



南京大學
NANJING UNIVERSITY

人工智能导论

推理与规划 (Reasoning & Planning)

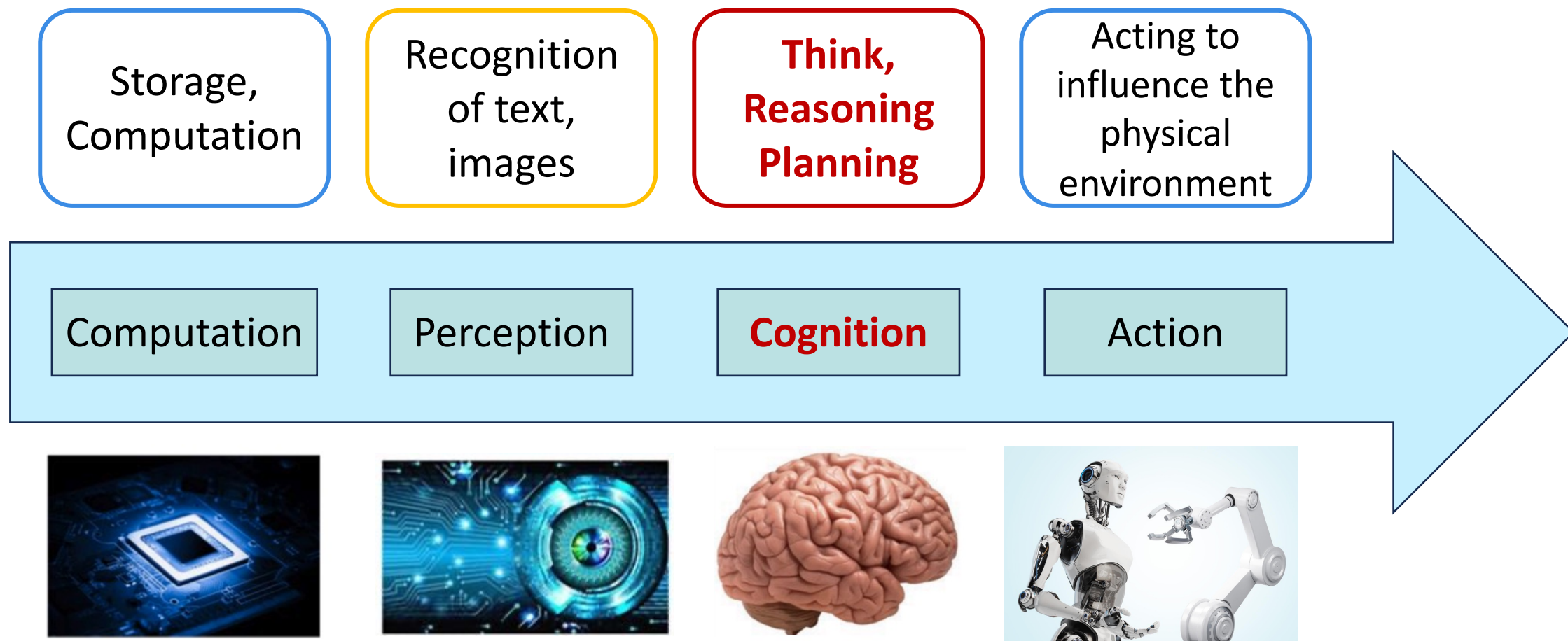
郭兰哲

南京大学 智能科学与技术学院

<https://www.lamda.nju.edu.cn/guolz/IntroAI/fall2025/index.html>

Email: guolz@nju.edu.cn

从感知到认知



从感知到认知

□ 构建具有强推理能力的AI模型被认为是实现AGI的重要里程碑

- OpenAI的AGI分级：L1：聊天机器人 -> L2：推理者 -> L3：智能体 -> L4：创新者 -> L5：组织者

Stages of Artificial Intelligence

Level 1	Chatbots, AI with conversational language
Level 2	Reasoners, human-level problem solving
Level 3	Agents, systems that can take actions
Level 4	Innovators, AI that can aid in invention
Level 5	Organizations, AI that can do the work of an organization

推理(Reasoning)

- 推理是进行思维模拟的基本形式，是从一个或几个已知的判断(前提)推出新判断(结论)的过程
- 一般包括：
 - 演绎推理 (deductive)
 - 归纳推理 (inductive)
 - 反绎推理 (abductive)
 - 因果推理 (causality)
 -

演绎推理(Deductive Reasoning)

- 演绎推理：从一般性前提出发，通过演绎推导，得到结论的过程
- **Example:** 数学推理、形式逻辑推理

Math Reasoning

Problem: Tom has a red marble, a green marble, a blue marble, and three identical yellow marbles. How many different groups of two marbles can Tom choose?

Solution: There are two cases here: either Tom chooses two yellow marbles (1 result), or he chooses two marbles of different colors ($\binom{4}{2} = 6$ results). The total number of distinct pairs of marbles Tom can choose is $1 + 6 = \boxed{7}$.

Theorem Proving

To prove: $n \in \mathbb{N} \wedge \frac{n+1}{2} \in \mathbb{N} \implies \exists m \in \mathbb{N} : n = 2m + 1$.
GPT-*f*'s generated proof:

```
| - ((N e. NN0 /\ ((N + 1) / 2) e. NN0) ->
      ((N - 1) / 2) e. NN0)
| - (N e. NN0 -> N e. CC)
| - 1 e. CC
| - ((N e. CC /\ 1 e. CC) ->
      (N - 1) e. CC )
```

演绎推理(Deductive Reasoning)

AI模型获IMO金牌

July 21, 2025 Research

Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

Thang Luong and Edward Lockhart

演绎推理(Deductive Reasoning)

LLM是记忆还是推理?

GSM-NoOp

Oliver picks 44 kiwis on Friday. Then he picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, but five of them were a bit smaller than average. How many kiwis does Oliver have?

o1-mini: Oliver picks kiwis over three days as follows:

- Friday: 44 kiwis
- Saturday: 58 kiwis
- Sunday: Double the number he picked on Friday, which is $2 \times 44 = 88$ kiwis

However, on Sunday, 5 of these kiwis were smaller than average. We need to subtract them from the Sunday total: 88 (Sunday's kiwis) - 5 (smaller kiwis) = 83 kiwis. Now, summing up the kiwis from all three days: 44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis. Oliver has a total of 185 kiwis.

Llama3-8B: Let's break it down step by step:

Oliver picks 44 kiwis on Friday. He picks 58 kiwis on Saturday. On Sunday, he picks double the number of kiwis he did on Friday, which is $2 \times 44 = 88$ kiwis.

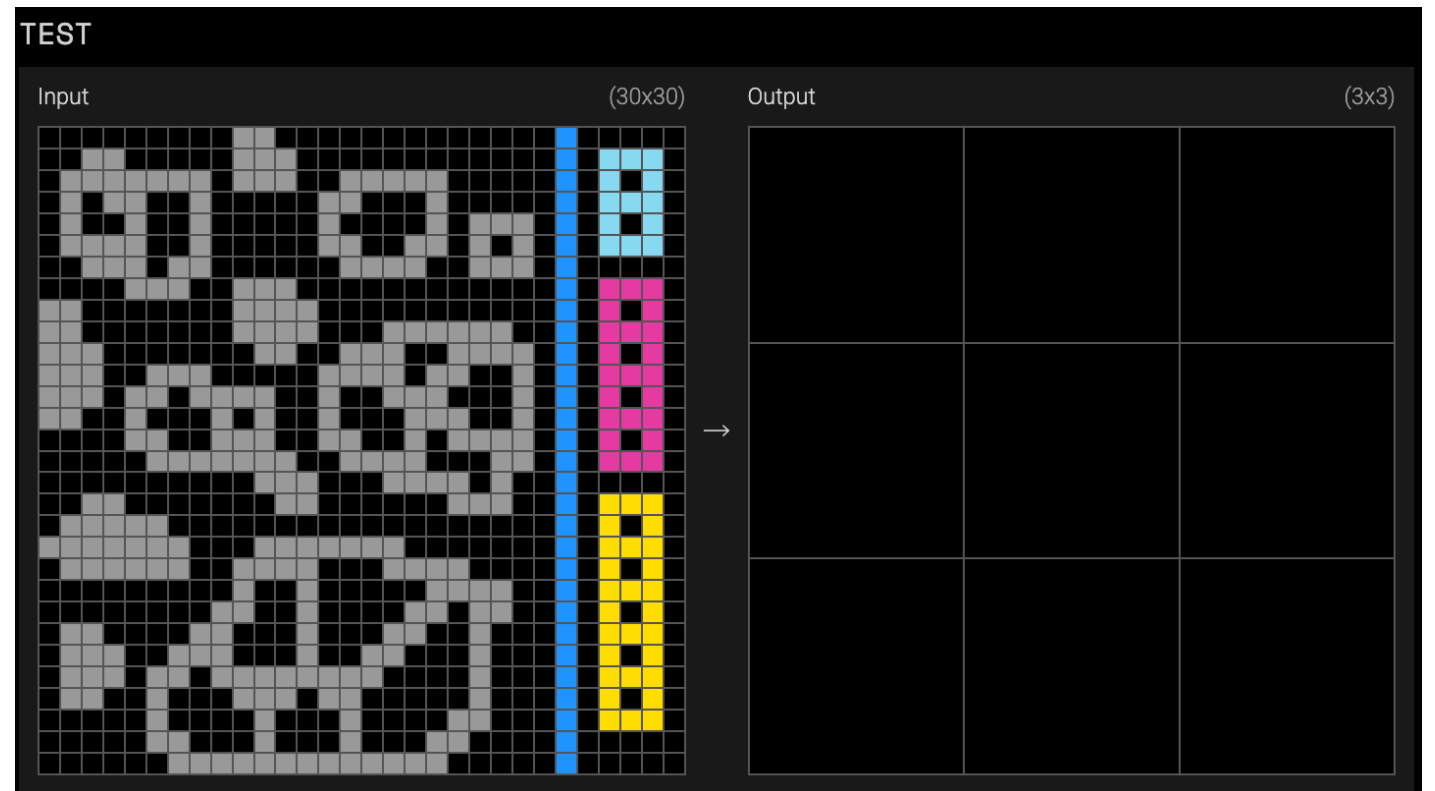
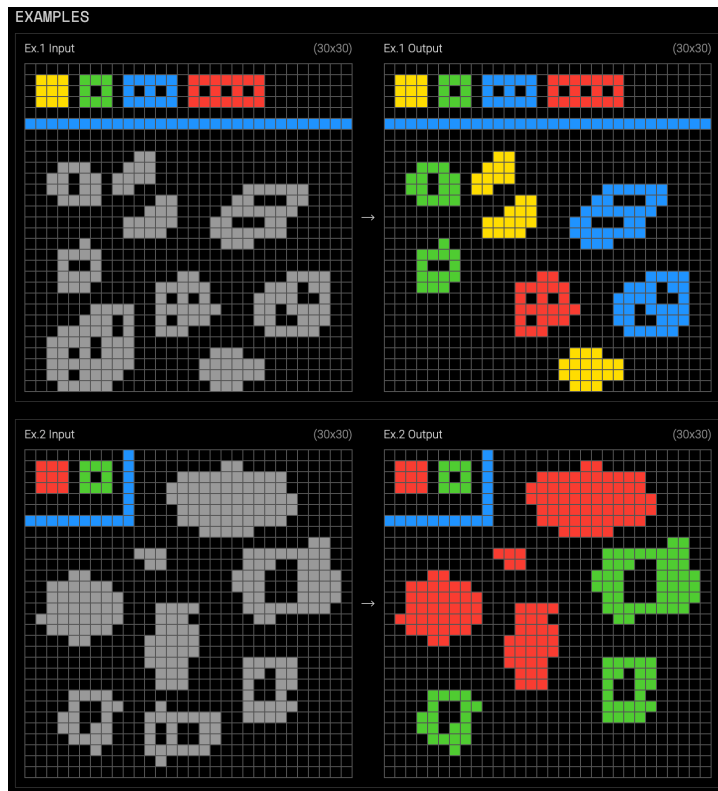
Five of the kiwis picked on Sunday are a bit smaller than average, so we subtract 5 from the total number of kiwis picked on Sunday: $88 - 5 = 83$ kiwis. Now, let's add up the total number of kiwis Oliver has:

44 (Friday) + 58 (Saturday) + 83 (Sunday) = 185 kiwis

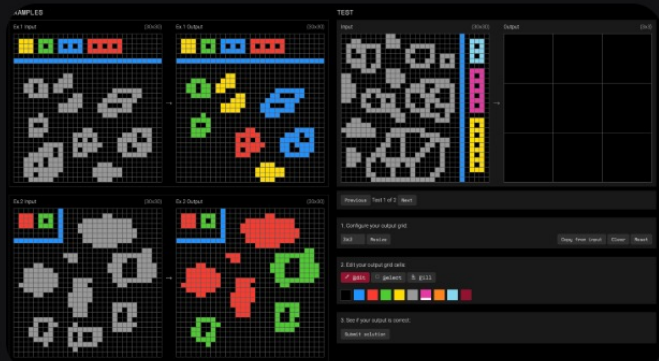
So, Oliver has 185 kiwis in total.

归纳推理(Inductive Reasoning)

- 归纳推理：从个别事实出发，推导出一般性知识作为结论
- **Example: ARC Challenge (<https://arcprize.org/>)**



归纳推理(Inductive Reasoning)



请你观察左边的输入图像样例，总结出其中的规则，并给出右边的问题对应的输出

这是一个非常经典的 ARC (Abstraction and Reasoning Corpus) 谜题。我们可以通过观察 Example 1 和 Example 2 来推导规则，并将其应用到 Test 中。

规则总结

这道题的核心逻辑是拓扑学中的“孔洞”计数 (Topology / Hole Counting)。

1. 观察“图例” (Legend):

- 输入图像被一条蓝线分割成两个部分。大的部分是主要绘图区，小的部分 (Example 1 在顶部, Test 在右侧) 是图例/对照表。
- 图例中的每个彩色形状代表了一个特定的孔洞数量。
- 你需要数一下图例中每个彩色形状内部有几个封闭的黑色区域 (孔)。

2. 应用规则:

- 观察主要绘图区中的灰色形状。
- 数一数每个灰色形状内部有多少个孔洞。
- 将该灰色形状填充为拥有相同孔洞数量的图例形状的颜色。

归纳推理(Inductive Reasoning)

ARC Is a Vision Problem!

Keya Hu Ali Cy Linlu Qiu Xiaoman Delores Ding
Runqian Wang Yeyin Eva Zhu Jacob Andreas Kaiming He

MIT

Abstract

The Abstraction and Reasoning Corpus (ARC) is designed to promote research on abstract reasoning, a fundamental aspect of human intelligence. Common approaches to ARC treat it as a language-oriented problem, addressed by large language models (LLMs) or recurrent reasoning models. However, although the puzzle-like tasks in ARC are inherently visual, existing research has rarely approached the problem from a vision-centric perspective. In this work, we formulate ARC within a vision paradigm, framing it as an image-to-image translation problem. To incorporate visual priors, we represent the inputs on a “canvas” that can be processed like natural images. It is then natural for us to apply standard vision architectures, such as a vanilla Vision Transformer (ViT), to perform image-to-image mapping. Our model is trained from scratch solely on ARC data and generalizes to unseen tasks through test-time training. Our framework, termed Vision ARC (VARC), achieves 60.4% accuracy on the ARC-1 benchmark, substantially outperforming existing methods that are also trained from scratch. Our results are competitive with those of leading LLMs and close the gap to average human performance.¹

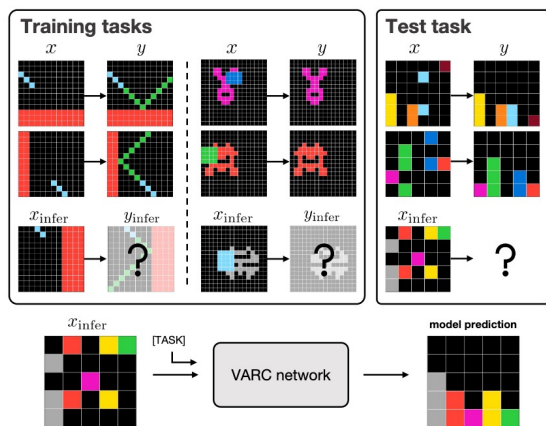


Figure 1. The ARC benchmark (top) consists of a collection of many different tasks, where each task has a few (e.g., 2-4) examples. We propose the Vision ARC (VARC) framework, which addresses the ARC problem as an image-to-image translation problem, from a computer vision perspective (bottom). In this illustration, the underlying concepts of the three tasks can be roughly described by humans as: “reflection” (left), “symmetry” (middle), and “gravity” (right). These concepts are closely related to the visual and physical world.


<https://arxiv.org/abs/2511.14761>

system	#params	ARC-1	ARC-2
<i>large language models (LLMs)</i>			
Deepseek R1 [21]	671B	15.8	1.3
Claude 3.7 8k [18]	N/A	21.2	0.9
o3-mini-high [18]	N/A	34.5	3.0
GPT-5 [18]	N/A	44.0	1.9
Grok-4-thinking [18]	1.7T	66.7	16.0
Bespoke (Grok-4) [8]	1.7T	79.6	29.4
<i>recurrent models</i>			
HRM [53]	27M	40.3	5.0
TRM [27]	7M	44.6	7.8
<i>vision models</i>			
VARC	18M	<u>54.5</u>	<u>8.3</u>
VARC (ensemble)	73M	60.4	11.1
<i>human results</i>			
avg. human [31]	-	60.2	-
best human [18]	-	98.0	100.0

反绎推理(Abductive Reasoning)

- 反绎推理：最佳解释推理，观察到某个现象，推出最可能的原因
- Example：医学诊断类任务

What can you **infer** from the **visual clues** ?



large semi truck and trailer on its side laying on a freeway

There was a major accident that occurred minutes ago

The people are inspecting damage to the vehicles in the accident

patches of snow spread throughout grass on the side of freeway

Cold weather is causing hazardous conditions at this location

The roads are very icy

a white license plate with five red English style numbers displayed

This accident happened in an English speaking country

This is Ohio

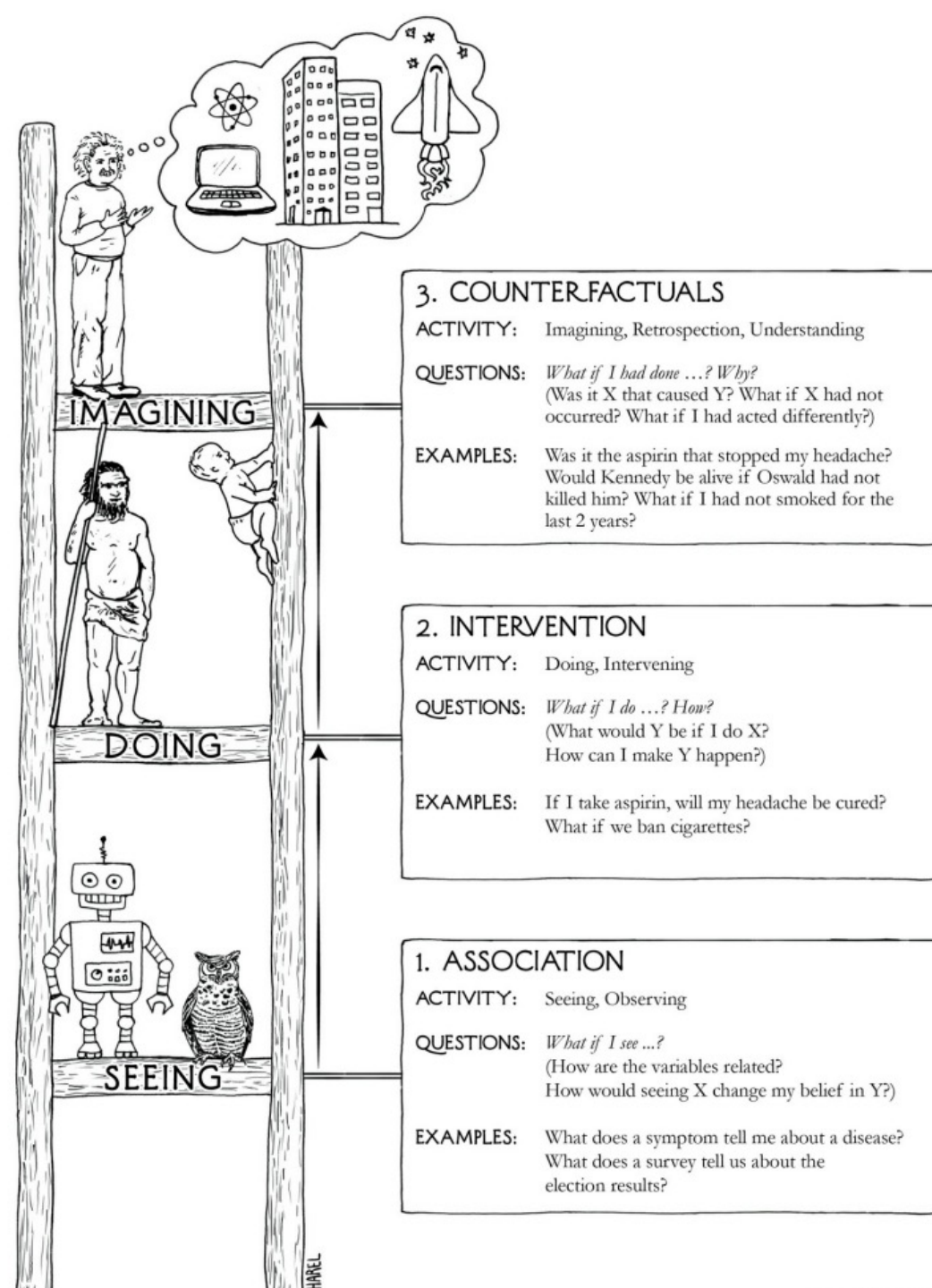
因果推理(Causal Reasoning)

因果推理：推理事件与事件之间的因果关系

相关性与因果性：鸡叫，天就会亮

Jude Pearl的因果三层级：

- **关联**：如果X发生，Y通常是什么
- **干预**：如果我做了X，Y会怎么样？
- **反事实**：如果我没做X，会发生什么？

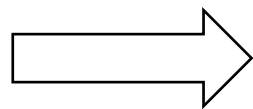


符号主义人工智能

符号主义人工智能：智能可以通过符号 + 规则来表达和推理

三段论

1. 所有人都会死
2. 苏格拉底是人
3. 苏格拉底会死



符号表示

1. $\forall x (\text{Human}(x)) \rightarrow \text{Mortal}(x)$
2. $\text{Human}(\text{Socrates})$
3. $\text{Mortal}(\text{Socrates})$

逻辑智能体：Wumpus世界

➤ Performance Measure

- 金块+1000，死亡-1000
- 每一步-1，用箭-10

➤ 环境

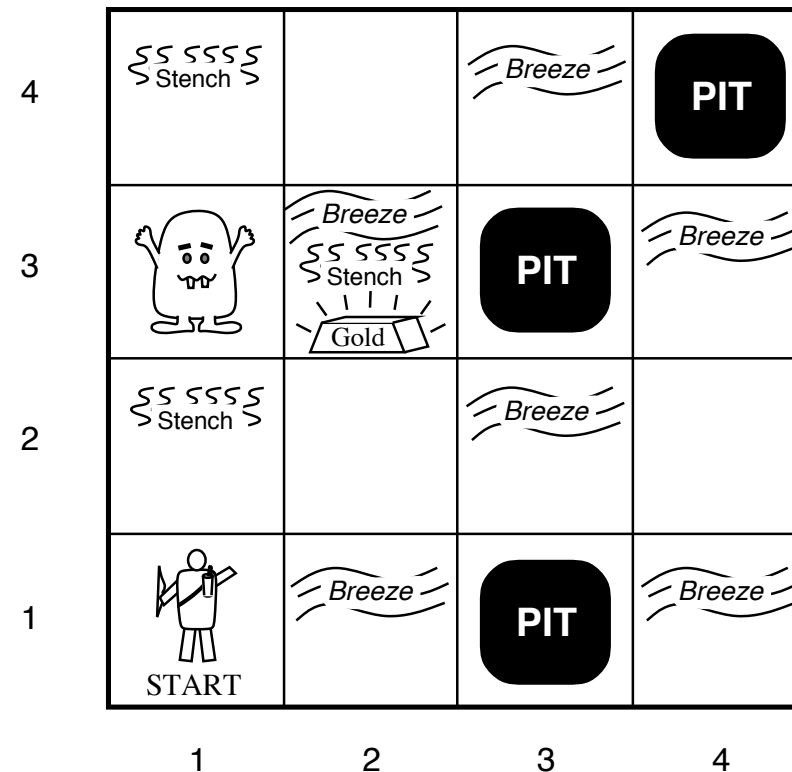
- 4*4房间
- 从[1,1]开始，面向东方

➤ 动作

- 向前、左转、右转、射箭、捡金块、爬出洞穴(仅能从[1,1]爬出)

➤ 感知

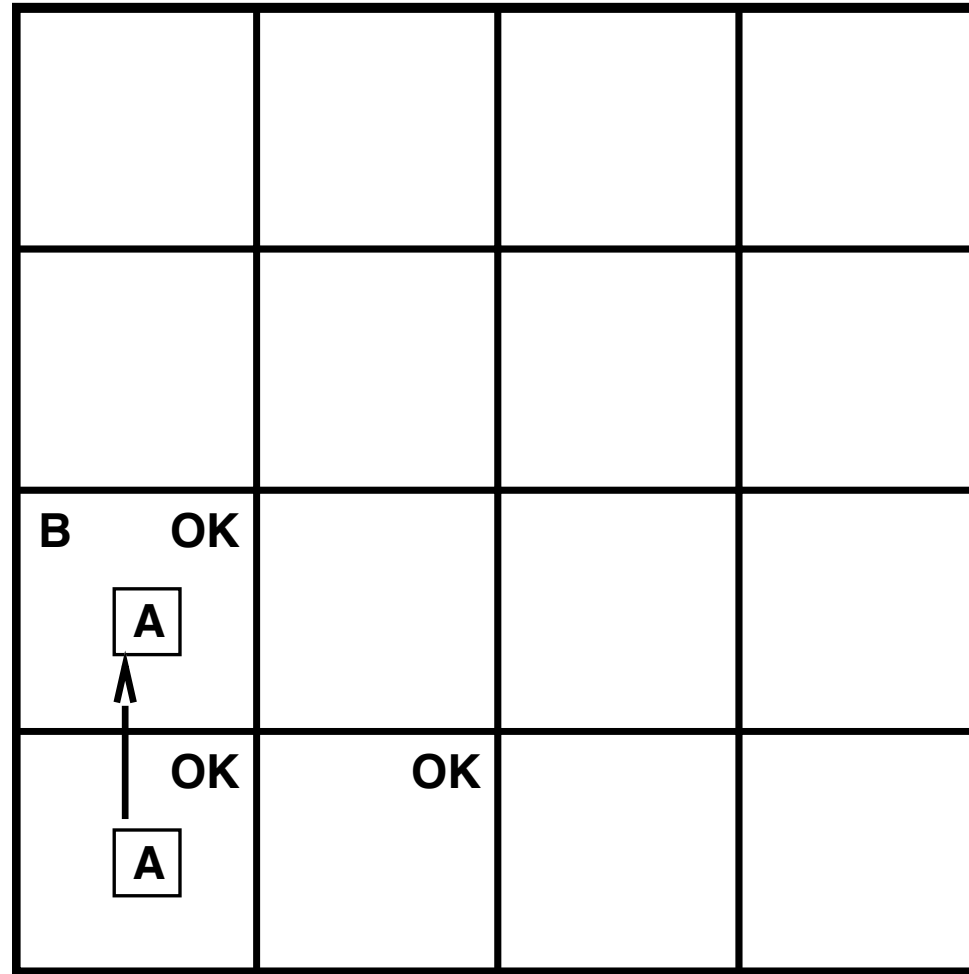
- wumpus相邻格子有臭味、无底洞相邻格子有微风、金块所在格子有闪光、走向墙壁感知到碰撞、wumpus被杀死可以在任意位置感知到惨叫



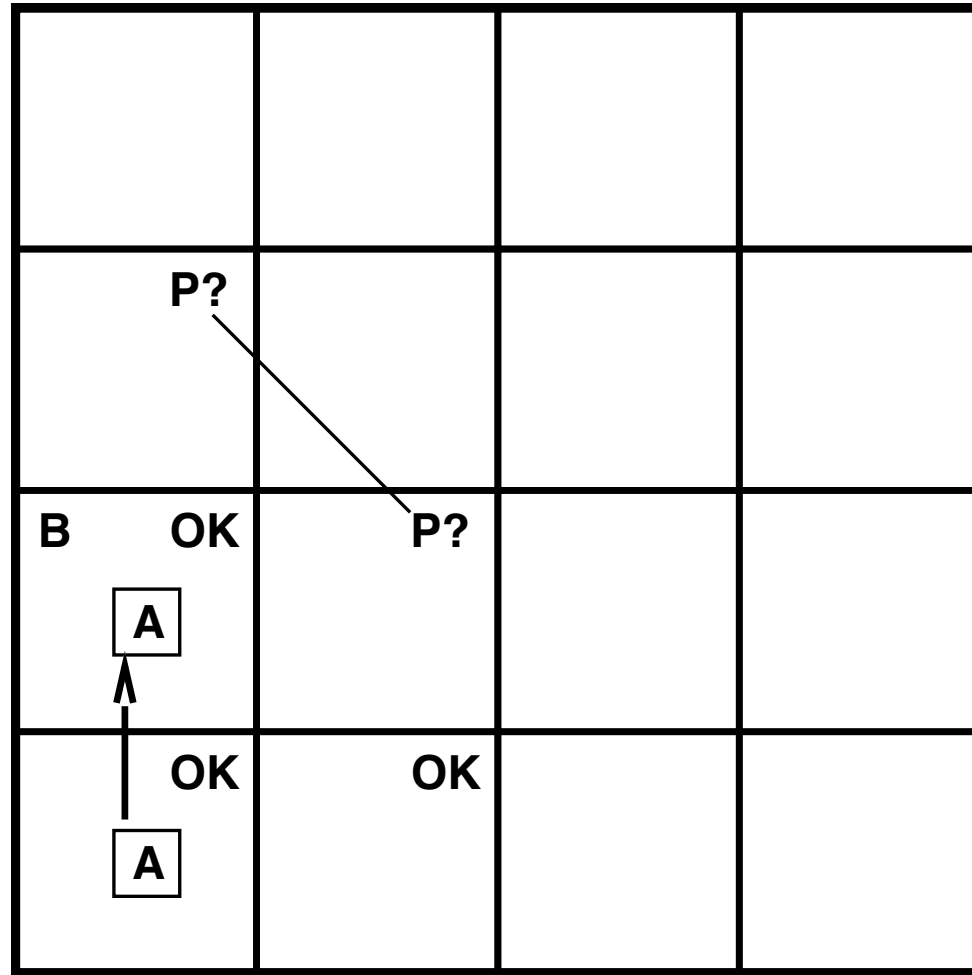
逻辑智能体：Wumpus世界

OK			
OK A	OK		

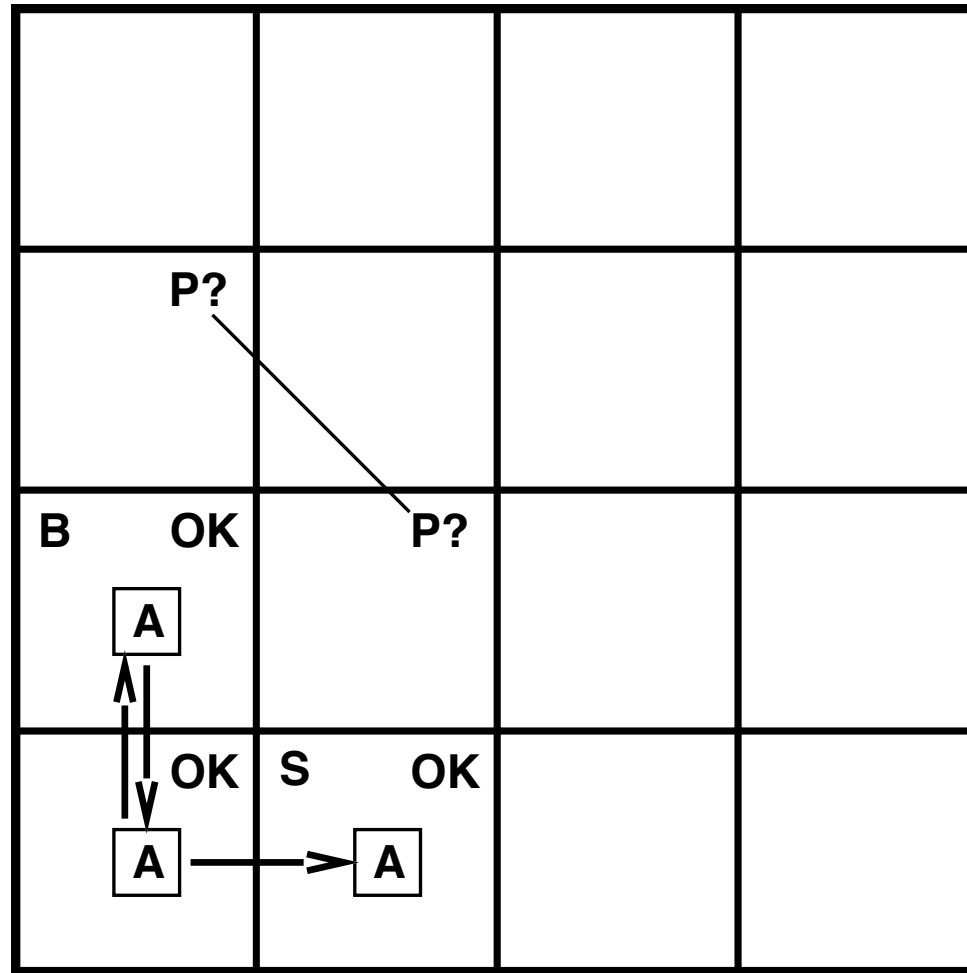
逻辑智能体：Wumpus世界



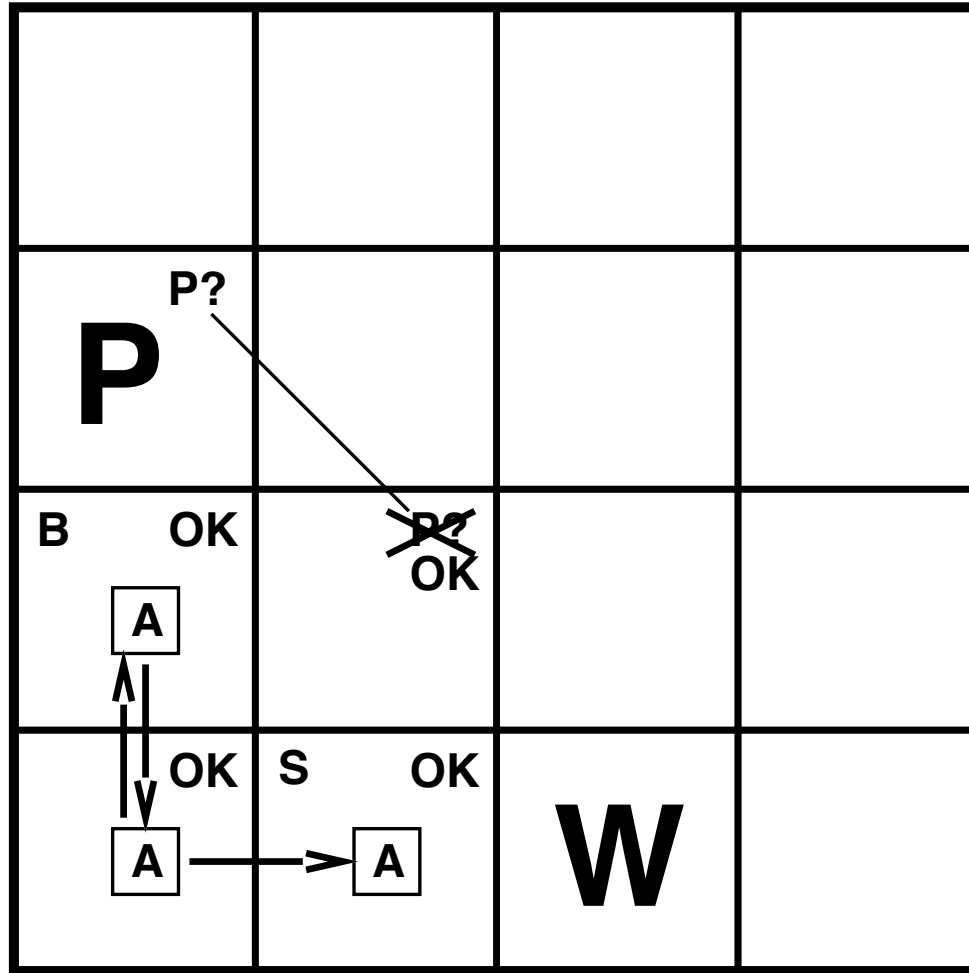
逻辑智能体：Wumpus世界



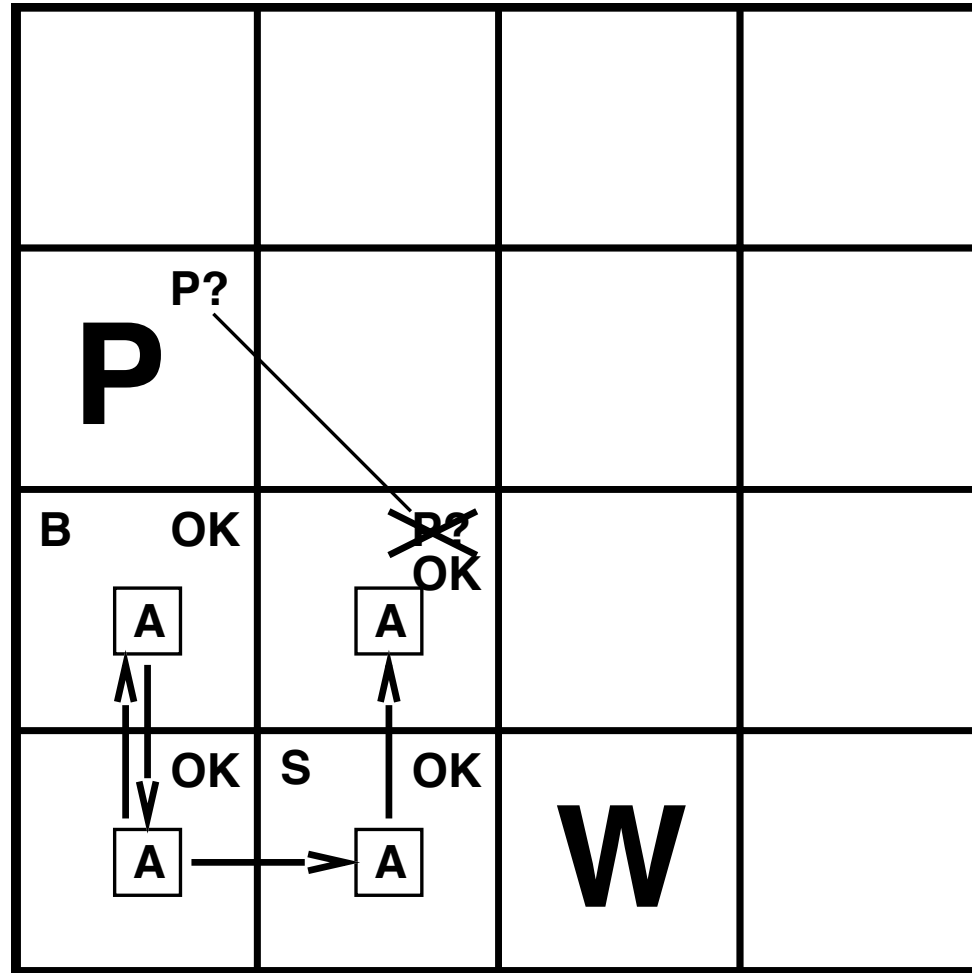
逻辑智能体：Wumpus世界



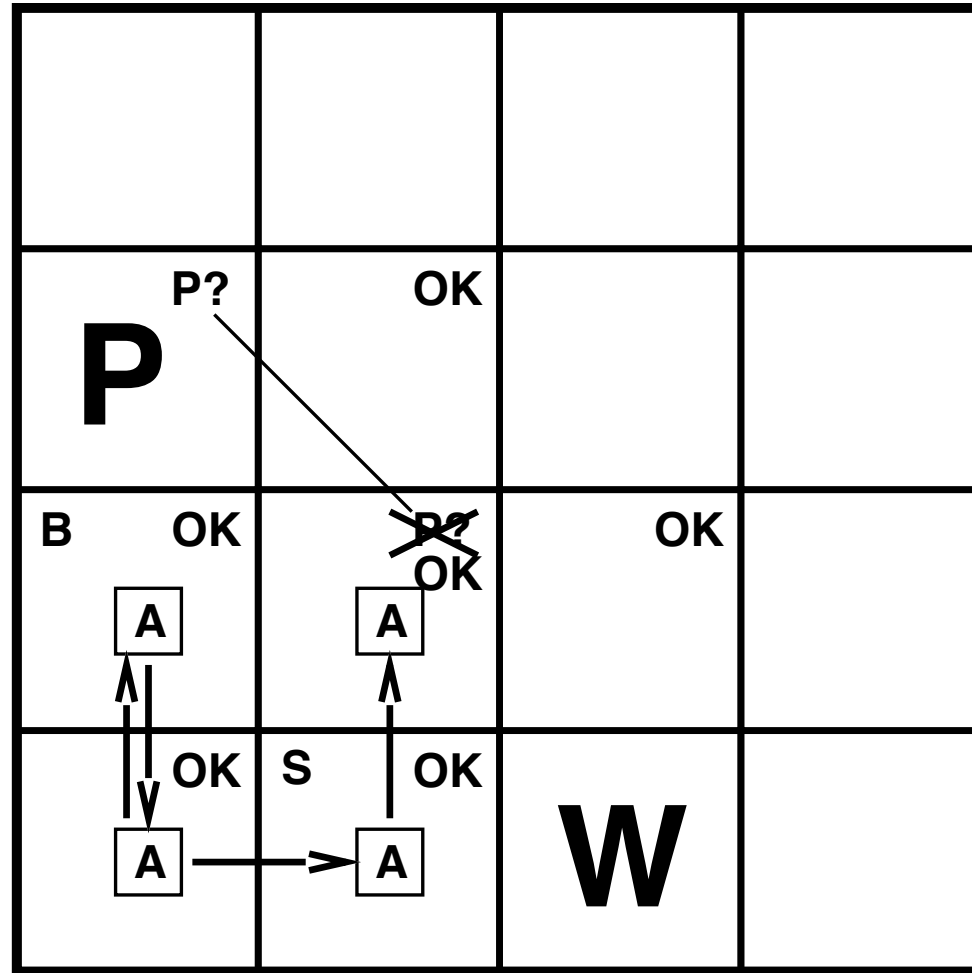
逻辑智能体：Wumpus世界



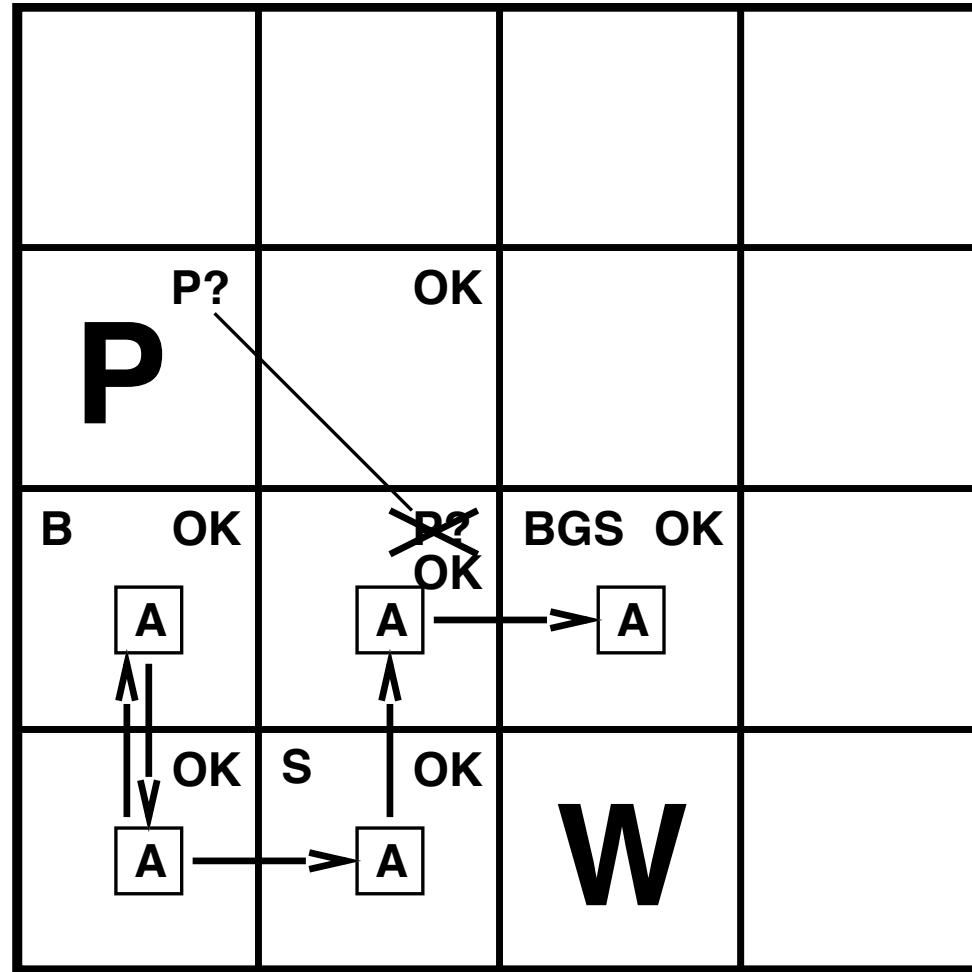
逻辑智能体：Wumpus世界



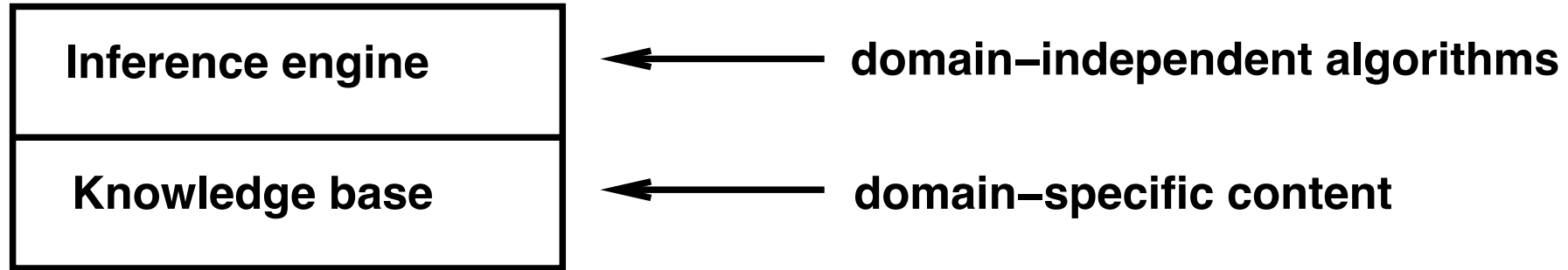
逻辑智能体：Wumpus世界



逻辑智能体：Wumpus世界



逻辑智能体 (Logic/Knowledge based Agent)



- 知识库 (Knowledge Base, KB): 一系列语句的集合, 语句用某种形式来表达 (符号)
- 感知 (Perception): 感知环境状态并转化为符号化信息, 写入知识库 (符号接地, **Symbolic Grounding**)
- 推理 (Reasoning): 知识库进行推理, 新知识加入知识库, 得出下一步行动

逻辑(Logic)

逻辑

- 语法 (syntax) : 规定所有合规的语句
- 语义 (semantics) : 定义了所有语句的含义, 或者说在每个可能世界的真值

$x + 2 \geq y$ is a sentence; $x^2 + y >$ is not a sentence

$x + 2 \geq y$ is true iff the number $x + 2$ is no less than the number y

$x + 2 \geq y$ is true in a world where $x = 7, y = 1$

$x + 2 \geq y$ is false in a world where $x = 0, y = 6$

蕴含/蕴涵(Entailment)

蕴含：一个语句在逻辑上引发另一语句， $\alpha \vdash \beta$

知识库 $KB \vdash \beta$

当且仅当

β is true in all worlds where KB is true

例如：

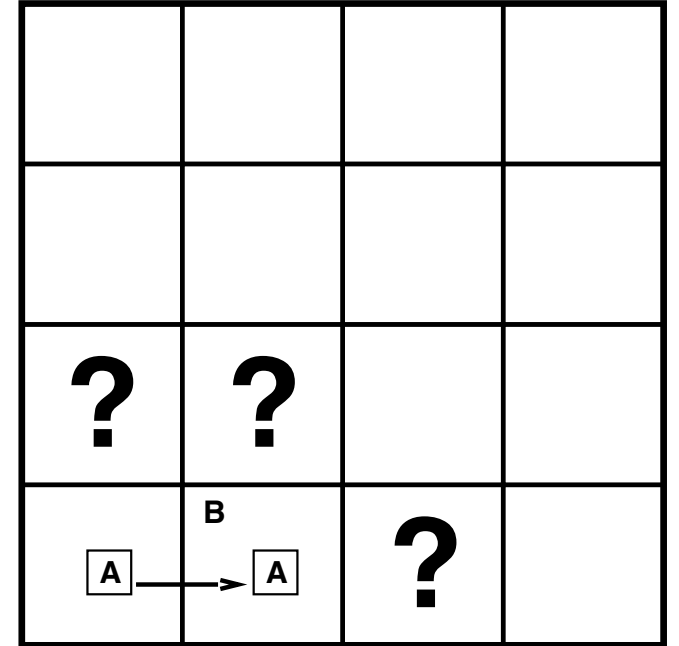
$x + y = 4$ 蕴涵 (entails) $4 = x + y$

所有奇数行没有无底洞 蕴涵 $[1,3]$ 没有无底洞

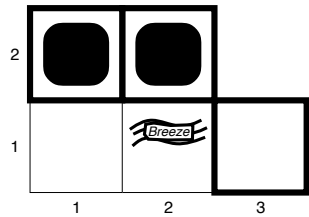
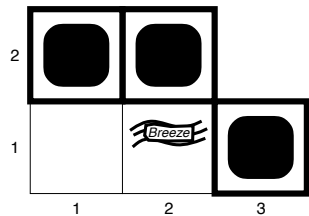
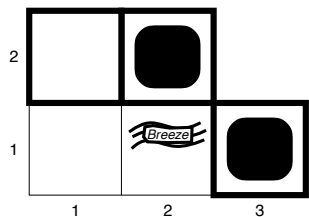
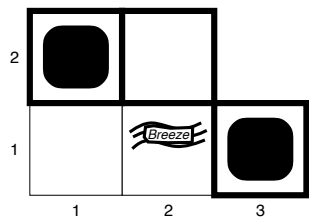
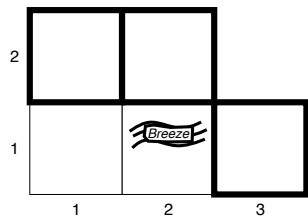
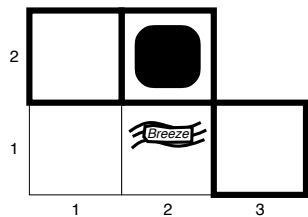
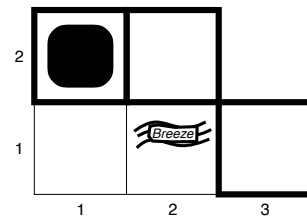
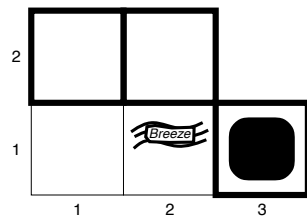
Wumpus世界

只考虑无底洞，有多少可能的世界？

3个方格每一个都可能有或者没有无底洞，总共有8个可能的世界

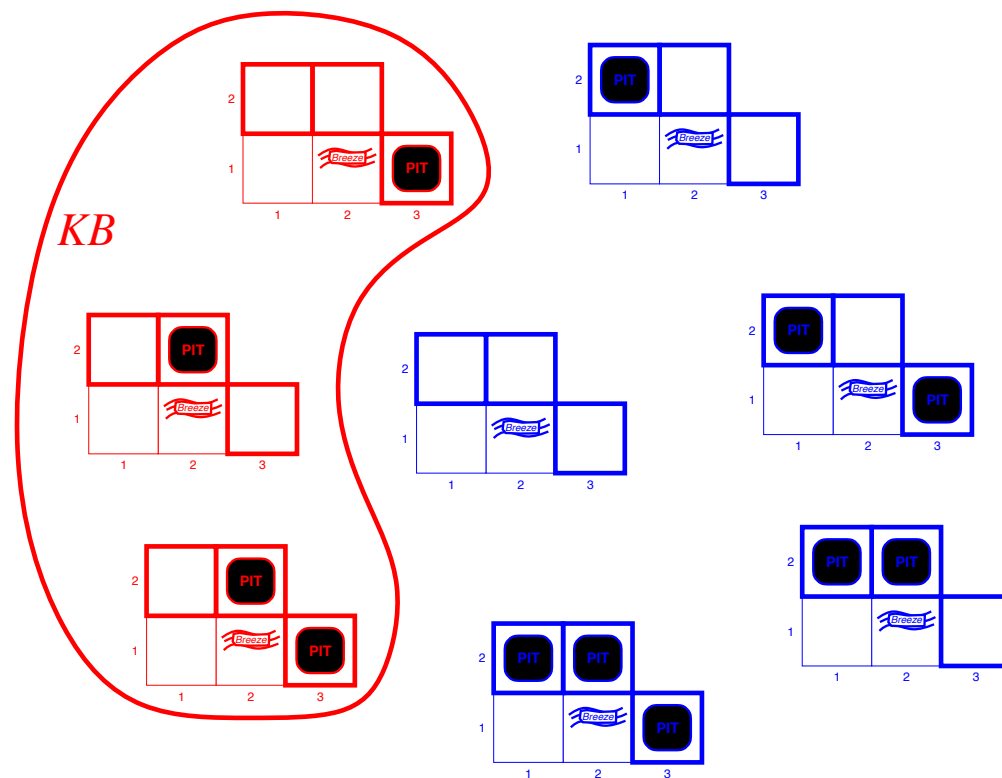


Wumpus世界



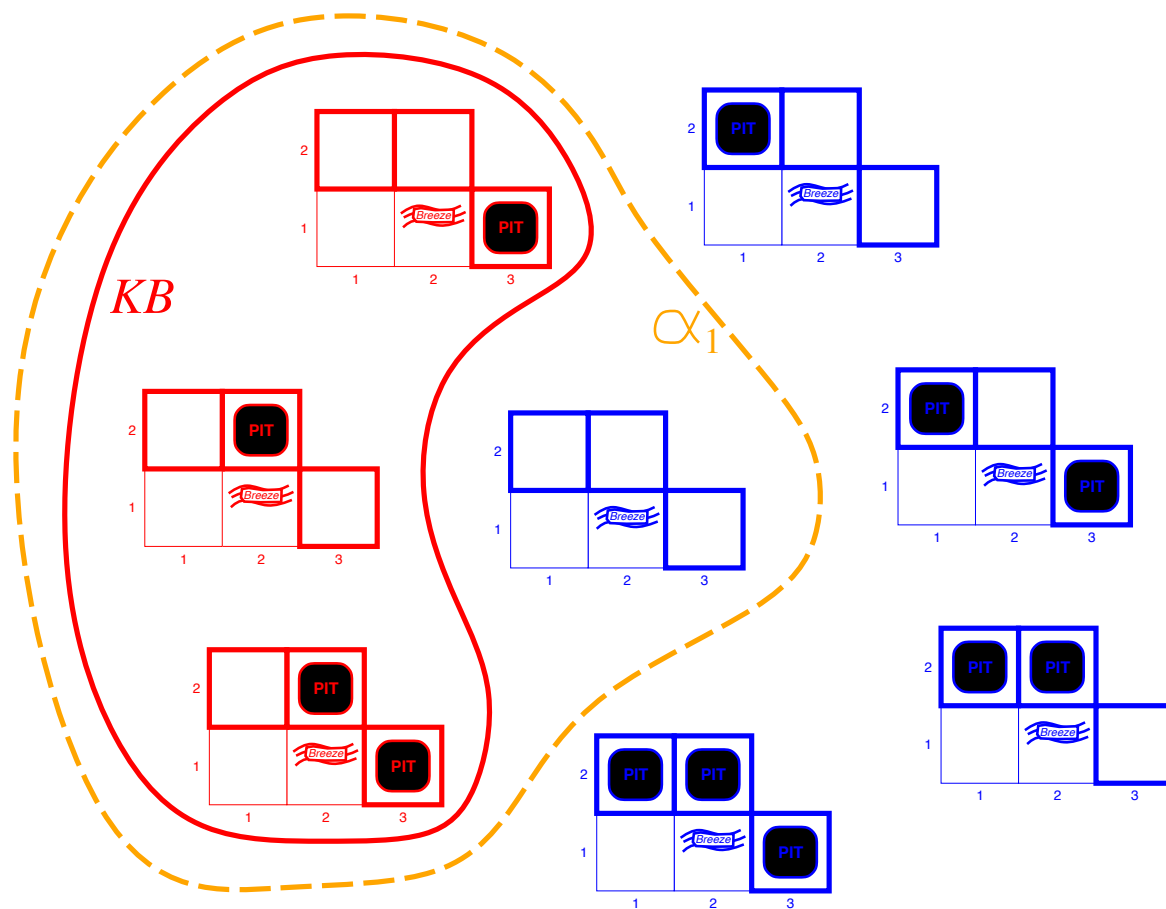
Wumpus世界：模型检验

智能体在[1, 1]什么都没有感知到，在[2, 1]感知到微风



KB = 规则 + 感知

Wumpus世界：模型检验

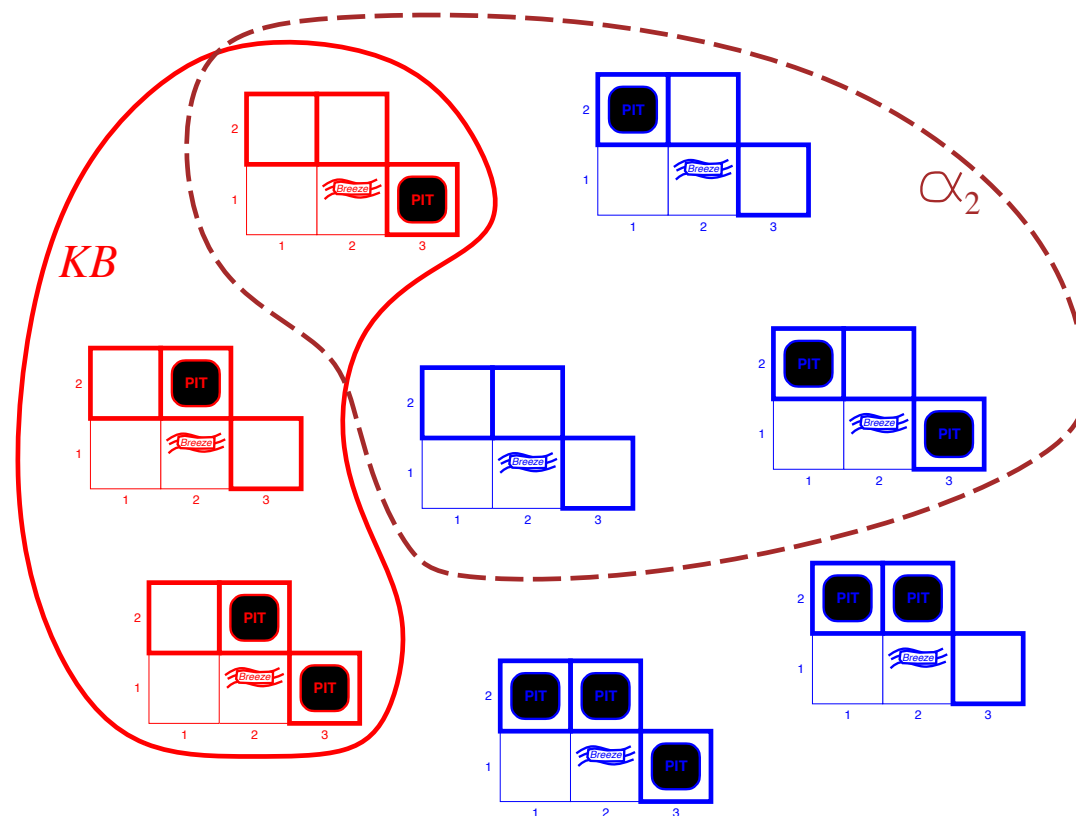


KB = 规则 + 观察

$\alpha_1 = [1,2]$ 中没有无底洞

KB \vdash α_1

Wumpus世界：模型检验



KB = 规则 + 观察

$\alpha_2 = [2,2]$ 中没有无底洞

KB \neq α_2

命题逻辑(Propositional Logic)

命题逻辑(Propositional Logic)

- 命题逻辑(Propositional Logic)是应用一套形式化规则对以符号表示的描述性陈述进行推理的系统
- 命题(或逻辑表达式): 是一个能够确定为真或假的陈述句
- 通常用字母表示命题逻辑的变量, 如 p 或者 q , 称为原语(Primitive)
- 变量通过逻辑连接符(Logical Connective)连接

逻辑连接词

逻辑连接词	表示形式	意义
与 (and)	$p \wedge q$	命题合取(conjunction), 即“ p 且 q ”
或 (or)	$p \vee q$	命题析取(disjunction), 即“ p 或 q ”
非 (not)	$\neg p$	命题否定(negation), 即“非 p ”
蕴含式 (implication)	$p \rightarrow q$	命题蕴含(implication), 即“如果 p 则 q ”
当且仅当 (bi-implication)	$p \leftrightarrow q$	命题双向蕴含(bi-implication), 即“ p 当且仅当 q ”

逻辑连接词运算规则

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
False	False	True	False	False	True	True
False	True	True	False	True	True	False
True	False	False	False	True	False	False
True	True	False	True	True	True	True

Wumpus世界：符号与规则

当 $[x, y]$ 有无底洞， $P_{x,y}$ 为真

$$R_1: \neg P_{11}$$

一个方格有微风，当且仅当其
相邻方格有无底洞

$$R_2: B_{11} \leftrightarrow (P_{12} \vee P_{21})$$

$$R_3: B_{21} \leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$$

当 $[x, y]$ 有微风， $B_{x,y}$ 为真

$$R_4: \neg B_{11}$$

$$R_5: B_{21}$$

Wumpus世界-枚举推理

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_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	<u>true</u>
false	true	false	false	false	true	false	true	true	true	true	true	<u>true</u>
false	true	false	false	false	true	true	true	true	true	true	true	<u>true</u>
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

完备

$O(2^n)$ for n symbols

余NP完全

逻辑等价

$\alpha \wedge \beta \equiv \beta \wedge \alpha$ (\wedge 的交换律)	$(\alpha \Rightarrow \beta) \equiv \neg \alpha \vee \beta$ (蕴涵消除)
$\alpha \vee \beta \equiv \beta \vee \alpha$ (\vee 的交换律)	$(\alpha \Leftrightarrow \beta) \equiv (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$ (等价消去)
$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$ (\wedge 的结合律)	$\neg(\alpha \wedge \beta) \equiv (\neg \alpha \vee \neg \beta)$ (De Morgan)
$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$ (\vee 的结合律)	$\neg(\alpha \vee \beta) \equiv (\neg \alpha \wedge \neg \beta)$ (De Morgan)
$\neg(\neg \alpha) \equiv \alpha$ (双重否定)	$(\alpha \wedge (\beta \vee \gamma)) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ (\wedge 对 \vee 的分配律)
$(\alpha \Rightarrow \beta) \equiv \neg \beta \Rightarrow \neg \alpha$ (逆否命题)	$(\alpha \vee (\beta \wedge \gamma)) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$ (\vee 对 \wedge 的分配律)

推理规则

肯定前件 (Modus Ponens)	$\frac{\alpha \Rightarrow \beta, \alpha}{\beta}$
合取消去 (And-Elimination)	$\frac{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n}{\alpha_i (1 \leq i \leq n)}$
合取导入 (And-Introduction)	$\frac{\alpha_1, \alpha_2, \dots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n}$
双重否定 (Double-Negation Elimination)	$\frac{\neg \neg \alpha}{\alpha}$

Wumpus世界：逻辑推导

(1) $R_1: \neg P_{11}$

(2) $R_2: B_{11} \leftrightarrow (P_{12} \vee P_{21})$

(3) $R_3: B_{21} \leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$

(4) $R_4: \neg B_{11}$

(5) $R_5: B_{21}$

证明： $\neg P_{12}$ ，即[1,2]中没有无底洞

- 对 R_2 进行等价消去：

$$R_6: B_{11} \rightarrow (P_{12} \vee P_{21}) \wedge (P_{12} \vee P_{21}) \rightarrow B_{11}$$

- 对 R_6 使用合取消去，得到

$$R_7: (P_{12} \vee P_{21}) \rightarrow B_{11}$$

- 逆否命题： $R_8: \neg B_{11} \rightarrow \neg(P_{12} \vee P_{21})$

- $R_9: \neg(P_{12} \vee P_{21})$

- $R_{10}: \neg P_{12} \wedge \neg P_{21}$

定义初始状态、动作、动作转移、目标，应用搜索算法完成证明

归结/消解(resolution)

命题逻辑中的表达式，如果不包含蕴含式、合取式、双重否定，就称为子句形式

归结：通过子句结合，消去互补文字(即各自是对方的否定)

单项消解或单项归结 (Unit Resolution)	$\frac{\alpha \vee \beta, \neg\beta}{\alpha}$
消解或归结 (Resolution)	$\frac{\alpha \vee \beta, \neg\beta \vee \gamma}{\alpha \vee \gamma}$

归结与任意完备的搜索算法结合，可以得到一个完备的推理算法

应用归结法进行证明

例1: 如下命题集是否是可满足的?

(1) $\alpha \vee \beta$

(2) $\neg\alpha \vee \beta$

(3) $\alpha \vee \neg\beta$

(4) $\neg\alpha \vee \neg\beta$

- 已知 $\alpha \vee \beta$
- 已知 $\neg\alpha \vee \beta$
- 推出: β
- 已知 $\alpha \vee \neg\beta$
- 已知 $\neg\alpha \vee \neg\beta$
- 推出: $\neg\beta$

应用归结法进行证明

例2: 如下命题集是否是可满足的?

(1) $\alpha \vee \gamma$

(2) $\neg\beta \vee \gamma$

(3) $\neg\gamma \vee \alpha$

(4) $\neg\alpha \vee \beta$

(5) $\neg\alpha \vee \neg\gamma$

- 已知 $\alpha \vee \gamma$
- 已知 $\neg\gamma \vee \alpha$
- 推出: α
- 已知 $\neg\alpha \vee \beta$
- 已知 α
- 推出: β
- 已知 $\neg\beta \vee \gamma$
- 已知 β
- 推出: γ
- 已知 $\neg\alpha \vee \neg\gamma$
- 推出: $\neg\alpha$

应用归结法进行证明

例3: 已知如下命题成立,

请证明命题 γ 是成立的

(1) $\alpha \vee \beta$

(2) $\alpha \rightarrow \gamma$

(3) $\beta \rightarrow \gamma$

- 已知 $\alpha \vee \beta$
- 对 (2) 进行蕴含消除: $\neg\alpha \vee \gamma$
- 对 (3) 进行蕴含消除: $\neg\beta \vee \gamma$
- $(\neg\alpha \vee \gamma) \wedge (\neg\beta \vee \gamma)$
- 使用分配率: $(\neg\alpha \wedge \neg\beta) \vee \gamma$
- 使用De Morgan定律: $\neg(\alpha \vee \beta) \vee \gamma$
- γ

合取范式

归结规则仅适用于子句，对于所有的命题逻辑，能否实现完备的推断过程？

命题逻辑的所有语句逻辑上都等价于子句合取式！

形式为子句合取式的语句被称为合取范式(Conjunctive Norm Form, CNF)

例：求 $\neg(\alpha \rightarrow \beta) \vee \neg\gamma$ 的合取范式

$$\neg(\alpha \rightarrow \beta) \vee \neg\gamma$$

$$\Leftrightarrow \neg(\neg\alpha \vee \beta) \vee \neg\gamma$$

$$\Leftrightarrow (\alpha \vee \neg\gamma) \wedge (\neg\beta \vee \neg\gamma) \text{ (合取范式)}$$

合取范式

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

1. Eliminate \Leftrightarrow , replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

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

2. Eliminate \Rightarrow , replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

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

3. Move \neg inwards using de Morgan's rules and double-negation:

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

4. Apply distributivity law (\vee over \wedge) and flatten:

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

归结算法

证明, $KB \rightarrow \alpha$, 采用反证法

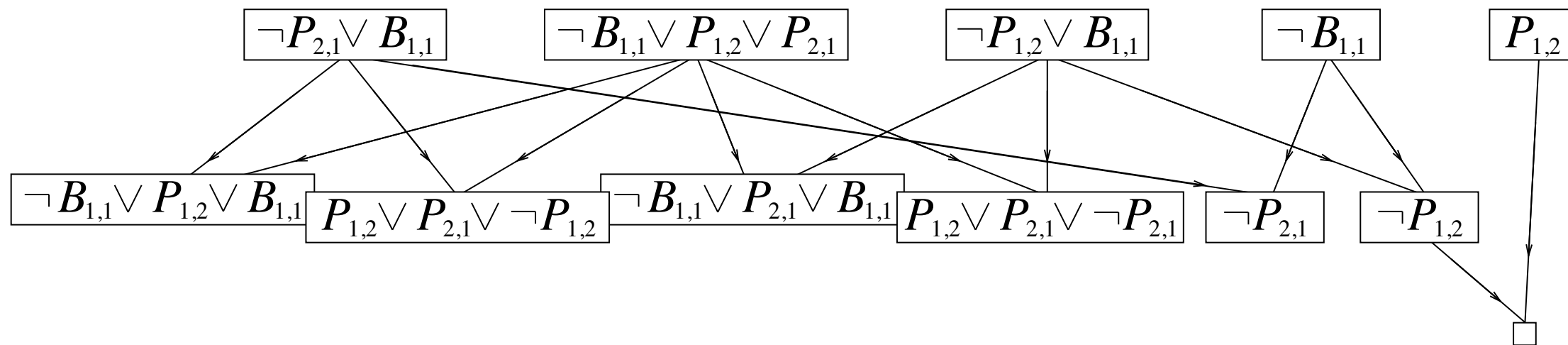
- 假设KB为真, 目标 α 为假, 即 $(KB \wedge \neg\alpha)$ 是不可满足的
- 将 $(KB \wedge \neg\alpha)$ 转换为CNF
- 不断应用归结规则, 产生新的子句
- 直到推导出矛盾

归结算法应用于Wumpus世界

当智能体位于[1, 1]时，该处没有微风，因此相邻的方格没有无底洞，此时的知识库：

$$KB = (B_{11} \leftrightarrow (P_{12} \vee P_{2,1}) \wedge \neg B_{11})$$

要证明 $\neg P_{12}$ ，将 $KB \wedge \neg\neg P_{12}$ 转换为CNF



有些归结步骤是无意义的，比如 $\neg B_{11} \vee P_{21} \vee B_{11}$

扩展：SAT与SMT

- 归结算法是现代SAT/SMT求解器的核心方法之一(改进版本)
- 布尔可满足性问题(SAT)
 - 给定一个布尔公式，判断是否存在一个变量赋值，使公式为真
- SAT 是第一个被证明为 **NP-Complete** 的问题，几乎所有组合问题都能规约到 SAT
- SMT (Satisfiability Modulo Theories)
 - SMT 是在 SAT 的基础上加入数学相关的（如算术、数组、位向量、几何等）
可满足性问题

扩展：SAT与SMT

```
In [1]: from z3 import *
```

```
In [2]: # 1. 声明数学变量 (Z3 会自动处理底层的位向量或逻辑转化)
x = Int('x')
y = Int('y')
```

```
In [3]: # 2. 构建求解器
s = Solver()
```

```
In [4]: # 3. 添加数学命题/约束
# 例如: 证明  $x + y > 10$  且  $x > 5$ ,  $y < 5$  是否有解
s.add(x + y > 10)
s.add(x > 5)
s.add(y < 5)
```

```
In [5]: # 4. 求解
if s.check() == sat:
    print("Found a model:", s.model())
else:
    print("No solution (Proof by contradiction succeeded if we were checking negation)")
```

```
Found a model: [x = 7, y = 4]
```

霍恩子句与确定子句

许多实际情形并不需要用到归结的全部能力，一些真实世界的知识库中的语句满足某些限制，这使得它们可以使用更为受限而更高效的推断算法

- **确定子句 (definite clause):** 文字的析取式，其中只有一个为正文字

例如： $\neg B_{11} \vee P_{12} \vee P_{21}$ 不是确定子句， $\neg B_{11} \vee \neg P_{12} \vee P_{21}$ 是确定子句

- **霍恩子句 (horn clause):** 文字的析取式，其中最多只有一个为正文字
- 每个确定子句都可以写成一个蕴涵式

$$\neg B_{11} \vee \neg P_{12} \vee P_{21}$$

$$(B_{11} \wedge P_{12}) \rightarrow P_{21}$$

前向推理

- 事实导向的推理(data driven)
- 找到前提满足知识库的规则，并把结论加入知识库
- 直到查询被添加或者无法进一步推断

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

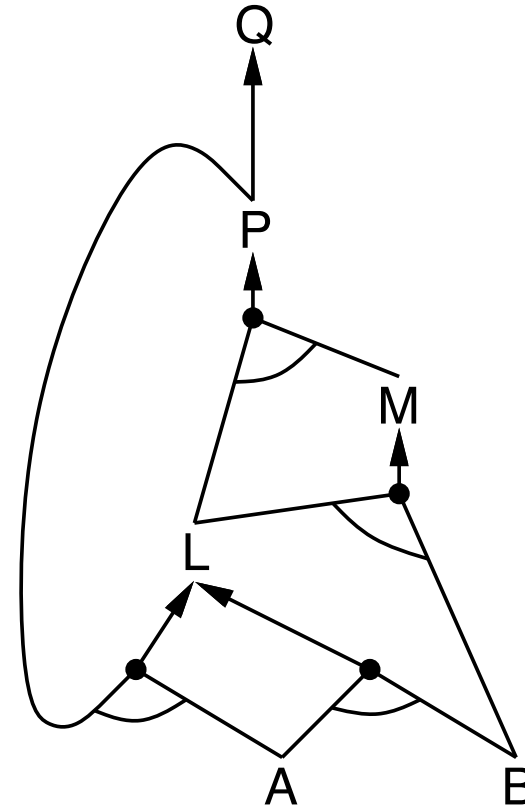
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

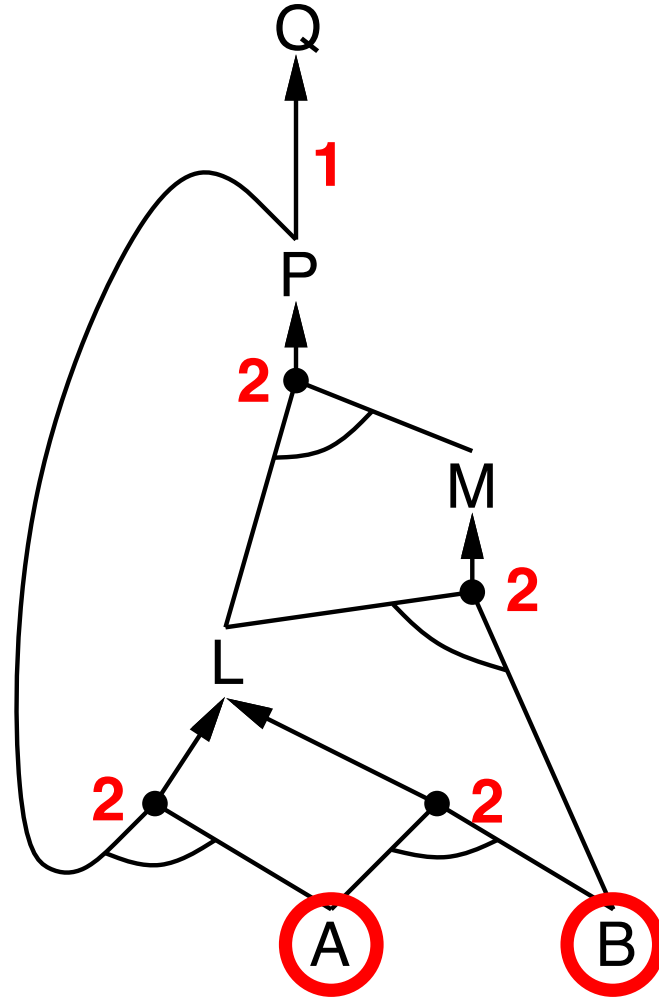
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

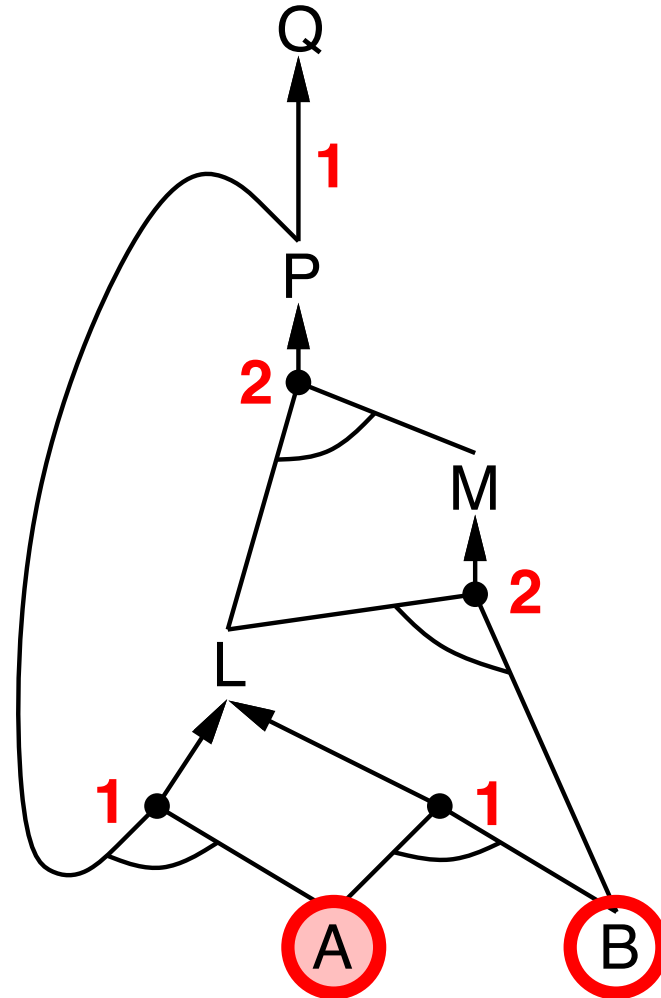
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

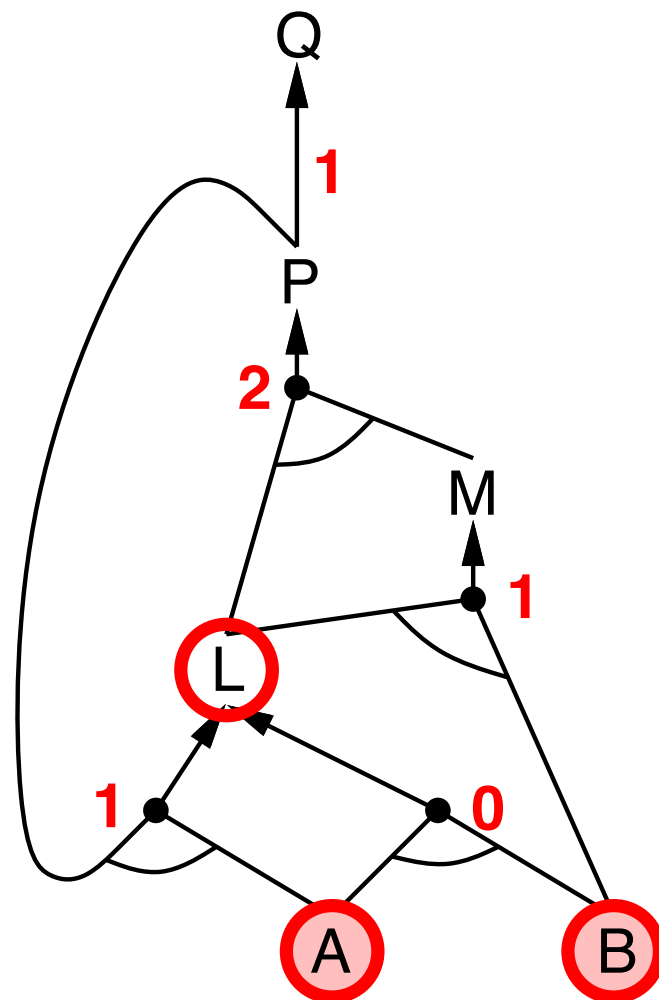
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

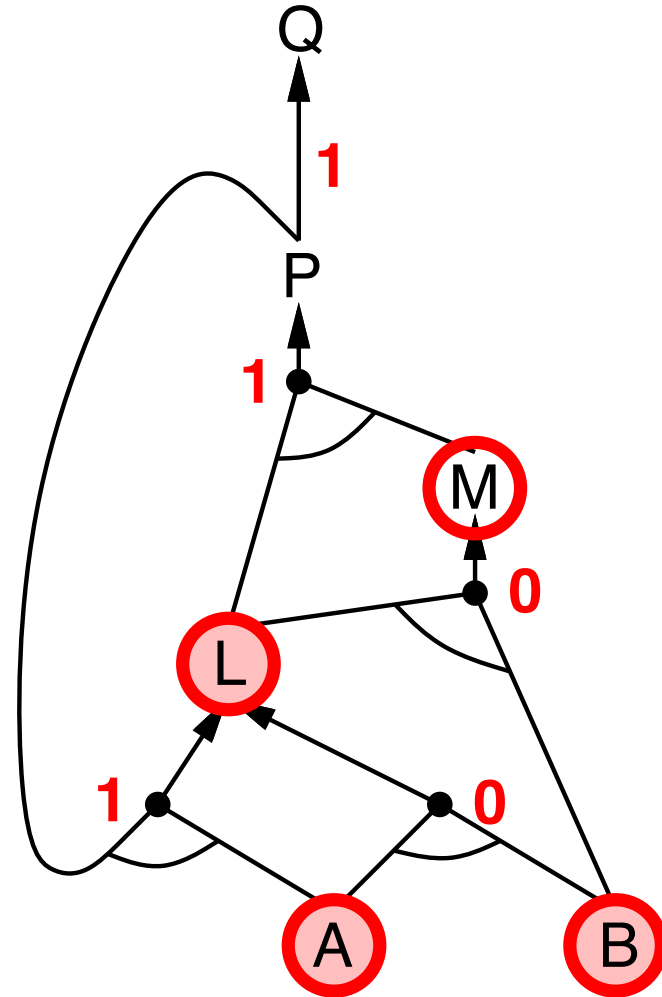
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

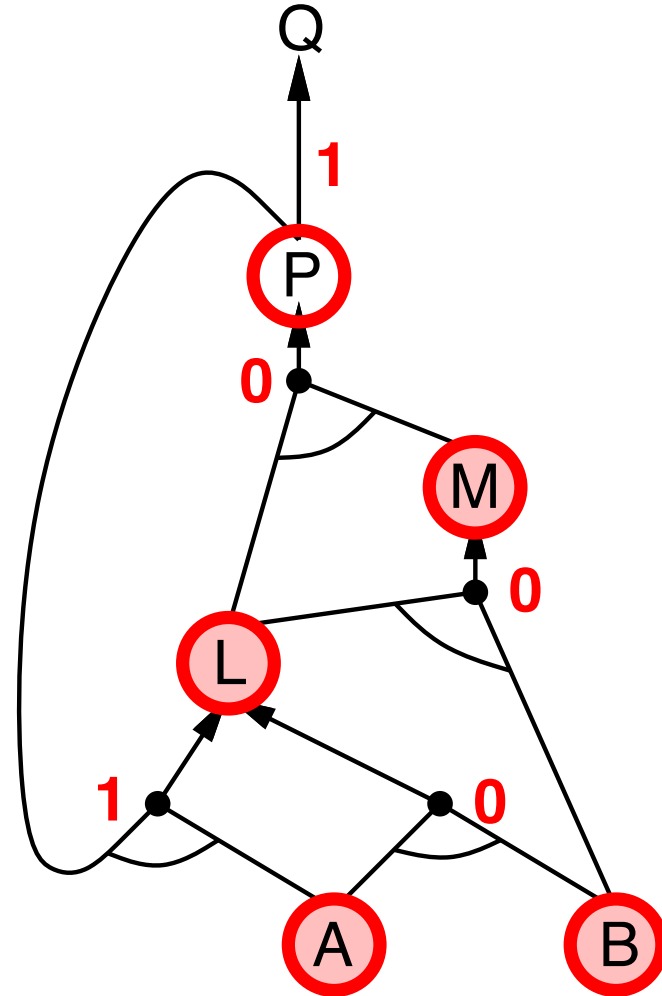
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

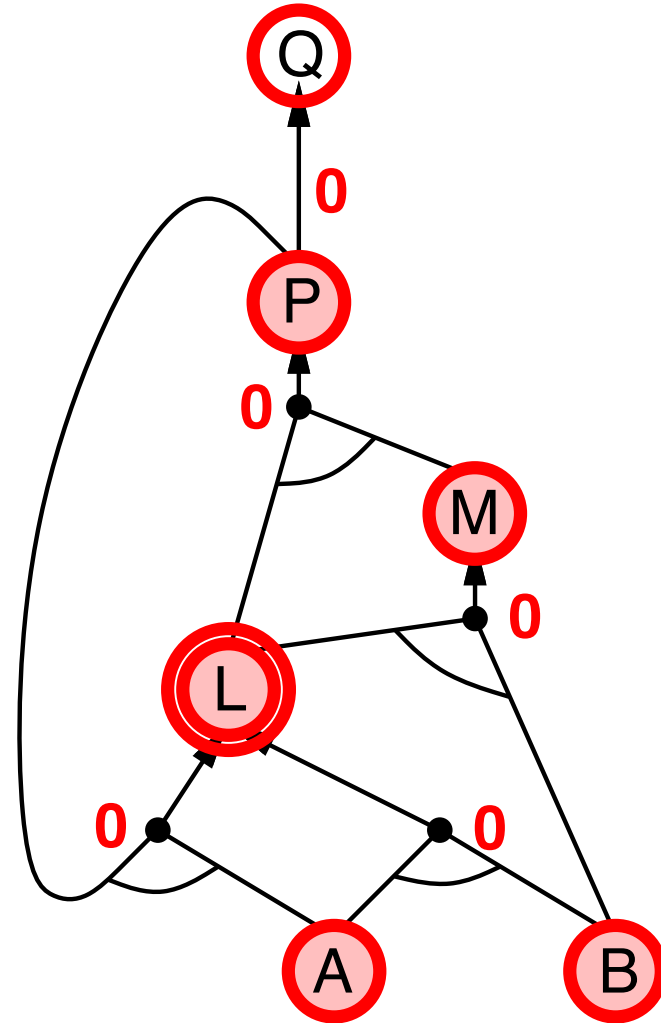
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



前向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

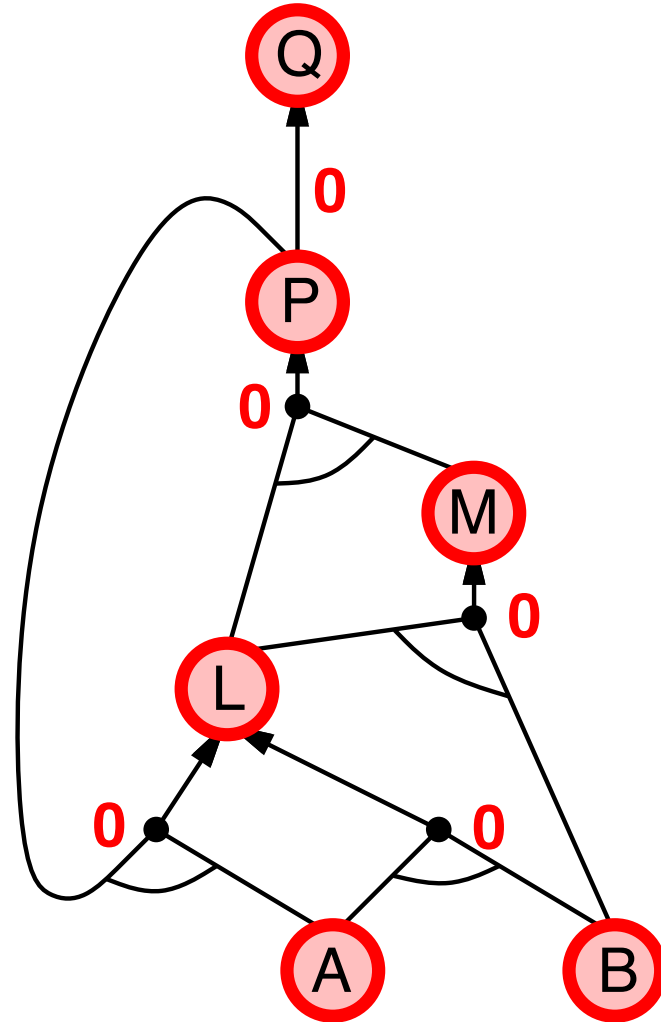
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

目标导向的推理(Goal-Driven)

- 从查询 q 开始, 反向运行, 不断在知识库中查询结论为 q 的蕴涵式
- 如果这些蕴涵式的所有前提都可以证明为真, 则 q 为真

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

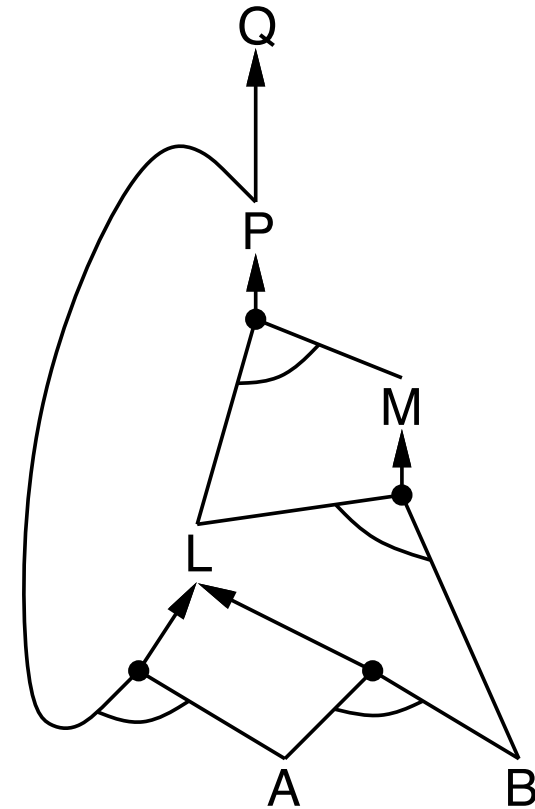
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

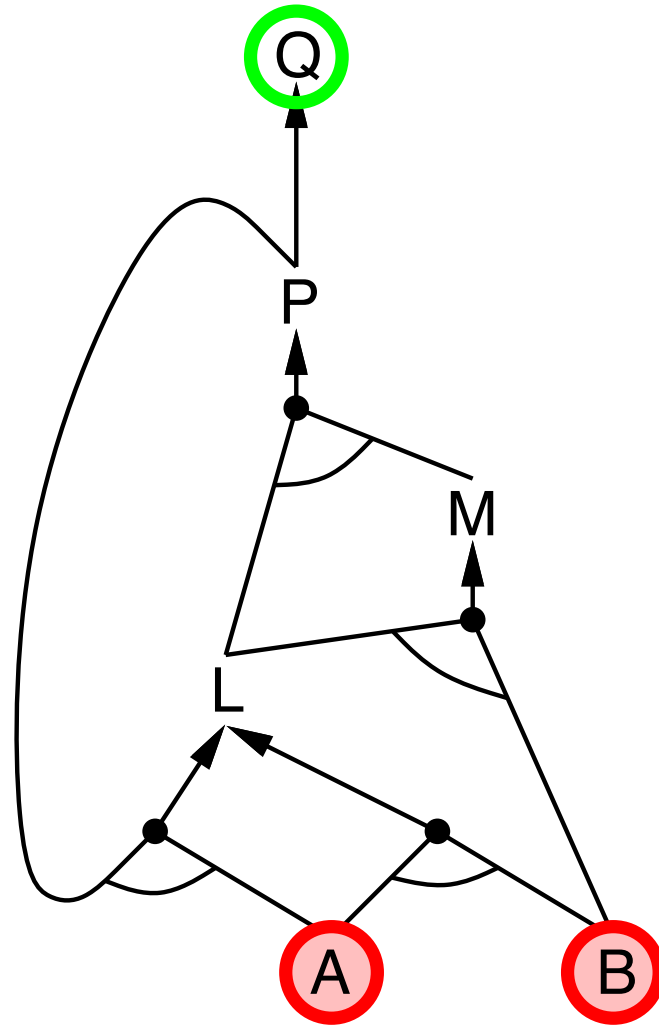
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

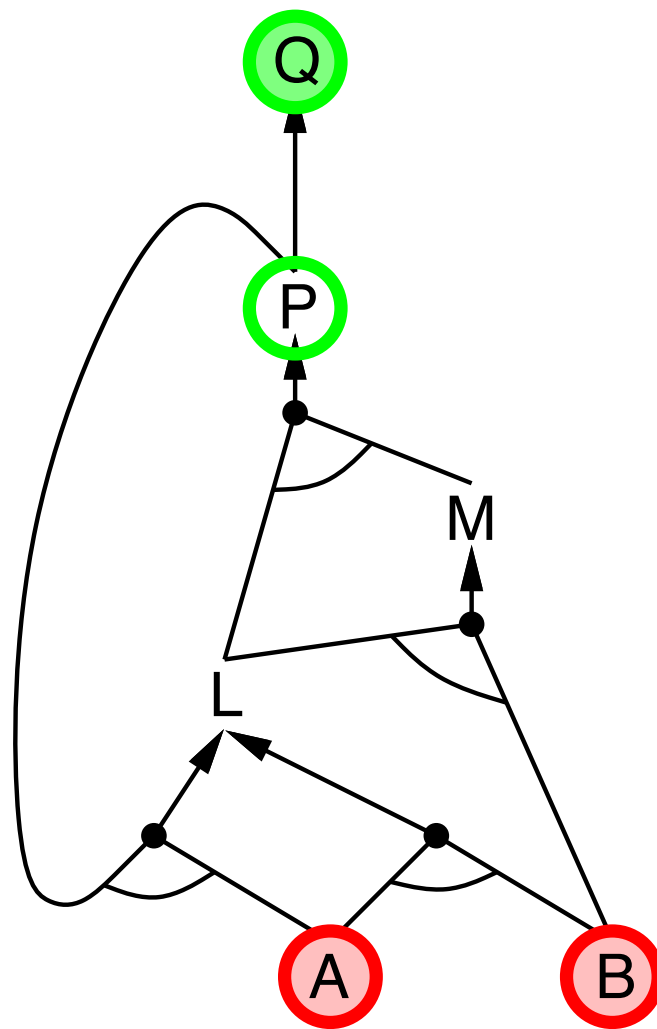
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

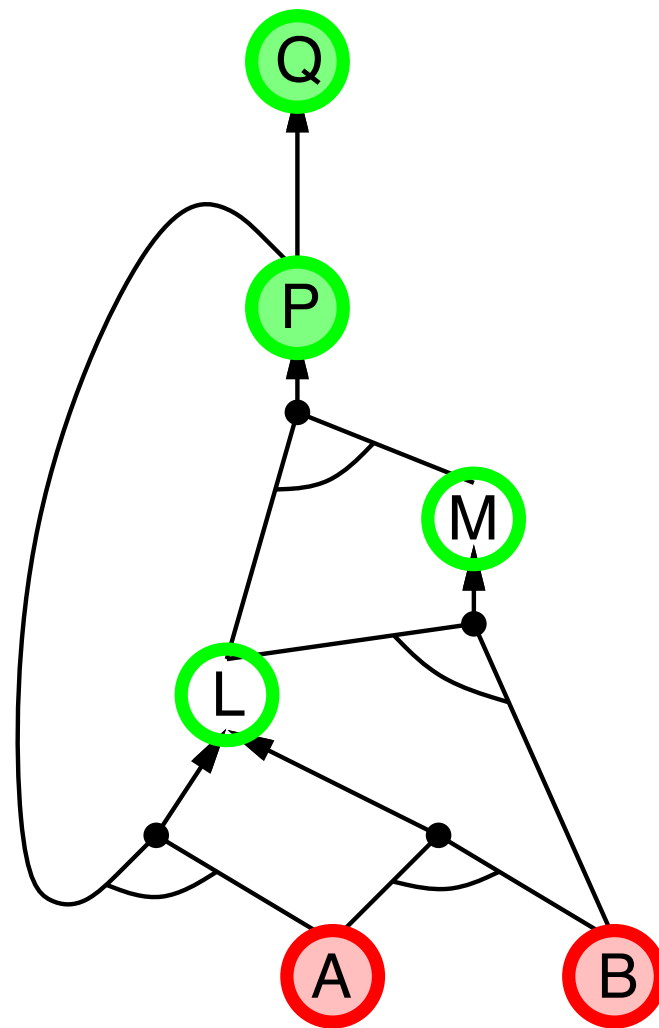
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

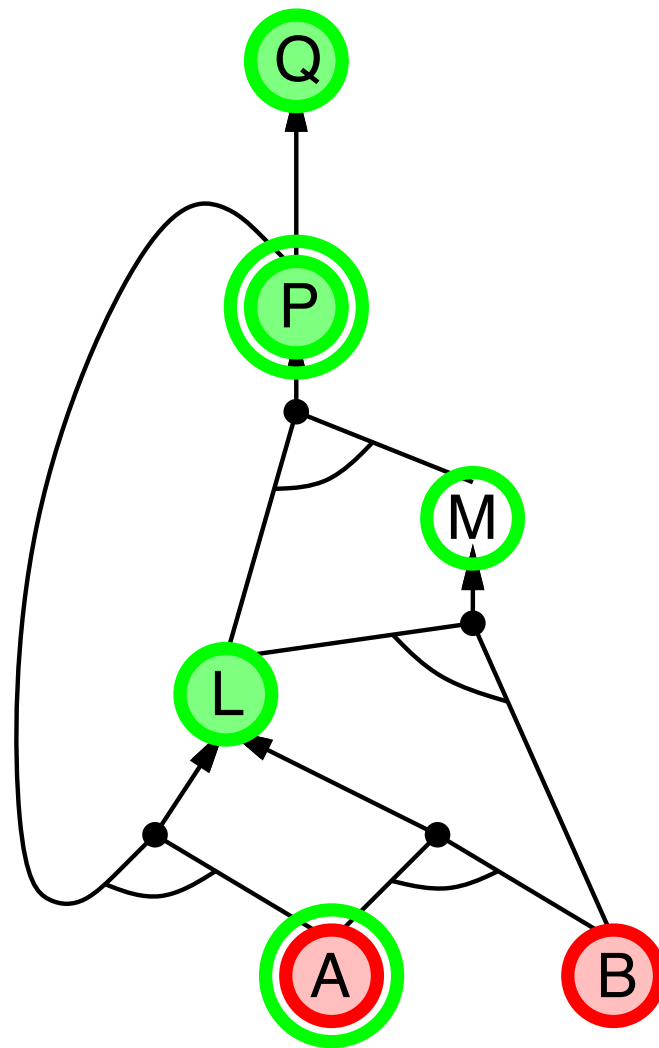
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

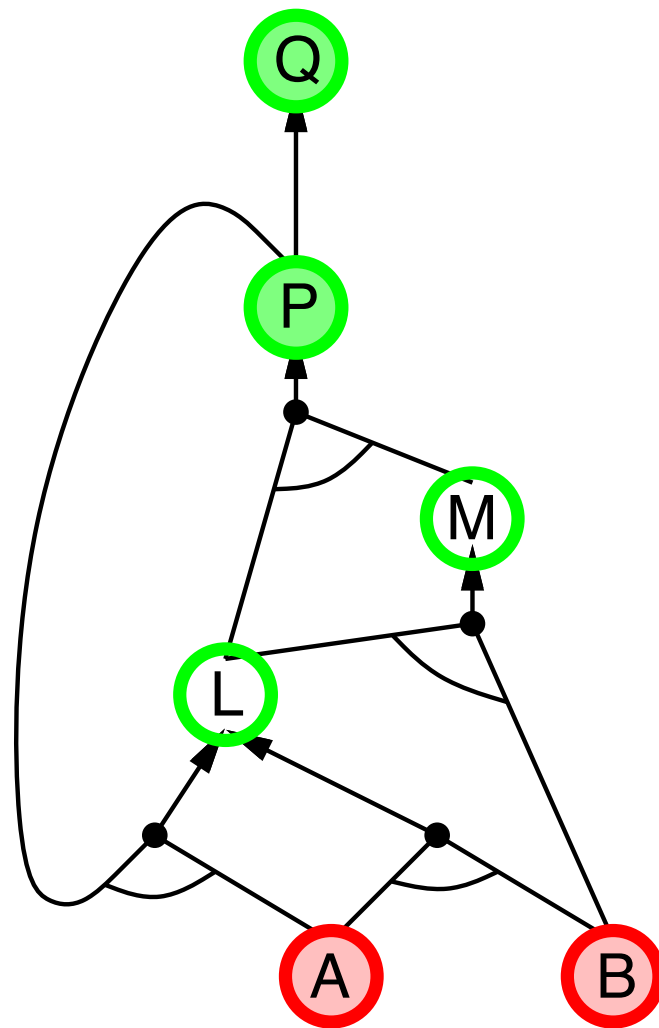
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

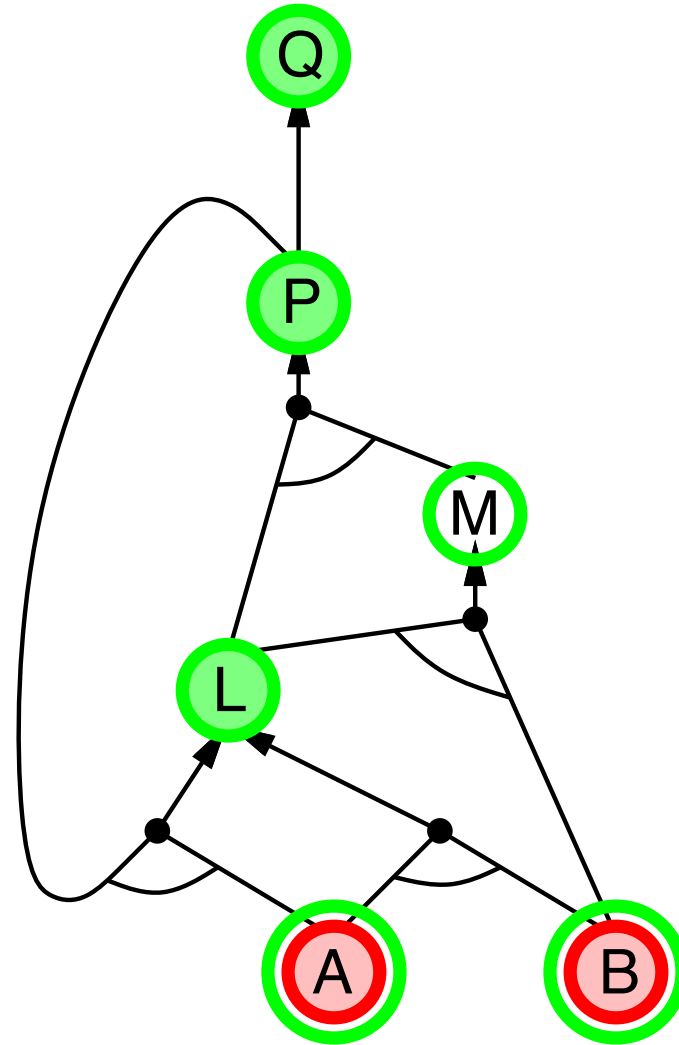
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

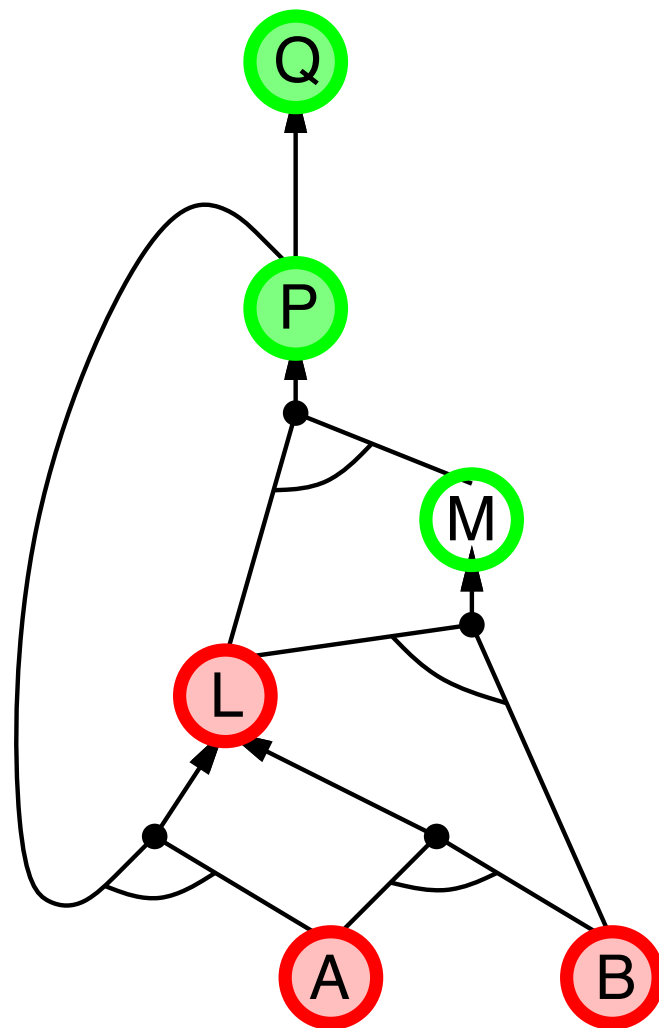
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

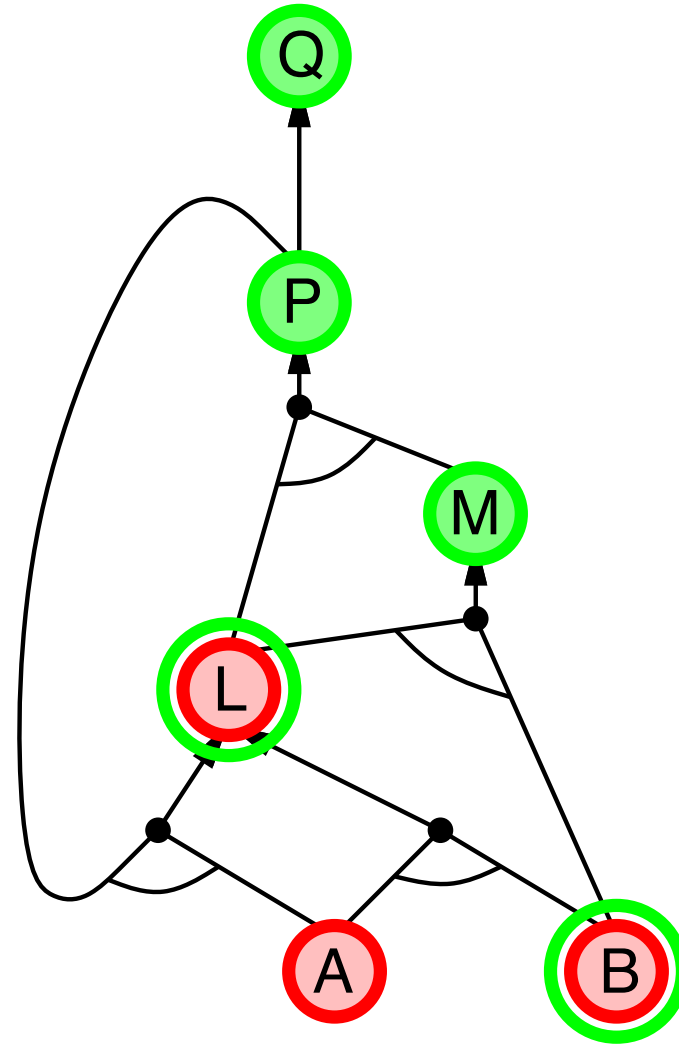
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

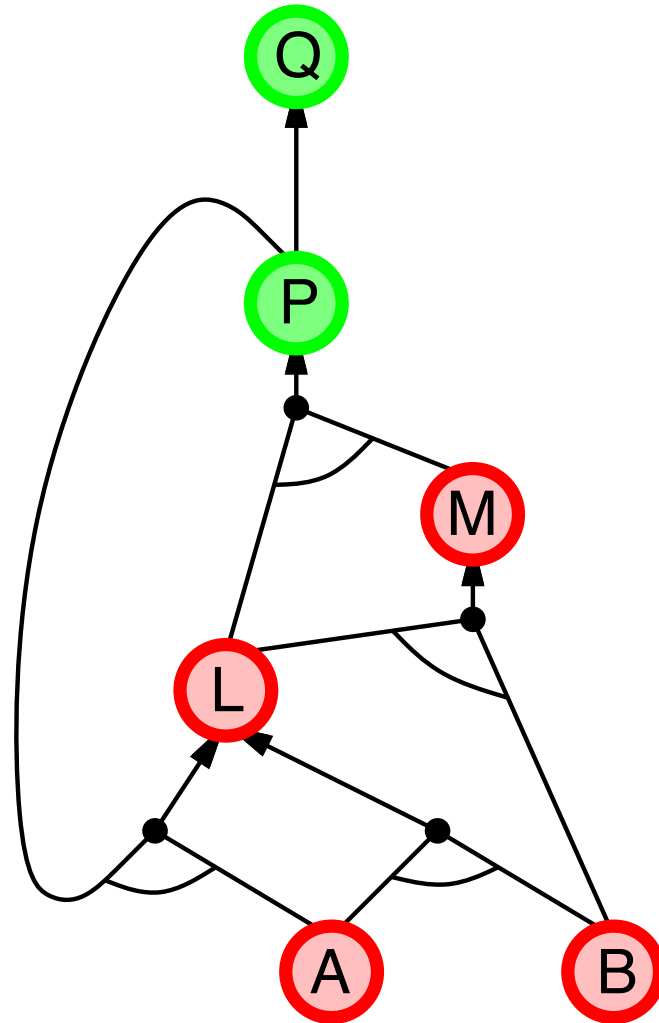
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



后向推理

$$P \Rightarrow Q$$

$$L \wedge M \Rightarrow P$$

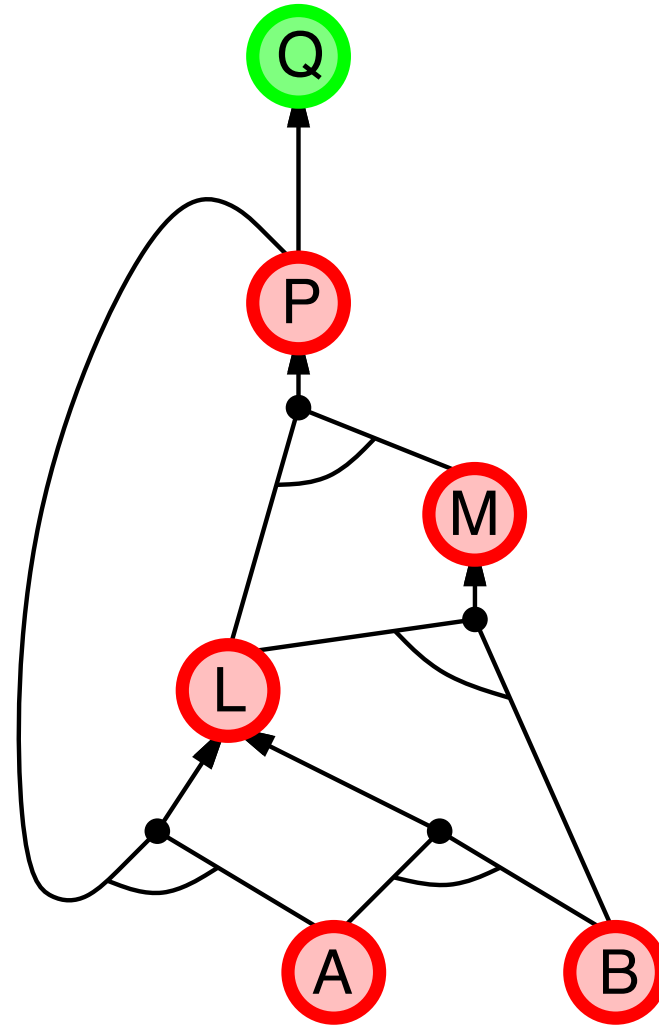
$$B \wedge L \Rightarrow M$$

$$A \wedge P \Rightarrow L$$

$$A \wedge B \Rightarrow L$$

A

B



命题逻辑的局限性

□ 命题逻辑的局限性：在命题逻辑中，每个陈述句是最基本的单位(即原子命题)，无法对原子命题进行分解，因此在命题逻辑中，不能表达局部与整体、一般与个别的关系

□ 例如，对于苏格拉底论断

✓ α : 所有的人总是要死的

✓ β : 苏格拉底是人

✓ γ : 所以苏格拉底是要死的

无法在命题逻辑基础上推导出： $\alpha \wedge \beta \rightarrow \gamma$

虽然是正确的，但是无法通过命题逻辑来进行推理判断