tìm đường đi có chi phí thấp nhất từ trạng thái S đến trạng thái G từ tri thức đã biết là: từ một trạng thái có thể đi đến những trạng thái nào với hành động nào và chi phí bao nhiêu, …

– Tri thức của hệ-thống-tìm-kiếm chuyên biệt cho nhóm bài toán tìm kiếm và cố định (không thêm tri thức mới vào tri thức đã có)

Dẫn nhập

Hôm nay: hệ thống logic (logical agent)

– Lập luận từ tri thức đã biết

– Dùng ngôn ngữ logic để biểu diễn tri thức

- Cho phép biểu diễn các tri thức nói chung
- Cho phép tích hợp tri thức mới vào tri thức đã biết
- Cho phép lập luận từ tri thức đã biết

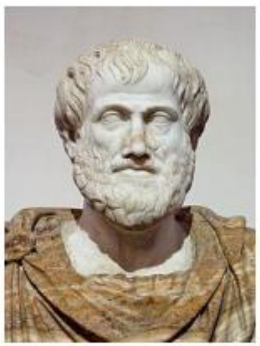
Logic là gì?

- **Logic** bao gồm các nghiên cứu có hệ thống về hình thức tranh luận
  - Một tranh luận được gọi là hợp lệ khi giả thuyết và kết luận liên hệ với nhau dựa trên nền tảng các khái niệm logic

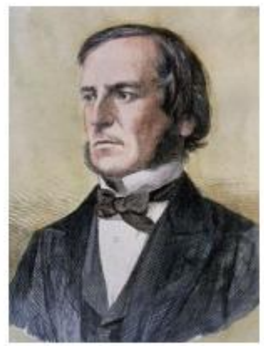
- Một số chủ đề chính trong logic
  - Phân loại tranh luận
  - Trình bày các tranh luận hợp lệ theo chuẩn hình thức logic chung
  - Nghiên cứu về suy diễn (và ngụy biện)
  - Nghiên cứu về ngữ nghĩa (và nghịch lý)

Quá trình nghiên cứu logic

- Logic được nghiên cứu trong triết học (từ thời cổ đại) và toán học (từ những năm 1800)



Aristole
(384 BC – 322 BC)



George Boole
(1815 – 1864)

Gottlob Frege
(1848 – 1925)

David Hillbert
(1862 - 1943)

- Gần đây, logic được nghiên cứu trong khoa học máy tính, ngôn ngữ học, tâm lý học và nhiều lĩnh vực khác

L, T, và John là 3 người khác nhau. L luôn luôn nói dối, T luôn luôn nói thật.

L cho biết vào ngày thứ 7:

- "John đi làm"
- "John không đọc báo. Anh ta có nấu ăn"

T cho biết vào ngày thứ 7:

- Khi John không làm việc, anh ta cũng không xem Tivi
- John đọc báo, xem Tivi hay nấu ăn

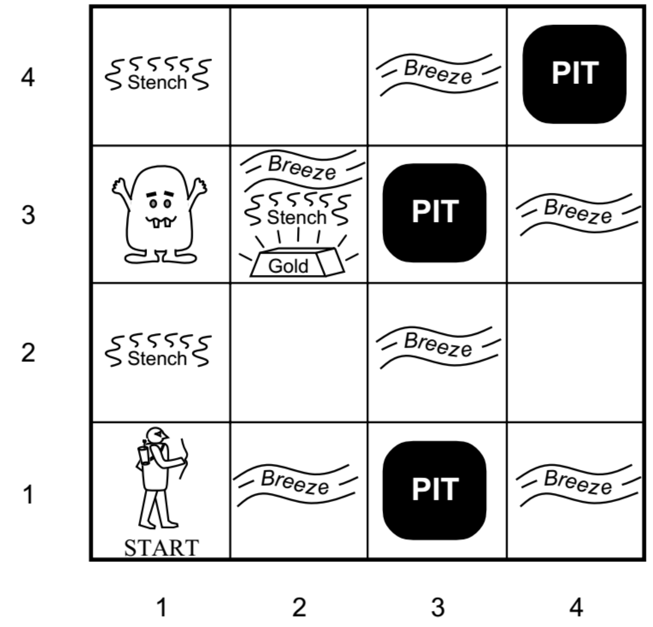
Xác định xem John làm gì vào ngày thứ 7.

Các thành phần chính của hệ thống logic

- **Kho tri thức KB (knowledge base)**
  – KB là một tập các câu biểu diễn tri thức về thế giới; các câu này được viết bằng ngôn ngữ logic
  – Hệ thống có thể thêm tri thức mới vào KB
- **Cơ chế suy diễn (inference mechanism)**
  – Dùng để suy diễn ra các câu mới từ KB
  – Hệ thống dùng các kết quả suy diễn này để quyết định sẽ thực hiện hành động nào

Ví dụ: hệ thống logic trong thế giới wumpus

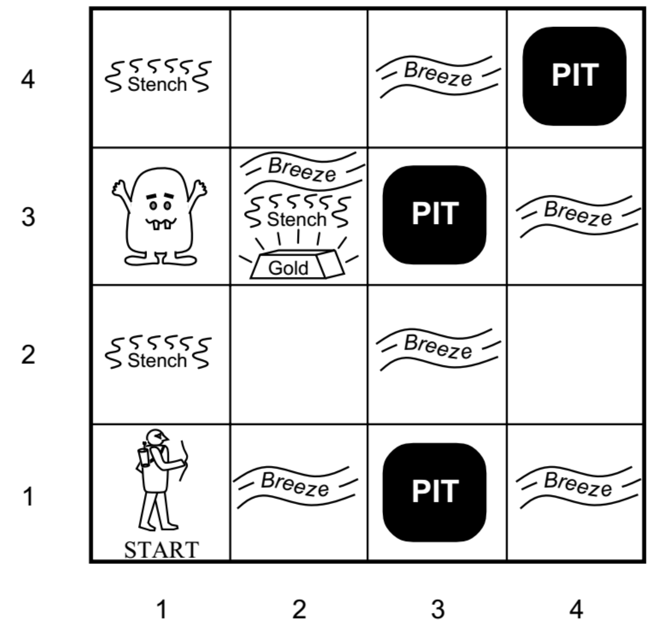
- Gồm 4×4 phòng
- Người chơi (hệ thống logic) luôn bắt đầu ở phòng [1, 1], quay mặt về hướng phải
- Người chơi có thể: đi thẳng, quay trái 90^0, quay phải 90^0, nhặt đồ vật, bắn tên (tên đi thẳng theo hướng quay mặt của người chơi, và sẽ trúng tường hoặc sẽ trúng và giết wumpus; người chơi chỉ có một mũi tên)
- Thực hiện hành động: -1 điểm, dùng tên: -10 điểm
- Nếu người chơi đi vào phòng có wumpus hoặc có hố sâu (PIT) → hy sinh: -1000 điểm
- Nếu người chơi nhặt vàng → thắng: +1000 điểm



Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus

- …
- Người chơi không có được thông tin của tất cả các phòng, mà chỉ có được thông tin của phòng mình đang đứng (và những phòng mình đã đi qua) thông qua các giác quan:
  - Nếu phòng mà người chơi đang đứng gần phòng của Wumpus (chéo không coi là gần) thì người chơi sẽ ngửi thấy mùi hôi (stench)
  - Nếu phòng mà người chơi đang đứng gần phòng có hố sâu (chéo không coi là gần) thì người chơi sẽ cảm thấy gió nhẹ (breeze)
  - Nếu phòng mà người chơi đang đứng có vàng thì người chơi sẽ nhìn thấy ánh vàng (glitter)
  - Nếu người chơi đụng vào tường thì sẽ cảm thấy đau (bump)
  - Nếu wumpus bị giết thì người chơi sẽ nghe được tiếng thét (scream) của nó dù ở bất kỳ phòng nào

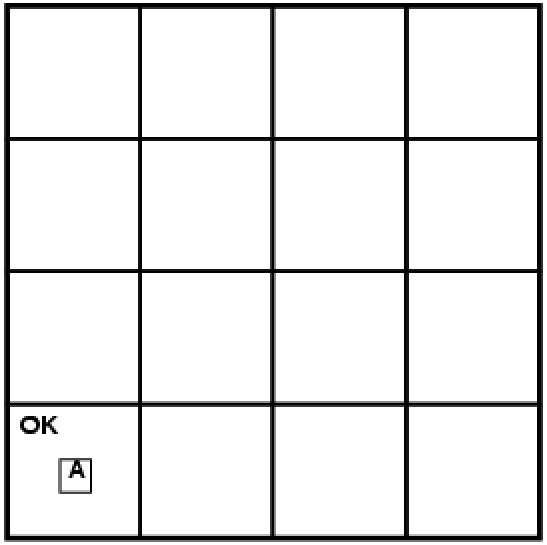


Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus

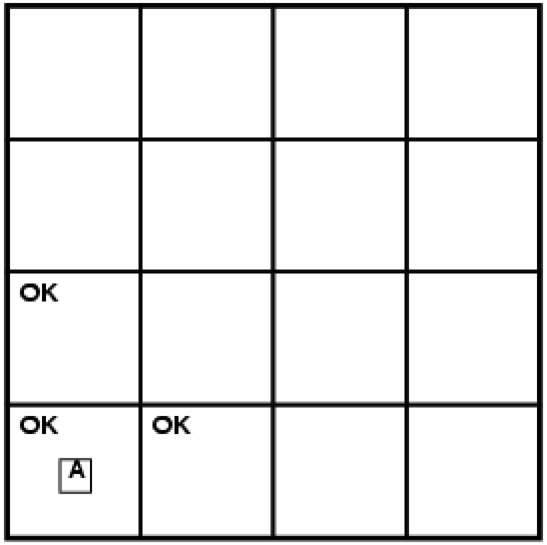
- Ban đầu, kho tri thức KB của hệ thống bao gồm các luật chơi của game
  - Một trong số đó là hệ thống đang ở phòng [1, 1] và phòng [1, 1] an toàn
- Ta hãy xem hệ thống logic sẽ hoạt động như thế nào trong thế giới wumpus …

Ví dụ: hệ thống logic trong thế giới wumpus



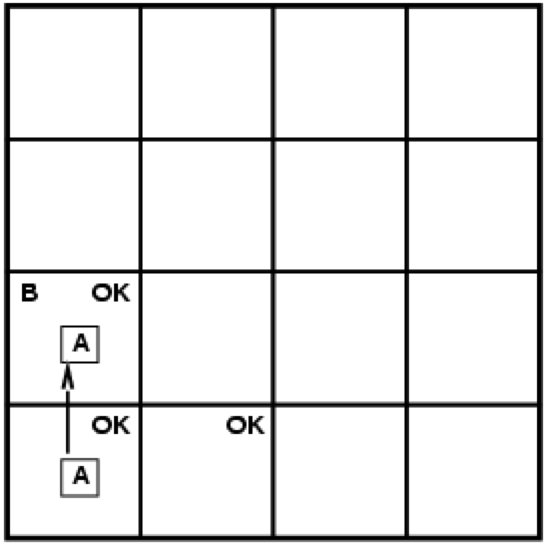
Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus



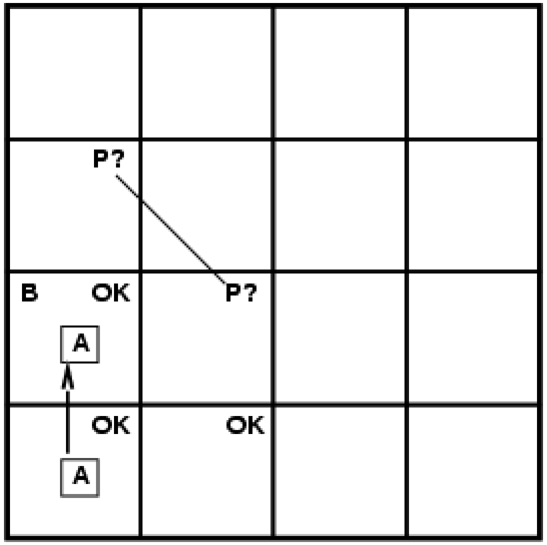
Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus



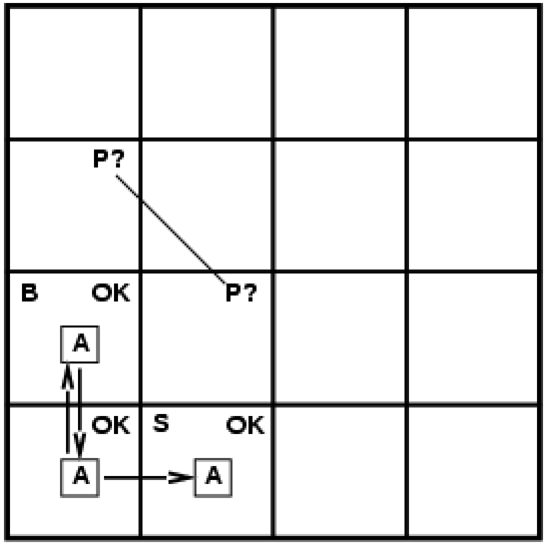
Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus



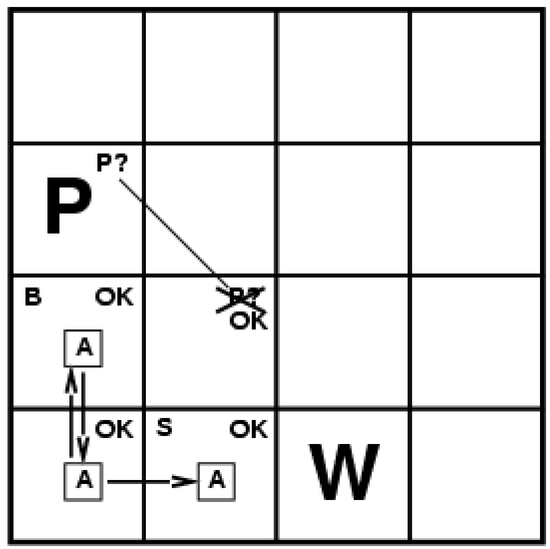
Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus

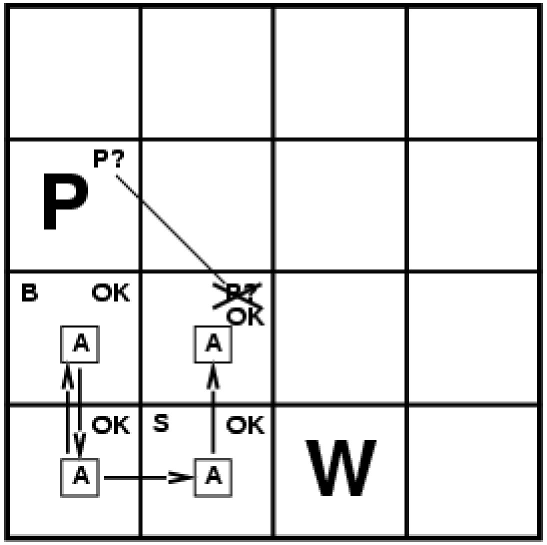


Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus

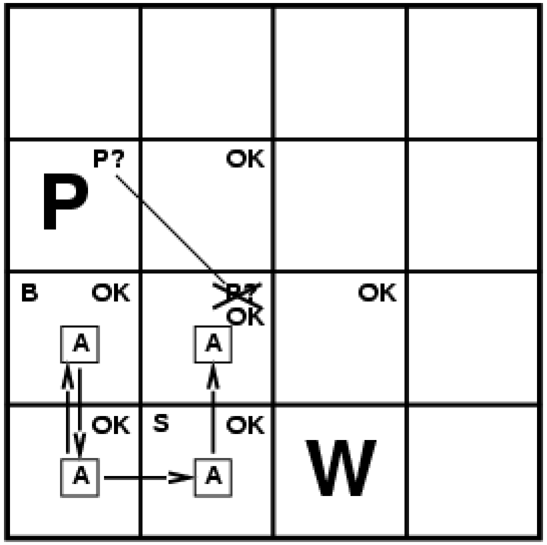


Ví dụ: hệ thống logic trong thế giới wumpus



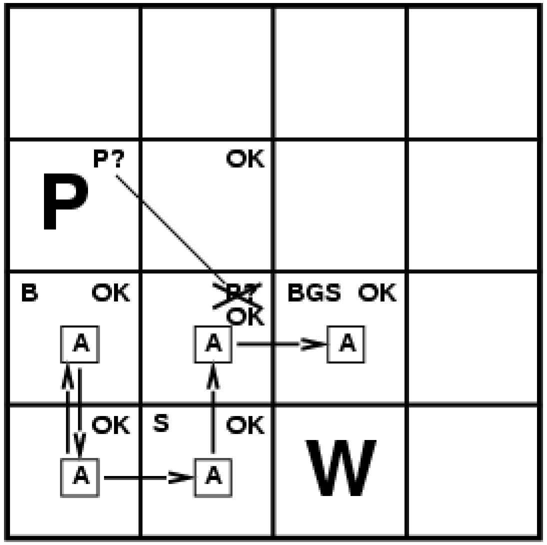
Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus



Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Ví dụ: hệ thống logic trong thế giới wumpus



Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Logic mệnh đề (propositional logic)

Cụ thể thì biểu diễn tri thức như thế nào?

Cụ thể thì suy diễn từ tri thức đã có như thế nào?

Tiếp theo, ta sẽ tìm hiểu về **logic mệnh đề** - một ngôn ngữ đơn giản dùng để biểu diễn tri thức

– Đầu tiên, ta sẽ nói về cú pháp và ngữ nghĩa của logic mệnh đề
– Kế đến, ta sẽ dùng logic mệnh đề để biễu diễn kho tri thức KB
– Cuối cùng, ta sẽ nói về cách suy diễn từ kho tri thức KB trong logic mệnh đề

Logic mệnh đề: Cú pháp

- **Mệnh đề** (proposition) là câu tường thuật có ý nghĩa hoặc đúng (true) hoặc sai (false)
  - Ví dụ: Hôm qua trời có mưa.
- Biến mệnh đề, vd: P, Q, a, b ..., được gọi là câu nguyên tử
- **true** và **false** là các câu hằng
- Giả sử α và β là câu. $\alpha \wedge \beta, \alpha \vee \beta, \alpha \Rightarrow \beta, \alpha \Leftrightarrow \beta, \ldots$ cũng là câu
- **Literal** là câu nguyên tử hoặc phủ định của câu nguyên tử
  - Ví dụ, α và $\neg\alpha$

Logic mệnh đề: cú pháp

(1) Các câu cơ bản:

- **Câu cơ bản** gồm một **biến mệnh đề**; biến mệnh đề ứng với một mệnh đề mà có thể true hoặc false (vd: có thể gọi $W_{1,1}$ là biến mệnh đề ứng với mệnh đề "có wumpus ở ô [1, 1]")
- Số lượng biến mệnh đề và tên biến mệnh đề là tùy ý người dùng

(2) Nếu α và β là câu thì những cái sau đây cũng là câu:

- Phủ định: $\neg\alpha$
- Giao / nối-liền: $\alpha \land \beta$
- Hội / nối-rời: $\alpha \lor \beta$
- Kéo theo: $\alpha \Rightarrow \beta$
- Tương đương: $\alpha \Leftrightarrow \beta$

Từ **(1)** và **(2)**, ta có thể tạo ra rất nhiều câu

Logic mệnh đề: cú pháp

Giả sử có 3 biến mệnh đề là A, B, C. Những cái nào sau đây là câu:

- ☐ A
- ☐ $\neg A$
- ☐ $\neg B \Rightarrow C$
- ☐ $A \wedge (\neg B \Rightarrow C)$
- ☐ $\left(A \wedge (\neg B \Rightarrow C)\right) \Leftrightarrow A$
- ☐ $A \neg B$
- ☐ $A + B$

Thứ tự ưu tiên của các toán tử trong logic mệnh đề (từ ưu tiên nhất cho đến ít ưu tiên nhất): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

Để rõ ràng, có thể dùng dấu đóng mở ngoặc

Logic mệnh đề: cú pháp

Giả sử có 3 biến mệnh đề là A, B, C. Những cái nào sau đây là câu:

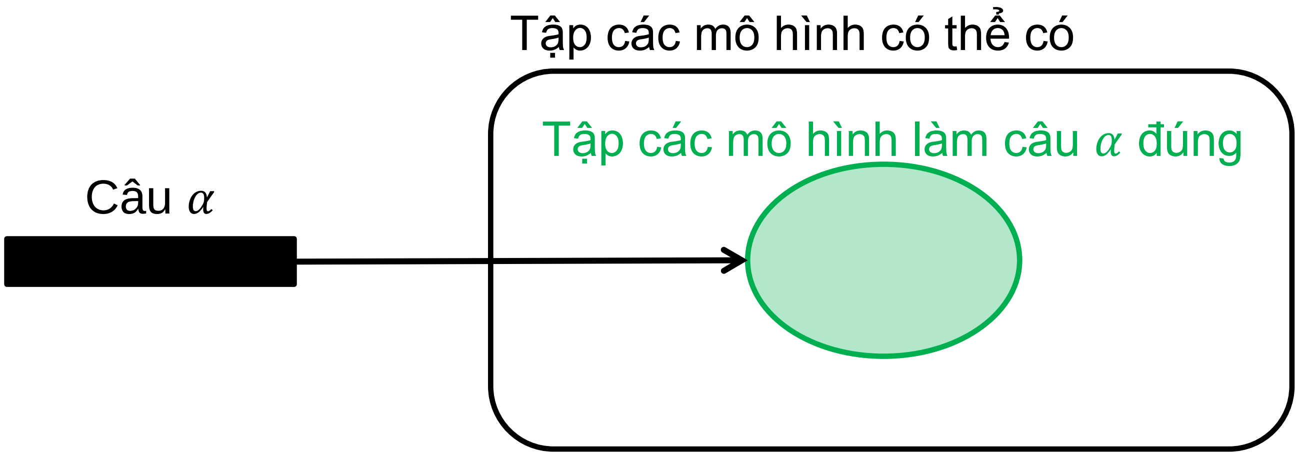
- ☑ A
- ☑ $\neg A$
- ☑ $\neg B \Rightarrow C$
- ☑ $A \wedge (\neg B \Rightarrow C)$
- ☑ $\left(A \wedge (\neg B \Rightarrow C)\right) \Leftrightarrow A$
- ☐ $A \neg B$
- ☐ $A + B$

Thứ tự ưu tiên của các toán tử trong logic mệnh đề (từ ưu tiên nhất cho đến ít ưu tiên nhất): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

Để rõ ràng, có thể dùng dấu đóng mở ngoặc

Logic mệnh đề: ngữ nghĩa

- Một **mô hình** m là một cách gán các giá trị true/false (1/0) cho các biến mệnh đề
  - Vd, với 3 biến mệnh đề A, B, C thì sẽ có tất cả $2^3 = 8$ mô hình có thể có: $\{A = 0,\ B = 0,\ C = 0\}, \{A = 0,\ B = 0,\ C = 1\}, \{A = 0,\ B = 1,\ C = 0\}$, ...
- **Ngữ nghĩa** cho biết câu sẽ có giá trị true/false với mỗi mô hình.



Logic mệnh đề: ngữ nghĩa

- Một **mô hình** m là một cách gán các giá trị true/false (1/0) cho các biến mệnh đề
  - Vd, với 3 biến mệnh đề A, B, C thì sẽ có tất cả $2^3 = 8$ mô hình có thể có: $\{A = 0,\ B = 0,\ C = 0\}, \{A = 0,\ B = 0,\ C = 1\}, \{A = 0,\ B = 1,\ C = 0\}$, ...
- **Ngữ nghĩa** cho biết câu sẽ có giá trị true/false với mỗi mô hình. Cụ thể hơn, ngữ nghĩa cung cấp cho ta **hàm dịch (interpretation function)** $\boldsymbol{I}$:
  - Input: câu α và mô hình m
  - Output:
    - True (ta nói: m thỏa α, hay m là mô hình của α)
    - False (ta nói: m <u>không</u> thỏa α, hay m <u>không</u> là mô hình của α)

Logic mệnh đề: ngữ nghĩa

Hàm dịch (interpretation function) I

- **Input**: câu α và mô hình m; **Output**: true/false (1/0)
- **Định nghĩa hàm**:
  - Nếu α là câu cơ bản (chỉ gồm một biến mệnh đề): $I(\alpha, m)$ bằng giá trị của biến mệnh đề của câu α trong mô hình m
  - Nếu α là câu phức tạp: $I(\alpha, m)$ được tính từ I của các câu đơn giản hơn một cách đệ qui theo luật sau

Cho β và γ là hai câu bất kỳ

$\boldsymbol{I(\beta, m)}$	$\boldsymbol{I(\gamma, m)}$	$\boldsymbol{I(\neg\beta, m)}$	$\boldsymbol{I(\beta \wedge \gamma, m)}$	$\boldsymbol{I(\beta \vee \gamma, m)}$	$\boldsymbol{I(\beta \Rightarrow \gamma, m)}$	$\boldsymbol{I(\beta \Leftrightarrow \gamma, m)}$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

– Câu α: $(\neg A \wedge B) \Leftrightarrow C$

– Mô hình m: $\{A{:}\,1, B{:}\,1, C{:}\,0\}$

– Dịch:

$$\boxed{I\big((\neg A \wedge B) \Leftrightarrow C, m\big) = ?}$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

– Câu α: $(\neg A \wedge B) \Leftrightarrow C$

– Mô hình m: $\{A{:}\,1, B{:}\,1, C{:}\,0\}$

– Dịch:

$$I\big((\neg A \wedge B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \wedge B), m\big) = ?$$

$$I(C, m) = ?$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

- Câu α: $(\neg A \land B) \Leftrightarrow C$
- Mô hình m: $\{A: 1, B: 1, C: 0\}$
- Dịch:

$$I\big((\neg A \land B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \land B), m\big) = ?$$

$$I(C, m) = 0$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

- Câu α: $(\neg A \wedge B) \Leftrightarrow C$
- Mô hình m: $\{A: 1, B: 1, C: 0\}$
- Dịch:

$$I\big((\neg A \wedge B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \wedge B), m\big) = ? \qquad I(C, m) = 0$$

$$I(\neg A, m) = ? \qquad I(B, m) = ?$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) *I*

- Câu α: $(\neg A \land B) \Leftrightarrow C$
- Mô hình m: $\{A{:}\,1, B{:}\,1, C{:}\,0\}$
- Dịch:

$$I\big((\neg A \land B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \land B), m\big) = ?$$

$$I(C, m) = 0$$

$$I(\neg A, m) = ?$$

$$I(B, m) = 1$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) *I*

- Câu α: $(\neg A \land B) \Leftrightarrow C$
- Mô hình m: $\{A: 1, B: 1, C: 0\}$
- Dịch:

$$I\big((\neg A \land B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \land B), m\big) = ?$$

$$I(C, m) = 0$$

$$I(\neg A, m) = ?$$

$$I(B, m) = 1$$

$$I(A, m) = ?$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

– Câu α: $(\neg A \land B) \Leftrightarrow C$

– Mô hình m: $\{A: 1, B: 1, C: 0\}$

– Dịch:

$$I\big((\neg A \land B) \Leftrightarrow C, m\big) =?$$

$$I\big((\neg A \land B), m\big) =?$$

$$I(C, m) = 0$$

$$I(\neg A, m) =?$$

$$I(B, m) = 1$$

$$I(A, m) = 1$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) I

- Câu α: $(\neg A \wedge B) \Leftrightarrow C$
- Mô hình m: $\{A{:}\,1, B{:}\,1, C{:}\,0\}$
- Dịch:

$$I\big((\neg A \wedge B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \wedge B), m\big) = ?$$

$$I(C, m) = 0$$

$$I(\neg A, m) = 0$$

$$I(B, m) = 1$$

$$I(A, m) = 1$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) *I*

- Câu α: $(\neg A \land B) \Leftrightarrow C$
- Mô hình m: $\{A: 1, B: 1, C: 0\}$
- Dịch:

$$I\big((\neg A \land B) \Leftrightarrow C, m\big) = ?$$

$$I\big((\neg A \land B), m\big) = 0 \qquad I(C, m) = 0$$

$$I(\neg A, m) = 0 \qquad I(B, m) = 1$$

$$I(A, m) = 1$$

Logic mệnh đề: ngữ nghĩa

Ví dụ về hàm dịch (interpretation function) *I*

- Câu α: $(\neg A \wedge B) \Leftrightarrow C$
- Mô hình m: $\{A: 1, B: 1, C: 0\}$
- Dịch:

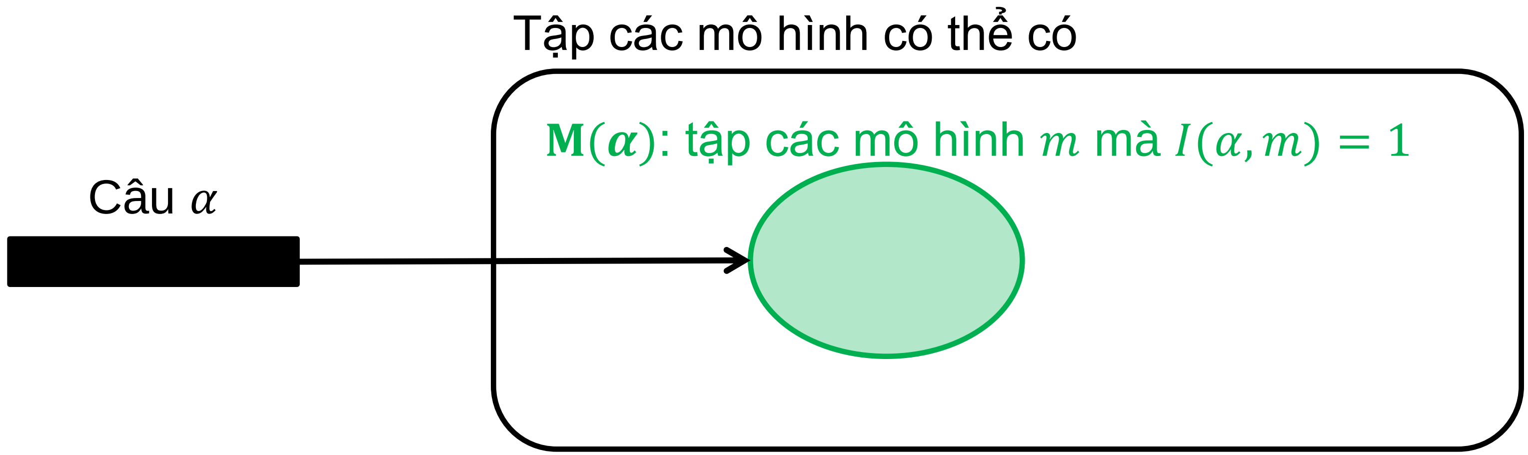
$$I\big((\neg A \wedge B) \Leftrightarrow C, m\big) = 1$$

$$I\big((\neg A \wedge B), m\big) = 0 \qquad I(C, m) = 0$$

$$I(\neg A, m) = 0 \qquad I(B, m) = 1$$

$$I(A, m) = 1$$

Logic mệnh đề: ngữ nghĩa

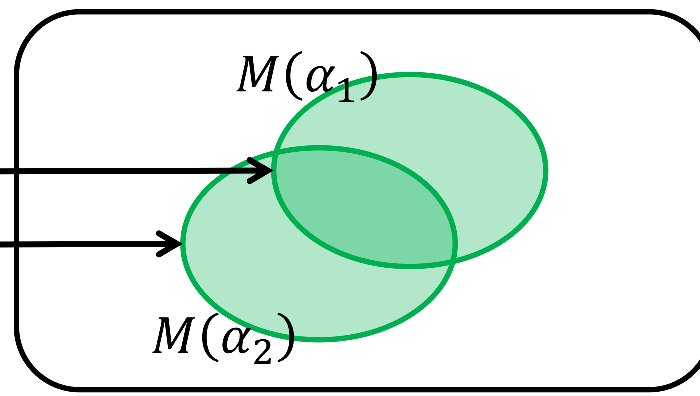


Kho tri thức KB (knowledge base)

- Một **kho tri thức $\boldsymbol{KB}$** là một tập các câu $\{\alpha_1, \alpha_2, \ldots\}$
- Có thể coi KB là một câu: $KB = \alpha_1 \land \alpha_2 \land \cdots$
- $M(KB) = M(\alpha_1) \cap M(\alpha_2) \cap \cdots$
- Ý nghĩa của KB: KB cho biết các ràng buộc về thế giới; $M(KB)$ là tập các thế giới (các mô hình) thỏa các ràng buộc này

$P_{1,1}$: có hố ở ô [1, 1]
$W_{1,1}$: có wumpus ở ô [1, 1]

Câu α_1: $\neg P_{1,1}$ [redacted]
Câu α_2: $\neg W_{1,1}$ [redacted]



L, T, và John là 3 người khác nhau. L luôn luôn nói dối, T luôn luôn nói thật.

L cho biết vào ngày thứ 7:

- "John đi làm"
- "John không đọc báo. Anh ta có nấu ăn"

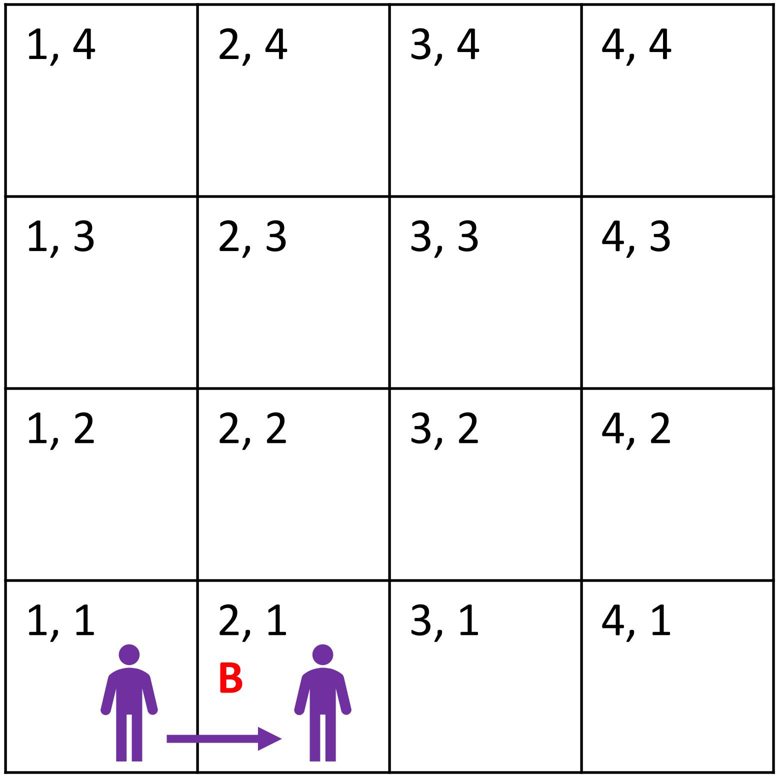
T cho biết vào ngày thứ 7:

- Khi John không làm việc, anh ta cũng không xem Tivi
- John đọc báo, xem Tivi hay nấu ăn

Xác định xem John làm gì vào ngày thứ 7.

Kho tri thức KB (knowledge base)

Thử xây dựng KB của hệ thống logic trong thế giới wumpus



Đơn giản hóa:

- Giả sử chỉ có hố, không có wumpus
- KB chỉ gồm những câu liên quan đến ngữ cảnh đang xét ở hình bên

Các biến mệnh đề: với mỗi ô $[i, j]$

- P_{ij}: "có hố ở ô $[i, j]$"
- B_{ij} : "có gió ở ô $[i, j]$"

Kho tri thức KB (knowledge base)

Thử xây dựng KB của hệ thống logic trong thế giới wumpus

- Luật chơi: "Không có hố ở ô [1, 1]"

$$\alpha_1 : \neg P_{1,1}$$

- Luật chơi: "Một ô có gió nếu và chỉ nếu có ít nhất một ô lân cận có hố"

$$\alpha_2 : B_{1,1} \Leftrightarrow \left(P_{1,2} \vee P_{2,1}\right)$$

$$\alpha_3 : B_{2,1} \Leftrightarrow \left(P_{1,1} \vee P_{2,2} \vee P_{3,1}\right)$$

- Thông tin mới: "Không có gió ở ô [1, 1]"

$$\alpha_4 : \neg B_{1,1}$$

- Thông tin mới: "Có gió ở ô [2, 1]"

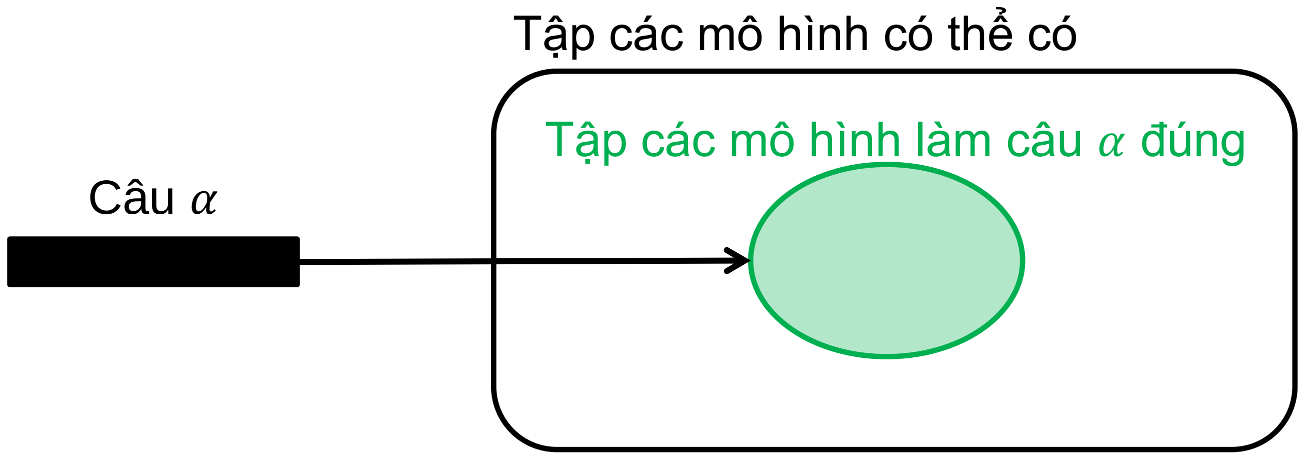
$$\alpha_5 : B_{2,1}$$

$KB = \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5$

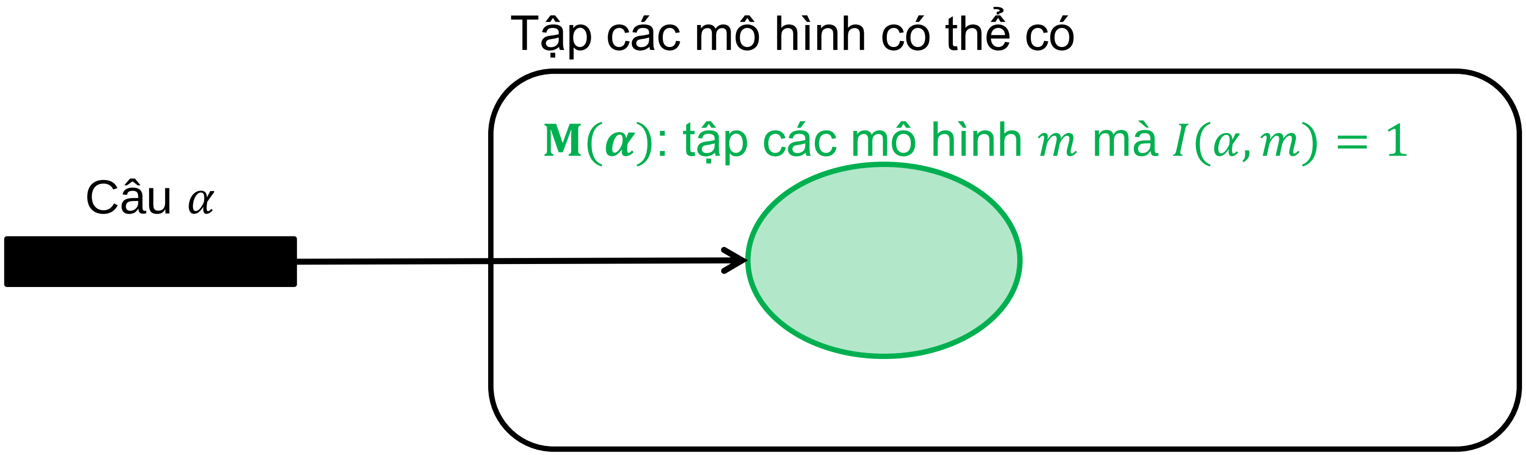
1, 4	2, 4	3, 4	4, 4
1, 3	2, 3	3, 3	4, 3
1, 2	2, 2	3, 2	4, 2
1, 1	2, 1 B	3, 1	4, 1

Logic mệnh đề: ngữ nghĩa

- Một **mô hình** m là một cách gán các giá trị true/false (1/0) cho các biến mệnh đề
  - Vd, với 3 biến mệnh đề A, B, C thì sẽ có tất cả $2^3 = 8$ mô hình có thể có: $\{A = 0,\ B = 0,\ C = 0\}, \{A = 0,\ B = 0,\ C = 1\}, \{A = 0,\ B = 1,\ C = 0\}$, ...
- **Ngữ nghĩa** cho biết câu sẽ có giá trị true/false với mỗi mô hình.



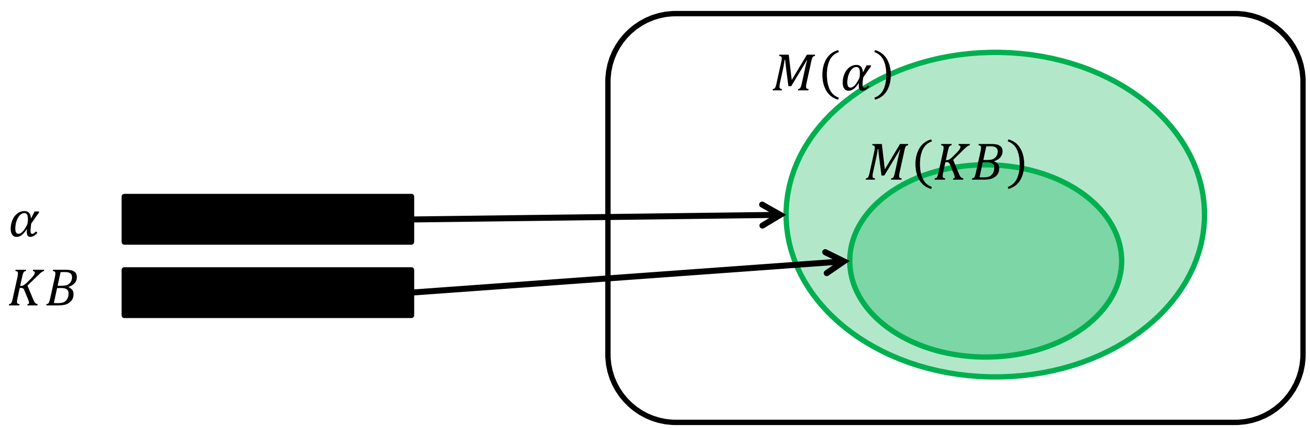
Logic mệnh đề: ngữ nghĩa



KB suy dẫn (entail) câu α

KB suy dẫn câu α (ký hiệu: $\boldsymbol{KB \vDash \alpha}$) nếu và chỉ nếu $M(\alpha) \supseteq M(KB)$; nói một cách khác: trong mọi thế giới (mọi mô hình) mà KB đúng thì α cũng sẽ đúng

– Vd: $Rain \land Snow \vDash Snow$



Liệt kê và kiểm tra thể hiện: Ví dụ

- Giả sử biết rằng
  - Nếu hôm nay trời nắng, thì Tomas sẽ vui vẻ

    Nếu Tomas vui vẻ, bài giảng sẽ tốt
  - Hôm nay trời nắng

Cơ sở tri thức (Knowledge base, KB)

- Có thể kết luận rằng **bài giảng sẽ tốt**?

- Biểu diễn bằng logic mệnh đề
  - Gọi S = Hôm nay trời nắng, H = Tomas vui vẻ, G = Bài giảng tốt
  - $(S \Rightarrow H), (H \Rightarrow G), (S) \Rightarrow (G)$?

Liệt kê và kiểm tra thể hiện: Ví dụ

S	H	G	$S \Rightarrow H$	$H \Rightarrow G$	S	G
t	t	t	t	t	t	t
t	t	f	t	f	t	f
t	f	t	f	t	t	t
t	f	f	f	t	t	f
f	t	t	t	t	f	t
f	t	f	t	f	f	f
f	f	t	t	t	f	t
f	f	f	t	t	f	f

- 3 biến, 8 thể hiện có thể có. Trong đó, chỉ có một thể hiện thỏa tất cả câu trong cơ sở tri thức và G cũng đúng trong thể hiện đó: $S = \boldsymbol{t}$, $H = \boldsymbol{t}$, $G = \boldsymbol{t}$. Vậy, **bài giảng sẽ tốt**.

Ví dụ

- Cho các luật và sự kiện như sau
  - IF nóng AND có khói THEN có lửa
  - IF chuông báo cháy reo THEN có khói
  - IF có lửa THEN bật vòi phun nước cứu hỏa

  Luật

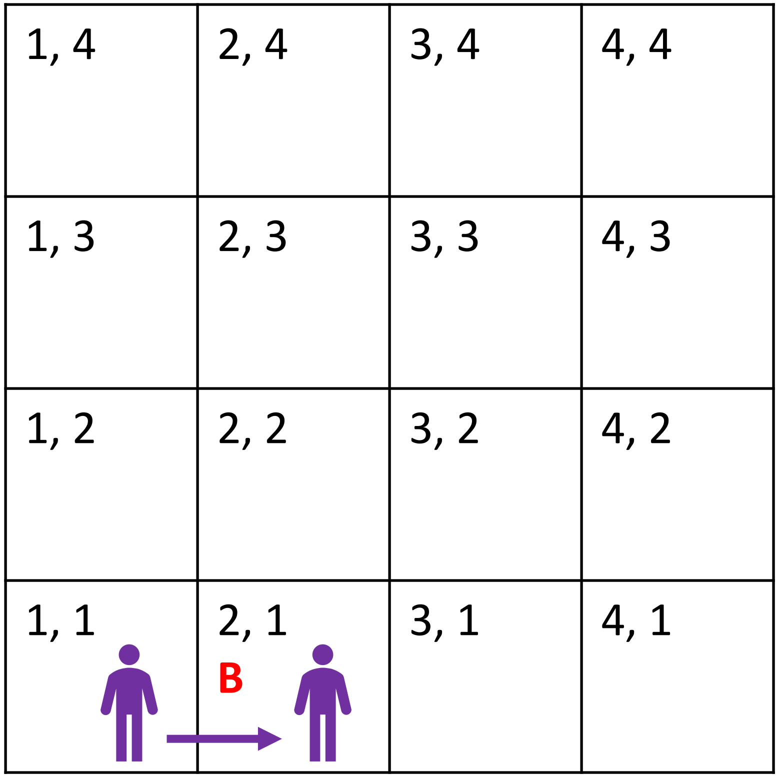
  - chuông báo cháy reo
  - nóng

  Sự kiện
- Chứng minh hành động **bật vòi phun nước cứu hỏa** xảy ra

Kiểm $KB \vDash \alpha$

Trong hệ thống logic, ta sẽ có nhu cầu kiểm tra xem một câu α nào đó có được suy dẫn từ KB không?

– Trong ví dụ wumpus, ta sẽ muốn biết $KB \vDash \neg P_{1,2}$? $KB \vDash \neg P_{2,2}$? $KB \vDash \neg P_{3,1}$?



Kiểm $KB \vDash \alpha$: bằng cách kiểm tra tất cả các mô hình

Quay lại với KB đã xây dựng ở ví dụ wumpus

Kho tri thức KB (knowledge base)

Thử xây dựng KB của hệ thống logic trong thế giới wumpus

- Luật chơi: “Không có hố ở ô [1, 1]”

$$\alpha_1 :\ \neg P_{1,1}$$

- Luật chơi: “Một ô có gió nếu và chỉ nếu có ít nhất một ô lân cận có hố”

$$\alpha_2 :\ B_{1,1} \Leftrightarrow \left(P_{1,2} \lor P_{2,1}\right)$$

$$\alpha_3 :\ B_{2,1} \Leftrightarrow \left(P_{1,1} \lor P_{2,2} \lor P_{3,1}\right)$$

- Thông tin mới: “Không có gió ở ô [1, 1]”

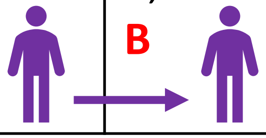
$$\alpha_4 :\ \neg B_{1,1}$$

- Thông tin mới: “Có gió ở ô [2, 1]”

$$\alpha_5 :\ B_{2,1}$$

$$KB = \alpha_1 \land \alpha_2 \land \alpha_3 \land \alpha_4 \land \alpha_5$$

1, 4	2, 4	3, 4	4, 4
1, 3	2, 3	3, 3	4, 3
1, 2	2, 2	3, 2	4, 2
1, 1	2, 1 B	3, 1	4, 1



Kiểm $KB \models \alpha$: bằng cách kiểm tra tất cả các mô hình

Quay lại với KB đã xây dựng ở ví dụ wumpus

- Có 7 biến mệnh đề → có $2^7 = 128$ mô hình có thể có
- Để kiểm $KB \models \neg P_{1,2}$: với mỗi mô hình mà KB đúng, xem thử $\neg P_{1,2}$ có đúng không? Nếu có một mô hình mà KB đúng nhưng $\neg P_{1,2}$ không đúng thì $KB \not\models \neg P_{1,2}$

Kiểm $KB \models \alpha$: bằng cách kiểm tra tất cả các mô hình

$$KB \models \neg P_{1,2}?$$

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	α_1	α_2	α_3	α_4	α_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Kiểm $KB \vDash \alpha$: bằng cách kiểm tra tất cả các mô hình

$$KB \vDash \neg P_{1,2}?$$

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	α_1	α_2	α_3	α_4	α_5	KB
false	*false*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	[illegible]	[illegible]	[illegible]	[illegible]	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	[illegible]	*false*	[illegible]	[illegible]	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

Nguồn ảnh: Stuart Russell and Peter Norvig, AI: A Modern Approach

Kiểm $KB \vDash \alpha$: bằng cách dùng luật suy diễn

Bạn thử trả lời xem: $KB \vDash \neg P_{1,2}$?

– Bạn có thể trả lời rất nhanh vì bạn dùng các **luật suy diễn** mà bạn đã được học trước đó trong logic

Kiểm $KB \vDash \alpha$: bằng cách dùng luật suy diễn

Luật suy diễn (inference rule)

– Giúp tạo ra các câu mới từ các câu đã có trong KB

– Ví dụ một luật suy diễn phổ biến là **luật modus ponens** (tam đoạn luận)

- $\dfrac{Tiền\ \text{đề}: \beta, \beta \Rightarrow \gamma}{Kết\ luận: \gamma}$
- Nghĩa là: cho β và γ là hai câu bất kỳ, nếu β và $\beta \Rightarrow \gamma$ có trong KB thì ta có thể thêm γ vào KB
- Ví dụ 1: $KB = \{Rain,\ Rain \Rightarrow Wet\}$, áp dụng luật modus ponens ta sẽ thêm được Wet vào KB

Kiểm $KB \vDash \alpha$: bằng cách dùng luật suy diễn

Luật suy diễn (inference rule)

– Giúp tạo ra các câu mới từ các câu đã có trong KB

– Ví dụ một luật suy diễn phổ biến là **luật modus ponens** (tam đoạn luận)

- Ví dụ 2: $KB = \{Rain,\ Rain \Rightarrow Wet,\ Wet \Rightarrow Slippery\}$
  - Áp dụng luật modus ponens cho $Rain$ và $Rain \Rightarrow Wet$:

    $KB = \{Rain,\ Rain \Rightarrow Wet,\ Wet \Rightarrow Slippery,\ Wet\}$
  - Áp dụng tiếp luật modus ponens cho Wet và $Wet \Rightarrow Slippery$

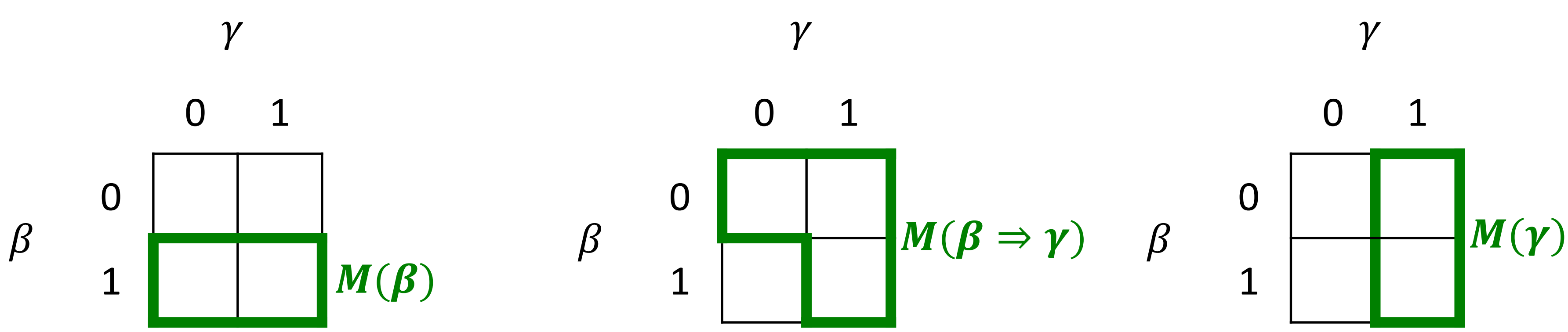
    $KB = \{Rain,\ Rain \Rightarrow Wet,\ Wet \Rightarrow Slippery,\ Wet,\ Slippery\}$
  - Không làm tiếp được nữa!

Kiểm $KB \vDash \alpha$: bằng cách dùng luật suy diễn

- Dùng các luật suy diễn để kiểm tra $KB \vDash \alpha$ như thế nào?
  – Áp dụng nhiều lần các luật suy diễn lên KB và xem có ra được α
- Tất nhiên, ta mong muốn các luật suy diễn phải **đúng**: các câu được tạo ra khi áp dụng nhiều lần các luật suy diễn lên KB phải là các câu được suy dẫn từ KB
- Ngoài ra, ta còn mong muốn các luật suy diễn phải **đầy đủ**: có khả năng tạo ra tất cả các câu được suy dẫn từ KB

Tính đúng và đầy đủ của luật modus ponens

- Luật modus ponens có đúng?
  - Kiểm γ có được suy dẫn từ $\beta \wedge (\beta \Rightarrow \gamma)$?
    - Có ☺



- Luật modus ponens có đầy đủ?
  - $KB = \{Rain, Rain \vee Snow \Rightarrow Wet\}$
  - $KB \vDash Wet$, nhưng có ra được Wet bằng luật modus ponens?
    - Không ☹

Vấn đề không đầy đủ của các luật suy diễn

- Các luật suy diễn thường đúng nhưng không đầy đủ
- Thậm chí, nếu dùng nhiều luật suy diễn thì vẫn thường sẽ không đầy đủ; hơn nữa, dùng nhiều luật suy diễn sẽ làm bùng nổ số câu được phát sinh ra
- Tiếp theo, ta sẽ tìm hiểu cách để khắc phục vấn đề không đầy đủ này

Nội dung tiếp theo

- Luật hợp giải
- Thuật toán sử dụng luật hợp giải để kiểm tra một câu α có được suy dẫn từ KB không
  - Thuật toán này sẽ đảm bảo tính đầy đủ (và tất nhiên là đúng nữa)

Luật hợp giải (resolution)

- Được đề xuất bởi Robinson vào năm 1965
- Một số khái niệm:
  - **Literal**: p hoặc $\neg p$ với p là một biến mệnh đề
  - **Clause**: là câu gồm các literal kết nối với nhau bởi phép ∨
- **Luật hợp giải (resolution)**: tiền đề là hai clause, trong đó literal l_i của clause 1 và literal m_j của clause 2 là hai literal tương phản (vd: A và $\neg A$); kết luận là một clause được tạo ra từ hai clause tiền đề, trong đó bỏ đi l_i và m_j

$$\frac{l_1 \vee l_2 \vee \cdots \vee l_{i-1} \vee l_i \vee l_{i+1} \vee \cdots \vee l_k, \quad m_1 \vee m_2 \vee \cdots \vee m_{j-1} \vee m_j \vee m_{j+1} \vee \cdots \vee m_n}{l_1 \vee l_2 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee m_2 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

Luật hợp giải (resolution)

Ví dụ 1:

– $KB = \{A \vee B,\ \neg B \vee C\}$

– Áp dụng luật hợp giải:

$$\frac{\begin{array}{c} A \vee B \\ \neg B \vee C \end{array}}{A \vee C}$$

Nếu B sai thì A phải đúng;

còn nếu B đúng thì C phải đúng;

do B có thể đúng hoặc sai nên A hoặc C phải đúng

– Như vậy, có thể thêm $A \vee C$ vào KB

Luật hợp giải (resolution)

Ví dụ 2:

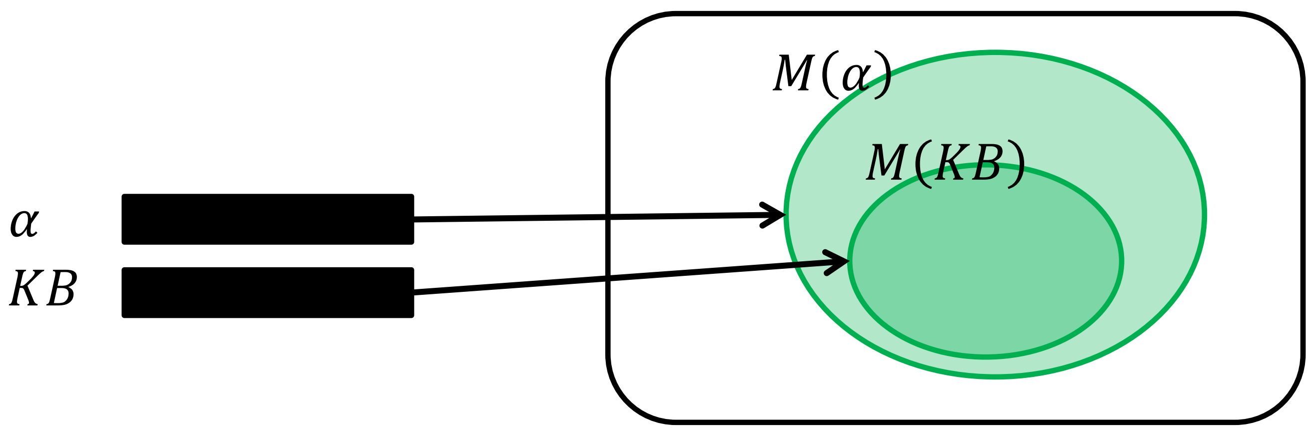
- $KB = \{A \lor B,\ \neg B \lor A\}$
- Áp dụng luật hợp giải ta sẽ thêm được A (ra được $A \lor A$, rút gọn thành A) vào KB

Luật hợp giải có đầy đủ không?

- Không ☺
  - Luật hợp giải yêu cầu các câu tiền đề có dạng clause, nhưng các câu trong KB có thể không có dạng clause
  - Thậm chí, nếu KB chỉ gồm các clause thì dùng luật hợp giải cũng không đảm bảo tạo ra được tất cả các câu được suy dẫn từ KB
    - Vd: với $KB = \{A\}$, ta không thể dùng luật hợp giải để tạo ra câu $A \vee B$ mặc dù $KB \vDash (A \vee B)$
- Nhưng thật ra là có ☺
  - Với KB bất kỳ, ta luôn có thể chuyển KB sang dạng chỉ gồm các clause
  - Với KB chỉ gồm các clause, tuy luật hợp giải không đảm bảo tạo ra được tất cả các câu được suy dẫn từ KB, nhưng tính đầy đủ sẽ được đảm bảo theo nghĩa: với một câu α bất kỳ, ta có thể dùng luật hợp giải để kiểm tra $KB \vDash \alpha$

Thuật toán hợp giải Robinson để kiểm tra $KB \vDash \alpha$

- Một câu β được gọi là **không thỏa mãn được** nếu và chỉ nếu không có mô hình nào làm cho β đúng, nghĩa là: $M(\beta) = \emptyset$; ngược lại, **thỏa mãn được** nếu $M(\beta) \neq \emptyset$
- $KB \vDash \alpha$ tương đương với $KB \land \neg\alpha$ thỏa mãn được hay không thỏa mãn được?
  - Không thỏa mãn được
  - Để kiểm $KB \vDash \alpha$, ta có thể kiểm $KB \land \neg\alpha$ không thỏa mãn được (giống như phương pháp chứng minh phản chứng)



Chứng minh phản chứng

1/ Chứng minh rằng: Với mọi số tự nhiên n nếu n^2 là số chẵn thì n là số chẵn

2/ Chứng minh rằng nếu nhốt 25 con thỏ vào 6 cái chuồng thì sẽ có ít nhất 1 chuồng chứa nhiều hơn 4 con thỏ.

Thuật toán hợp giải Robinson để kiểm tra $KB \vDash \alpha$

Thuật toán hợp giải Robinson: để kiểm $KB \vDash \alpha$, ta sẽ kiểm $KB \wedge \neg\alpha$ không thỏa mãn được

- Thêm $\neg\alpha$ vào KB và chuyển KB sang dạng tập các clause
- Áp dụng nhiều lần luật hợp giải lên KB
- Nếu trong KB xuất hiện hai clause mâu thuẫn nhau (vd: A và $\neg A$) thì dừng: $KB \wedge \neg\alpha$ không thỏa mãn được, tức là $KB \vDash \alpha$
- Còn nếu cho đến khi áp dụng luật hợp giải không làm thay đổi KB nữa mà vẫn chưa xuất hiện hai clause mâu thuẫn thì nghĩa là: $KB \wedge \neg\alpha$ thỏa mãn được, tức là $KB \nvDash \alpha$

Thuật toán hợp giải Robinson: mã giả

Input: KB (là một tập các câu), câu α**; output:** true nếu $KB \vDash \alpha$, false nếu ngược lại

Định nghĩa hàm:

- $KB \leftarrow KB \cup \{\neg\alpha\}$
- $KB \leftarrow convertToClauses(KB)$
- Lặp:
  - $newClauses \leftarrow \{\}$
  - Duyệt từng cặp clause C_i và C_j trong KB; với mỗi cặp:
    - Nếu C_i và C_j mâu thuẫn nhau: RETURN TRUE!
    - $results \leftarrow resolve(C_i, C_j)$ // Hợp giải C_i và C_j
    - $newClauses \leftarrow neuwClauses \cup results$
  - Nếu $newClauses \subseteq KB$: RETURN FALSE!
  - $KB \leftarrow KB \cup newClauses$

Làm sao để chuyển KB sang dạng tập các clause?

- **Câu dạng CNF** (Conjunctive Normal Form) là câu gồm các clause được nối lại với nhau bằng phép ∧
  - Ví dụ: $(A \lor B) \land (\neg C \lor D)$
- KB dạng tập các clause cũng chính là một câu dạng CNF
  - Vì như đã nói, có thể coi KB là một câu lớn gồm các câu thành phần được nối lại với nhau bởi phép ∧
- Bất kỳ một câu nào cũng đều có thể chuyển về dạng CNF

Làm sao để chuyển *KB* sang dạng tập các clause?

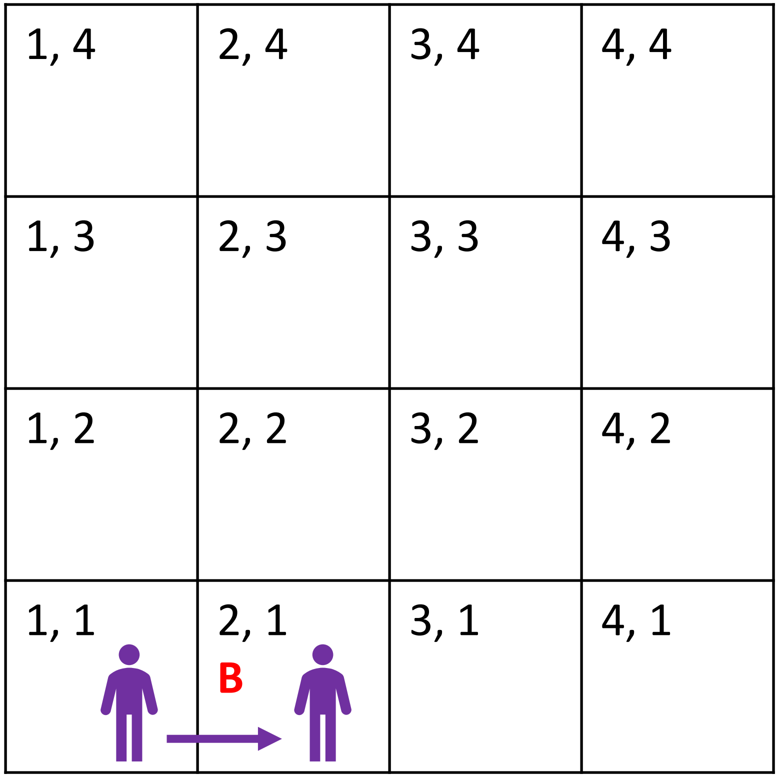
- Bất kỳ một câu nào cũng đều có thể chuyển về dạng CNF theo các bước sau:
  - Loại bỏ ⇔: chuyển $\alpha \Leftrightarrow \beta$ thành $(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)$
  - Loại bỏ ⇒: chuyển $\alpha \Rightarrow \beta$ thành $\neg\alpha \lor \beta$
  - Làm cho ¬ chỉ xuất hiện ở literal
    - Chuyển $\neg(\neg\alpha)$ thành α
    - Chuyển $\neg(\alpha \land \beta)$ thành $\neg\alpha \lor \neg\beta$
    - Chuyển $\neg(\alpha \lor \beta)$ thành $\neg\alpha \land \neg\beta$
  - Phân phối ∨ vào ∧: chuyển $\alpha \lor (\beta \land \gamma)$ thành $(\alpha \lor \beta) \land (\alpha \lor \gamma)$
- Như vậy, ta chuyển KB sang dạng tập các clause bằng cách chuyển các câu trong KB sang dạng CNF

Thuật toán hợp giải Robinson: ví dụ

Quay lại với KB đã xây dựng ở ví dụ wumpus

Kho tri thức KB (knowledge base)

Thử xây dựng KB của hệ thống logic trong thế giới wumpus



- Luật chơi: "Không có hố ở ô [1, 1]"

$$\alpha_1:\ \neg P_{1,1}$$

- Luật chơi: "Một ô có gió nếu và chỉ nếu có ít nhất một ô lân cận có hố"

$$\alpha_2:\ B_{1,1} \Leftrightarrow \left(P_{1,2} \vee P_{2,1}\right)$$

$$\alpha_3:\ B_{2,1} \Leftrightarrow \left(P_{1,1} \vee P_{2,2} \vee P_{3,1}\right)$$

- Thông tin mới: "Không có gió ở ô [1, 1]"

$$\alpha_4:\ \neg B_{1,1}$$

- Thông tin mới: "Có gió ở ô [2, 1]"

$$\alpha_5:\ B_{2,1}$$

$KB = \alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5$

Thuật toán hợp giải Robinson: ví dụ

Quay lại với KB đã xây dựng ở ví dụ wumpus; ta muốn kiểm tra $KB \vDash \neg P_{1,2}$?

1. Thêm $P_{1,2}$ vào KB và chuyển KB sang dạng tập các clause:

KB
$\neg P_{1,1}$
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

Thuật toán hợp giải Robinson: ví dụ

Quay lại với KB đã

$KB \vDash \neg P_{1,2}$?

1. Thêm $P_{1,2}$ vào

KB
$\neg P_{1,1}$
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

$$\equiv \left(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\right) \wedge \left((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\right)$$

$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$

$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge \left((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}\right)$$

$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

Thuật toán hợp giải Robinson: ví dụ

Quay lại với KB đã

$KB \vDash \neg P_{1,2}$?

1. Thêm $P_{1,2}$ vào

KB
$\neg P_{1,1}$
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$\neg B_{1,1}$

$$\equiv \left(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})\right) \wedge \left((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1}\right)$$
$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg(P_{1,2} \vee P_{2,1}) \vee B_{1,1})$$
$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge \left((\neg P_{1,2} \wedge \neg P_{2,1}) \vee B_{1,1}\right)$$
$$\equiv (\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}) \wedge (\neg P_{1,2} \vee B_{1,1}) \wedge (\neg P_{2,1} \vee B_{1,1})$$

$$\equiv \left(B_{2,1} \Rightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})\right) \wedge \left((P_{1,1} \vee P_{2,2} \vee P_{3,1}) \Rightarrow B_{2,1}\right)$$
$$\equiv (\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}) \wedge (\neg(P_{1,1} \vee P_{2,2} \vee P_{3,1}) \vee B_{2,1})$$
$$\equiv (\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}) \wedge \left((\neg P_{1,1} \wedge \neg P_{2,2} \wedge \neg P_{3,1}) \vee B_{2,1}\right)$$
$$\equiv (\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}) \wedge (\neg P_{1,1} \vee B_{2,1}) \wedge (\neg P_{2,2} \vee B_{2,1}) \wedge (\neg P_{3,1} \vee B_{2,1})$$

Thuật toán hợp giải Robinson: ví dụ

1. Thêm $P_{1,2}$ vào KB và chuyển KB sang dạng tập các clause:

KB
$\neg P_{1,1}$
$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$\neg P_{1,2} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$\neg P_{1,1} \vee B_{2,1}$
$\neg P_{2,2} \vee B_{2,1}$
$\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

Thuật toán hợp giải Robinson: ví dụ

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$\neg P_{1,2} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$\neg P_{1,1} \vee B_{2,1}$
$\neg P_{2,2} \vee B_{2,1}$
$\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

Thuật toán hợp giải Robinson: mã giả

Input: KB (là một tập các câu), câu α**; output:** true nếu $KB \vDash \alpha$, false nếu ngược lại

Định nghĩa hàm:

- $KB \leftarrow KB \cup \{\neg\alpha$
- $KB \leftarrow convertT$
- Lặp:
  - $newClauses$
  - Duyệt từng cặp clause C_i và C_j trong KB; với mỗi cặp:
    - Nếu C_i và C_j mâu thuẫn nhau: RETURN TRUE!
    - $results \leftarrow resolve(C_i, C_j)$ // Hợp giải C_i và C_j
    - $newClauses \leftarrow neuwClauses \cup results$
  - Nếu $newClauses \subseteq KB$: RETURN FALSE!
  - $KB \leftarrow KB \cup newClauses$

Theo mã giả thuật toán thì ta cần duyệt một cách có hệ thống các cặp clause trong KB

Nhưng ở ví dụ đang xét, KB khá lớn; duyệt một cách có hệ thống sẽ khá lâu nên mình sẽ duyệt một cách có ý đồ để minh họa một số ý và để kết thúc thuật toán sớm

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause	
$\neg P_{1,1}$	
$\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}$	$P_{1,2} \lor P_{2,1} \lor \neg P_{1,2}$
$\neg P_{1,2} \lor B_{1,1}$	
$\neg P_{2,1} \lor B_{1,1}$	
$\neg B_{2,1} \lor P_{1,1} \lor P_{2,2} \lor P_{3,1}$	
$\neg P_{1,1} \lor B_{2,1}$	
$\neg P_{2,2} \lor B_{2,1}$	
$\neg P_{3,1} \lor B_{2,1}$	
$\neg B_{1,1}$	
$B_{2,1}$	
$P_{1,2}$	

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause	
$\neg P_{1,1}$	
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$	$P_{1,2} \vee P_{2,1} \vee \neg P_{1,2}$
$\neg P_{1,2} \vee B_{1,1}$	$\neg B_{1,1} \vee P_{2,1} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$	
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$	
$\neg P_{1,1} \vee B_{2,1}$	
$\neg P_{2,2} \vee B_{2,1}$	
$\neg P_{3,1} \vee B_{2,1}$	
$\neg B_{1,1}$	
$B_{2,1}$	
$P_{1,2}$	

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \lor P_{1,2} \lor P_{2,1}$
$\neg P_{1,2} \lor B_{1,1}$
$\neg P_{2,1} \lor B_{1,1}$
$\neg B_{2,1} \lor P_{1,1} \lor P_{2,2} \lor P_{3,1}$
$\neg P_{1,1} \lor B_{2,1}$
$\neg P_{2,2} \lor B_{2,1}$
$\neg P_{3,1} \lor B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

$\cancel{P_{1,2} \lor P_{2,1} \lor \neg P_{1,2}}$

$\cancel{\neg B_{1,1} \lor P_{2,1} \lor B_{1,1}}$

Clause mà có chứa 2 literal tương phản là clause luôn đúng với mọi mô hình → không giúp ích được gì → có thể bỏ đi

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$\neg P_{1,2} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$\neg P_{1,1} \vee B_{2,1}$
$\neg P_{2,2} \vee B_{2,1}$
$\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

Ta có thể chọn lựa thứ tự hợp giải một cách khôn ngoan để mau chóng tạo ra mâu thuẫn và kết thúc thuật toán

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$\neg P_{1,2} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$\neg P_{1,1} \vee B_{2,1}$
$\neg P_{2,2} \vee B_{2,1}$
$\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

$\neg P_{1,2}$

Ta có thể chọn lựa thứ tự hợp giải một cách khôn ngoan để mau chóng tạo ra mâu thuẫn và kết thúc thuật toán

Thuật toán hợp giải Robinson: ví dụ 1

2. Áp dụng nhiều lần luật hợp giải lên KB:

KB dạng tập các clause
$\neg P_{1,1}$
$\neg B_{1,1} \vee P_{1,2} \vee P_{2,1}$
$\neg P_{1,2} \vee B_{1,1}$
$\neg P_{2,1} \vee B_{1,1}$
$\neg B_{2,1} \vee P_{1,1} \vee P_{2,2} \vee P_{3,1}$
$\neg P_{1,1} \vee B_{2,1}$
$\neg P_{2,2} \vee B_{2,1}$
$\neg P_{3,1} \vee B_{2,1}$
$\neg B_{1,1}$
$B_{2,1}$
$P_{1,2}$

$\neg P_{1,2}$

Ta có thể chọn lựa thứ tự hợp giải một cách khôn ngoan để mau chóng tạo ra mâu thuẫn và kết thúc thuật toán

Mâu thuẫn!

Vậy $KB \wedge P_{1,2}$ không thỏa mãn được

Tức là $KB \vDash \neg P_{1,2}$

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C?$

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

1. Thêm $\neg C$ vào KB và chuyển KB sang dạng tập các clause

STT	KB dạng tập các clause
1	$\neg A \vee B \vee \neg C$
2	A
3	$\neg B$
4	$\neg C$

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C?$

2. Áp dụng nhiều lần luật hợp giải lên KB

STT	KB dạng tập các clause
1	$\neg A \vee B \vee \neg C$
2	A
3	$\neg B$
4	$\neg C$

Nhìn sơ bộ thì thấy khó có thể chọn nhanh các cặp clause để hợp giải và ra mâu thuẫn → nên duyệt các cặp clause một cách có hệ thống để tránh sót trường hợp

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

	STT	KB dạng tập các clause
KB	1	$\neg A \vee B \vee \neg C$
	2	A
	3	$\neg B$
	4	$\neg C$
	5	$B \vee \neg C$ (hợp giải 1 và 2)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

	STT	KB dạng tập các clause
KB	1	$\neg A \vee B \vee \neg C$
	2	A
	3	$\neg B$
	4	$\neg C$
Tập các câu mới được tạo ra từ KB	5	$B \vee \neg C$ (hợp giải 1 và 2)
	6	$\neg A \vee \neg C$ (hợp giải 1 và 3)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \lor \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

KB mới

STT	KB dạng tập các clause
1	$\neg A \lor B \lor \neg C$
2	A
3	$\neg B$
4	$\neg C$
5	$B \lor \neg C$ (hợp giải 1 và 2)
6	$\neg A \lor \neg C$ (hợp giải 1 và 3)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

KB mới

STT	KB dạng tập các clause
1	$\neg A \vee B \vee \neg C$
2	A
3	$\neg B$
4	$\neg C$
5	$B \vee \neg C$ (hợp giải 1 và 2)
6	$\neg A \vee \neg C$ (hợp giải 1 và 3)
7	¬C (hợp giải 2 và 6)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C?$

2. Áp dụng nhiều lần luật hợp giải lên KB

	STT	KB dạng tập các clause
KB mới	1	$\neg A \vee B \vee \neg C$
	2	A
	3	$\neg B$
	4	$\neg C$
	5	$B \vee \neg C$ (hợp giải 1 và 2)
	6	$\neg A \vee \neg C$ (hợp giải 1 và 3)
Đã có trong KB!	7	¬C (hợp giải 2 và 6)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

KB mới

STT	KB dạng tập các clause
1	$\neg A \vee B \vee \neg C$
2	A
3	$\neg B$
4	$\neg C$
5	$B \vee \neg C$ (hợp giải 1 và 2)
6	$\neg A \vee \neg C$ (hợp giải 1 và 3)
7	¬C (hợp giải 3 và 5)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

	STT	KB dạng tập các clause
KB mới	1	$\neg A \vee B \vee \neg C$
	2	A
	3	$\neg B$
	4	$\neg C$
	5	$B \vee \neg C$ (hợp giải 1 và 2)
	6	$\neg A \vee \neg C$ (hợp giải 1 và 3)
Đã có trong KB!	7	¬C (hợp giải 3 và 5)

Thuật toán hợp giải Robinson: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

2. Áp dụng nhiều lần luật hợp giải lên KB

STT	KB dạng tập các clause
1	$\neg A \vee B \vee \neg C$
2	A
3	$\neg B$
4	$\neg C$
5	$B \vee \neg C$ (hợp giải 1 và 2)
6	$\neg A \vee \neg C$ (hợp giải 1 và 3)

KB mới

Không thể ra thêm câu mới được nữa,
mà trong KB vẫn chưa xuất hiện mâu thuẫn
Vậy: $KB \wedge \neg C$ thỏa mãn được, tức là $KB \nvDash C$

Cải tiến thuật toán hợp giải Robinson

- Việc duyệt hết các cặp clause có thể có rất tốn thời gian
  - Khi mà cho KB tương đối lớn + câu α không thể suy dẫn từ KB → phải duyệt các cặp clause có thể có và áp dụng luật hợp giải cho đến khi không ra được câu mới nữa → hy sinh ☹
- Liệu có cách làm nào khác hiệu quả hơn không?
  - Thuật toán hợp giải Robinson **+ Davis Putnam** ☺

Thuật toán hợp giải Robinson + Davis Putnam (DP)

Để kiểm $KB \vDash \alpha$, ta sẽ kiểm $KB \land \neg\alpha$ không thỏa mãn được

– Thêm $\neg\alpha$ vào KB và chuyển KB sang dạng tập các clause

– Với mỗi biến mệnh đề mà có cặp clause để hợp giải:

  • Hợp giải tất cả các cặp clause có thể có của biến mệnh đề này; nếu có cặp clause mâu thuẫn nhau thì dừng: $KB \land \neg\alpha$ không thỏa mãn được, tức là $KB \vDash \alpha$

  • Thêm các clause kết quả (sẽ không còn chứa biến mệnh đề này) vào KB; bỏ các clause kết quả mà có chứa 2 literal tương phản (vd: $A \lor \neg A \lor \cdots$)

  • Bỏ tất cả các clause có chứa biến mệnh này ra khỏi KB

– Cho đến cuối cùng mà vẫn chưa xuất hiện cặp clause mâu thuẫn thì nghĩa là: $KB \land \neg\alpha$ thỏa mãn được, tức là $KB \nvDash \alpha$

Thuật toán hợp giải Robinson + DP: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

- Thêm $\neg C$ vào KB và chuyển KB sang dạng tập các clause: $KB = \{\neg A \vee B \vee \neg C, A, \neg B, \neg C\}$

Thuật toán hợp giải Robinson + DP: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

- Thêm $\neg C$ vào KB và chuyển KB sang dạng tập các clause: $KB = \{\neg A \vee B \vee \neg C, A, \neg B, \neg C\}$
- Hợp giải biến A:

Thuật toán hợp giải Robinson + DP: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

- Thêm $\neg C$ vào KB và chuyển KB sang dạng tập các clause: $KB = \{\neg A \vee B \vee \neg C, A, \neg B, \neg C\}$
- Hợp giải biến A: $KB = \{B \vee \neg C, \neg B, \neg C\}$

Thuật toán hợp giải Robinson + DP: ví dụ 2

Cho: $KB = \{A \Rightarrow (B \vee \neg C),\ A,\ \neg B\}$

Hỏi: $KB \vDash C$?

- Thêm $\neg C$ vào KB và chuyển KB sang dạng tập các clause: $KB = \{\neg A \vee B \vee \neg C, A, \neg B, \neg C\}$
- Hợp giải biến A: $KB = \{B \vee \neg C, \neg B, \neg C\}$
- Hợp giải biến B: $KB = \{\neg C\}$

Không ra cặp clause mâu thuẫn; vậy: $KB \wedge \neg C$ thỏa mãn được, tức là $KB \nvDash C$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

$$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

$$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

- Hợp giải biến A: $KB = \{B \vee C, \neg B \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

  $$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

- Hợp giải biến A: $KB = \{B \vee C, \neg B \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến B: $KB = \{C \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

$$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

- Hợp giải biến A: $KB = \{B \vee C, \neg B \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến B: $KB = \{C \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến C: $KB = \{D \vee E, \neg D \vee E, \neg E\}$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

  $$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

- Hợp giải biến A: $KB = \{B \vee C, \neg B \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến B: $KB = \{C \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến C: $KB = \{D \vee E, \neg D \vee E, \neg E\}$
- Hợp giải biến D: $KB = \{E, \neg E\}$

Thuật toán hợp giải Robinson + DP: ví dụ 3

Cho: $KB = \{A \Rightarrow (B \vee C),\ B \Rightarrow (D \vee E),\ (A \wedge D) \Rightarrow E,\ C \Rightarrow E\ \}$

Hỏi: $KB \vDash (A \Rightarrow E)$?

- Thêm $\neg(A \Rightarrow E)$ vào KB và chuyển KB sang dạng tập các clause:

  $$KB = \{\neg A \vee B \vee C, \neg B \vee D \vee E, \neg A \vee \neg D \vee E, \neg C \vee E, A, \neg E\}$$

- Hợp giải biến A: $KB = \{B \vee C, \neg B \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến B: $KB = \{C \vee D \vee E, \neg D \vee E, \neg C \vee E, \neg E\}$
- Hợp giải biến C: $KB = \{D \vee E, \neg D \vee E, \neg E\}$
- Hợp giải biến D: $KB = \{E, \neg E\}$
- Hợp giải biến E: thấy cặp clause mâu thuẫn; vậy: ?

Hợp giải Robinson – Bài tập

- Chứng minh $\boldsymbol{A \Rightarrow F}$ từ KB như bên cạnh

KB

$A \Rightarrow (B \vee C)$

$B \Rightarrow (D \vee F)$

$A \wedge D \Rightarrow F$

$C \Rightarrow F$

- Chứng minh $\boldsymbol{R}$ từ KB như bên cạnh

KB

$(P \Rightarrow Q) \Rightarrow Q$

$R \Rightarrow (R \Rightarrow \neg P)$

$(R \Rightarrow S) \Rightarrow \neg(S \Rightarrow Q)$

- Chứng minh $\boldsymbol{C}$ từ KB như bên cạnh
- Chứng minh $\boldsymbol{B \Rightarrow \neg C}$ từ KB như bên cạnh

KB

$A \Rightarrow B \vee C$

$A \Rightarrow D$

$C \wedge D \Rightarrow \neg F$

$B \Rightarrow F$

A