人工智能与机器学习

叶琦

工业控制研究所

杭州·浙江大学·2022

回顾--问题求解

- **问题求解智能体** —— 一类基于目标的智能体
  - 问题的形式化
  - 问题举例
  - 问题的类型
- **基本的搜索算法**
  - 盲目搜索
    - 广度优先、代价一致、深度优先、有限深度、迭代加深
  - 信息搜索
    - 最佳优先、A*、爬山法、模拟退火、遗传算法、
  - 对抗搜索
    - 极小极大值算法、 α–β剪枝、概率博弈、不完全信息博弈

知识表示与推理

- 知识：逻辑Agent（基于知识的Agent）
- 知识表示：一阶谓词逻辑
- 推理：一阶谓词逻辑推理

第6章 逻辑智能体

- 基于知识的智能体
  - 总体的智能体设计
- 妖兽(wumpus)世界：一个基于知识的智能体
- 逻辑
- 命题逻辑
- 命题逻辑的推理模式
- 基于命题逻辑的智能体（确定的知识）

6.1 基于知识的智能体

- **逻辑智能体是基于知识的智能体，它采用推理过程来得到关于新世界的表示，并且用这些新表示推到下一步做什么。**
- **基于知识的智能体，从通用的形式表达的知识中获益，通过对信息的组合和再组合，以适应各种用途。**
  - **在部分可观察的环境，**能够将常识和当前的感知结合起来，在选择行动之前推导出当前状态的隐藏部分
  - **自然语言理解：**要对隐含状态即说话者的意图进行推理
  - 具有很好的灵活性：
    - 接受新的任务：能够接受以明确描述模板的形式表示的新任务
    - 适应环境的变化：通过被告知或主动学习环境的新知识，从而快速获得能力，并可以通过更新相关知识来适应环境的变化

基于知识的智能体

- 核心构件是其**知识库（KB）**
  - 知识库是一个语句集合，语句由知识表示语言来表达，表示了关于世界的某些断言，包括问题的一些背景知识
- 将新语句添加到知识库以及查询目前所知内容的途径
  - **TELL**和**ASK**：可能涉及推理
  - 当ASK知识库一个问题时，答案必须遵循（follow）事先被告知的知识库的内容
- 基于知识的智能体也是用**感知信息**作为输入，并返回一个**行动**

一个通用的基于知识的智能体

```
function KB-AGENT(percept) returns an action
   static: KB, a knowledge base
           t, a counter, initially 0, indicating time

   TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))
   action ← ASK(KB, MAKE-ACTION-QUERY(t))
   TELL(KB, MAKE-ACTION-SENTENCE(action, t))
   t ← t + 1
   return action
```

- TELL→ASK→TELL
- 表示语言的细节隐含于三个MAKE-xxx函数中
- 推理机制的细节隐藏于TELL和ASK中
- KB的构造
  - TELL（陈述性 vs 过程性），涉及知识表示方法
  - 智能体的自我学习实现自治

7.2 妖兽世界的PEAS

Wumpus世界是由多个房间组成，用通道连接起来的洞穴。在洞穴某处隐藏着妖兽Wumpus，它会吃掉进入它所在房间的人，智能体可以射杀Wumpus，但只有一次机会。某些房间有陷阱，进入陷阱房间，智能体将被陷阱吞噬。有些房间有金子。

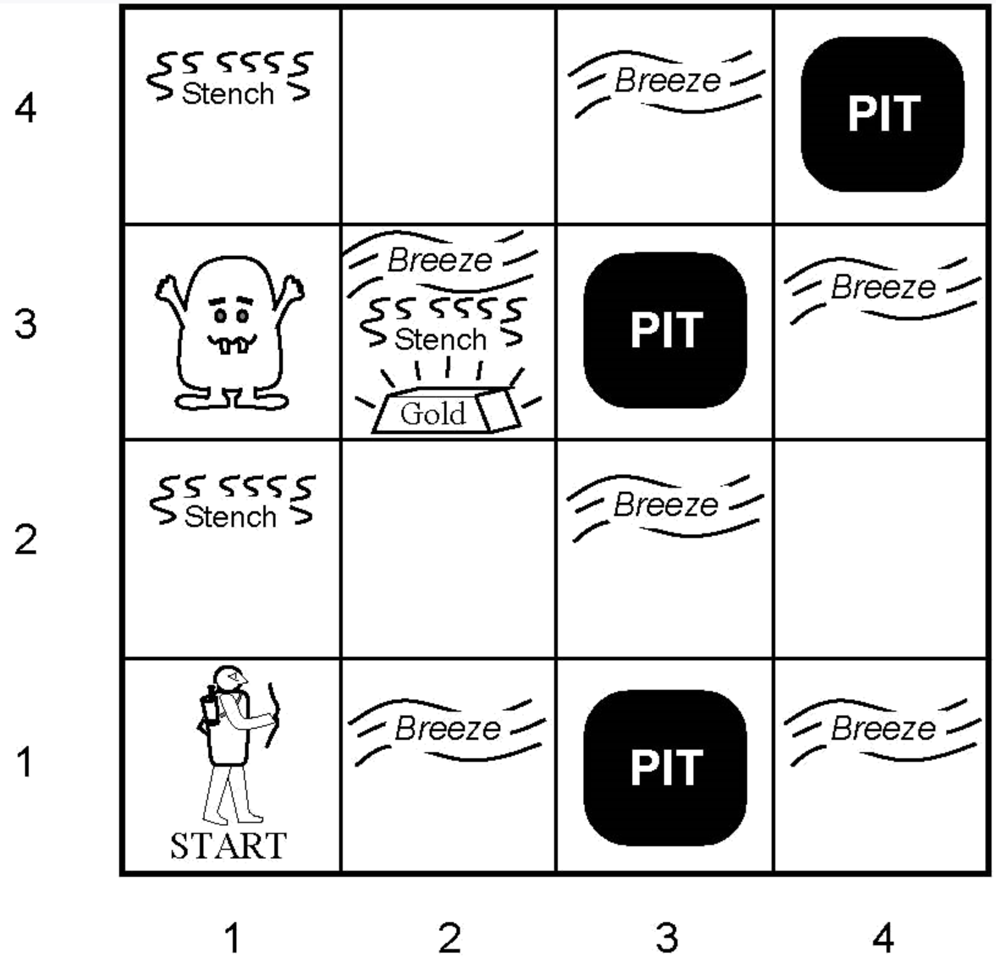
- **性能指标P**

✓ 得到金子 +1000

✓ 死亡 -1000

✓ 每个动作 -1

✓ 使用弓箭 -10

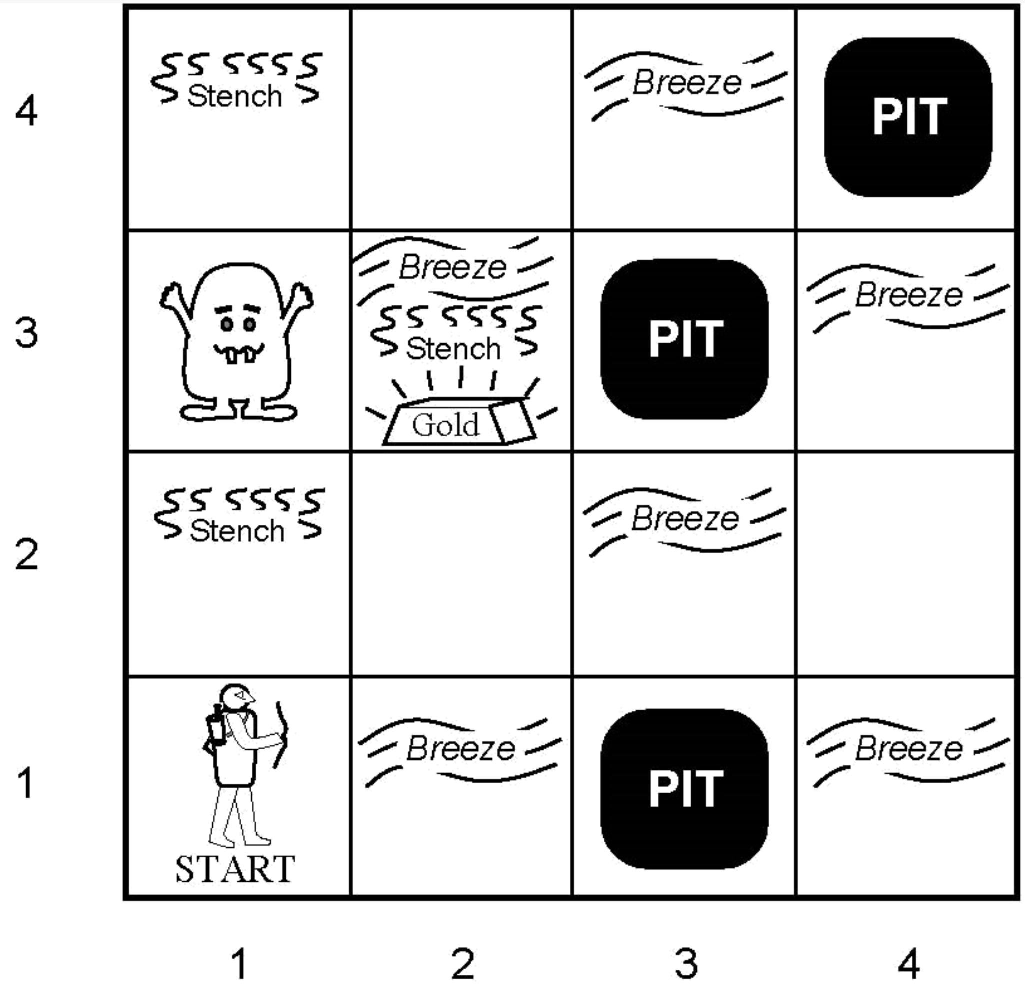


妖兽世界的PEAS

- **环境E**

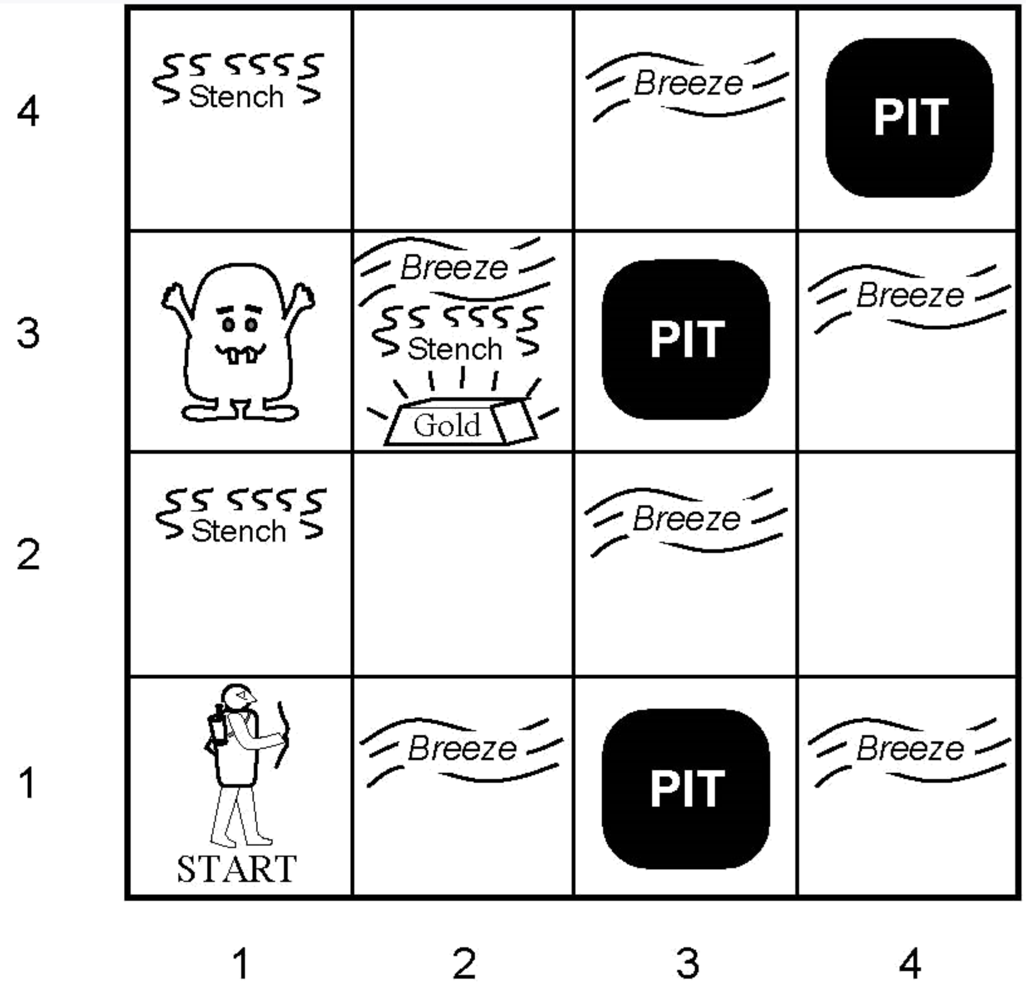
  4×4的空间

  - 从[1,1]开始，面向右方
  - 金子的位置随机生成
  - 妖兽的位置随机生成
  - 每格有陷阱的概率为0.2



妖兽世界的PEAS

- **动作A**
  - TurnLeft / TurnRight / Forward：进入有陷阱或妖兽的位置会死，遇到墙壁阻挡则不能成功。
  - Grab：在有金子的位置拾起金子
  - Shoot：射死面向方向的妖兽，或者打到墙上。只有第一箭有效。
  - Climb：爬出洞口，只能从[1,1]爬出



妖兽世界的PEAS

- **感知S**

有5个传感器：

✓妖兽及相邻位置有臭气

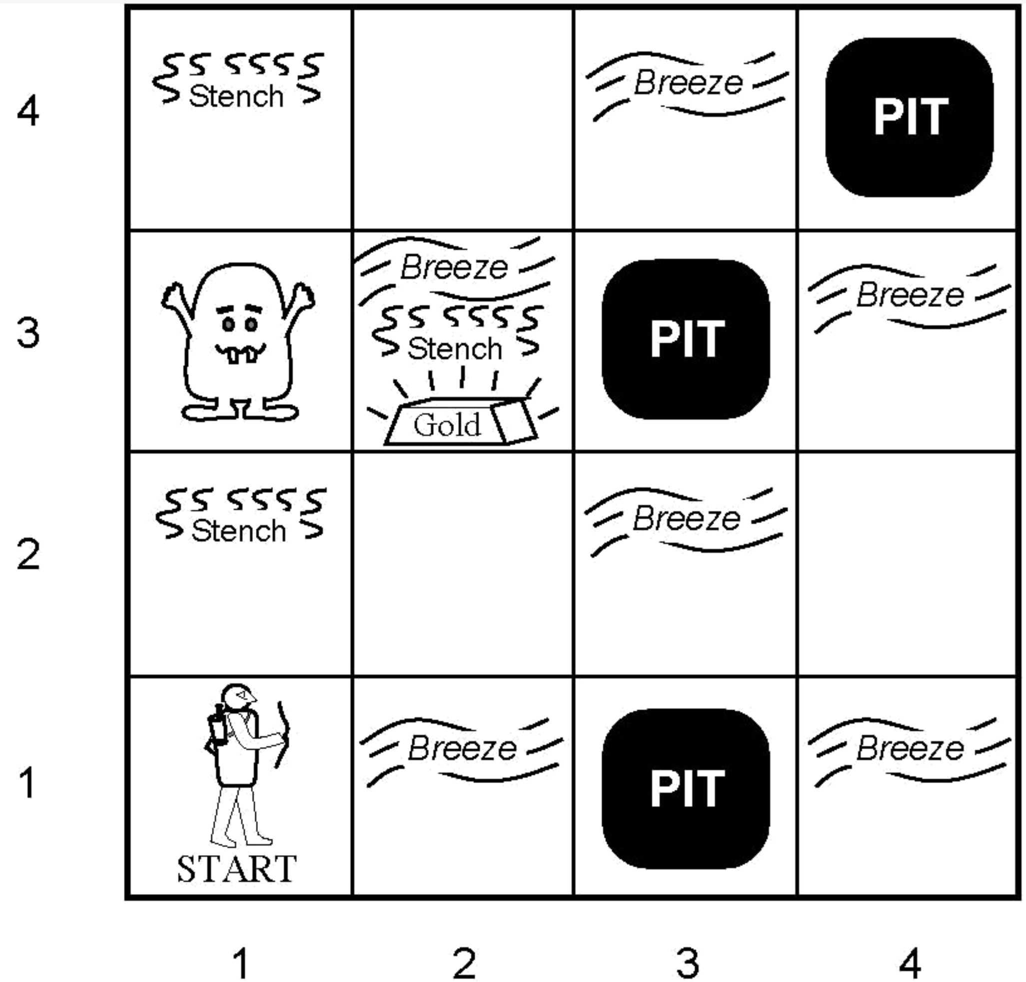
✓陷阱及相邻位置有微风

✓金子所在的位置有闪光

✓杀死妖兽会有惨叫声

✓碰到墙上会有撞击声

以5个符号的列表形式将感知信息提供给智能体，例如(stench, breeze, none, none, none)。

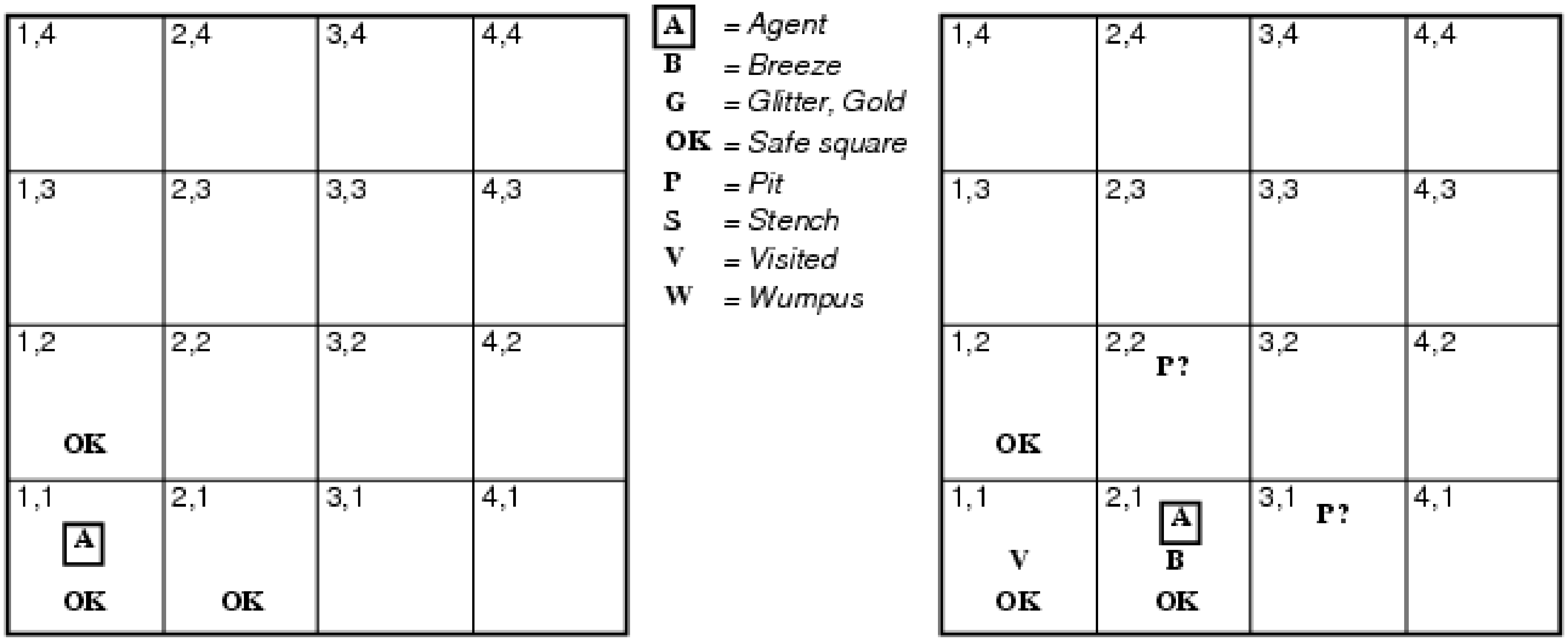


妖兽世界的环境类型

- 可观性??
- 确定性??
- 片段式的??
- 静态的??
- 离散的??
- 单主体的??

- No，只有局部感知
- Yes，状态完全由智能体动作决定
- No，前后动作是有关联的
- Yes，妖兽和陷阱都不会移动
- Yes
- Yes

探索妖兽世界



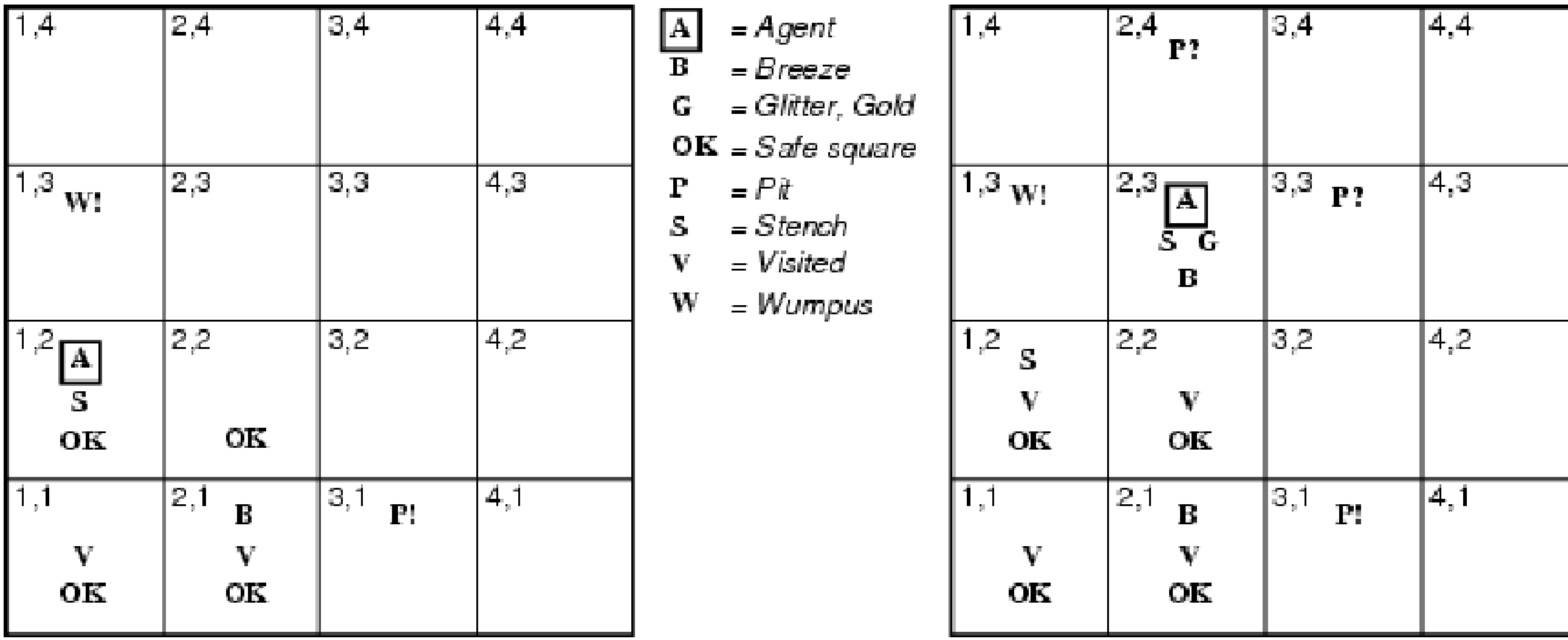
(a) (b)

初始KB包含环境规则，位于[1,1]位置，

感知为[none, none, none, none, none]

在[2,1]检测到微风(B)，表示相邻的方格 [2,2] 和 [3,1]中必定存在陷阱(P)，因此返回安全位置[1,1]

探索妖兽世界（续）

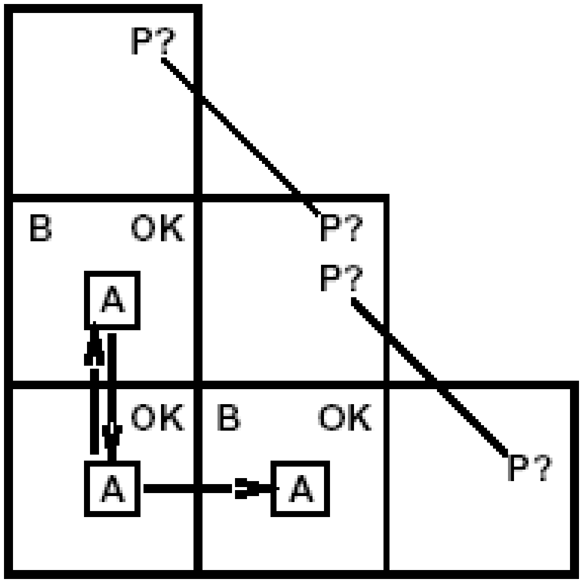


(a) (b)

[1,2]位置的感知为[*stench, none, none, none, none*]，则相邻的方格 [2,2] 和 [1,3]中必定存在妖兽(W)。

W肯定不在[2,2]，则W在[1,3]，[2,2]是安全的，因为[1,2]没有检测到微风(B)

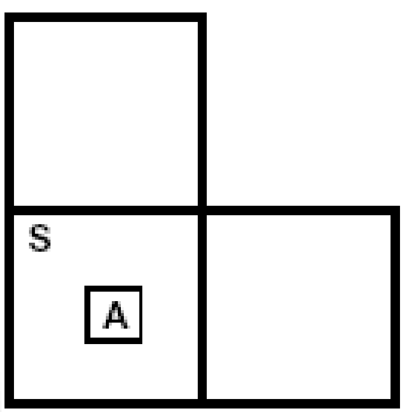
其它情况



[1,2] 和 [2,1] 中都有微风
⇒ 没有安全的行动

假设陷阱独立随机均匀分布，概率为0.2

则[2,2]有陷阱的概率为0.86， [3,1]和[1,3]有陷阱的概率均为0.31（不确定推理）



[1,1]中有臭气 ⇒ 不能移动

可以采取**强制**手段：向前方射箭

Wumpus在前方 ⇒ 死了 ⇒ 安全

Wumpus不在前方 ⇒ 安全

基于知识的智能体的功能

- 基于知识的智能体必须能够：
  - 表达状态，动作等
  - 合并新的感知
  - 更新内部对世界的表达
  - 演绎世界的隐藏特性
  - 演绎合适的动作
- 智能体根据可用的信息得出结论，如果可用信息正确，那么结论保证是正确的。——**逻辑推理**

逻辑是用来表达信息的形式语言

- **语法**：定义语言中的语句，怎样的表达是合法的
- **语义**：定义语句的“意思”

在逻辑中，语义定义了每条语句关于每种可能世界的**真值**，**非真即假**

以算术语言为例

$x + 2 \geq y$ 是一条语句；$x2 + y >$ 不是一条语句

$x + 2 \geq y$ 当 $x = 7$，$y = 1$ 时**为真**

$x + 2 \geq y$ 当 $x = 0$，$y = 6$ 时**为假**

蕴涵 entailment

⇔ **当且仅当**

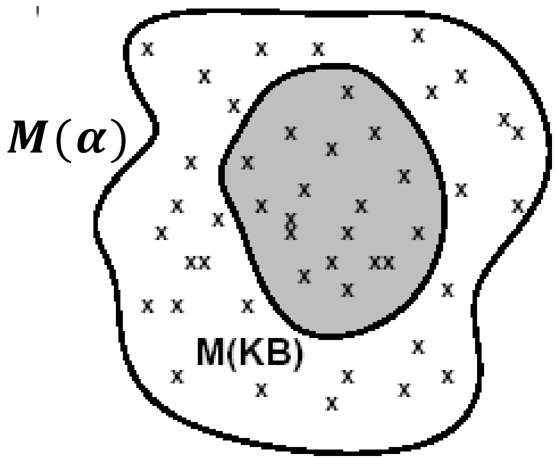
- **蕴涵**表示**一件事为真可得出另一件事也为真**：
  - α|=β（语句α蕴涵语句β）⇔ 使α为真的每个**模型**中，β也为真
  - KB |= α （知识库KB蕴涵语句α) ⇔ 在KB为真的所有模型中α也为真
- 如：
  - KB中包含语句“罗姆尼输了”和“奥巴马是美国总统”，则KB蕴涵语句“要么罗姆尼输了，要么奥巴马是美国总统”
  - $x + y = 4$ 蕴涵 $4 = x + y$
  - $x=0$ 蕴涵 $xy=0$
- 蕴涵是语句之间的关系，它是基于语义的

模型

- **模型：对“可能世界”的数学抽象。**
  - **m是α的一个模型 表示 语句α在模型m中为真，也称m满足α**
  - **用M(α)表示满足α的所有模型**

我们在模型的框架下考虑问题
模型是一个形式化构造的世界，是数学抽象，只关注语句的真假

$$\alpha \models \beta \Leftrightarrow \boldsymbol{M}(\boldsymbol{\alpha}) \subseteq \boldsymbol{M}(\boldsymbol{\beta})$$



因此有

$$KB \models \alpha \Leftrightarrow M(KB) \subseteq M(\alpha)$$

例如： KB = “罗姆尼输了” 和 “奥巴马是美国总统”
α = “罗姆尼输了”

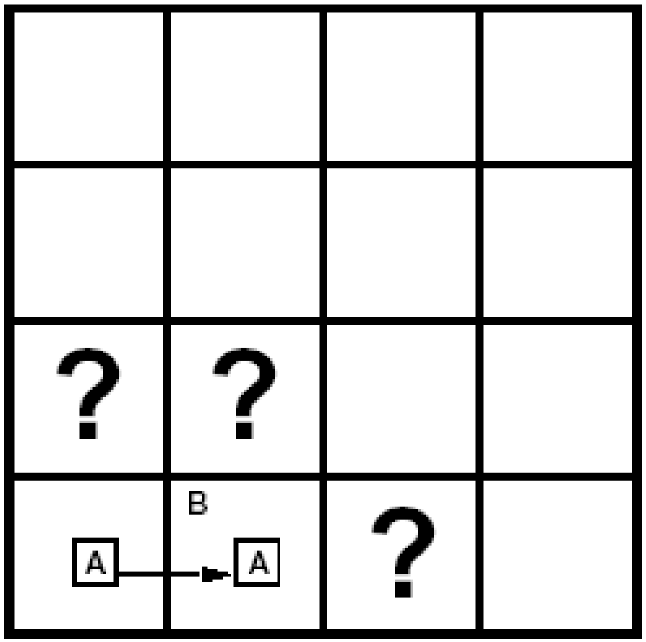
$$\boldsymbol{KB \models \alpha}$$

妖兽世界中的蕴涵

考虑下图中的情况，[1, 1]中什么都没有，在[2, 1]中有微风

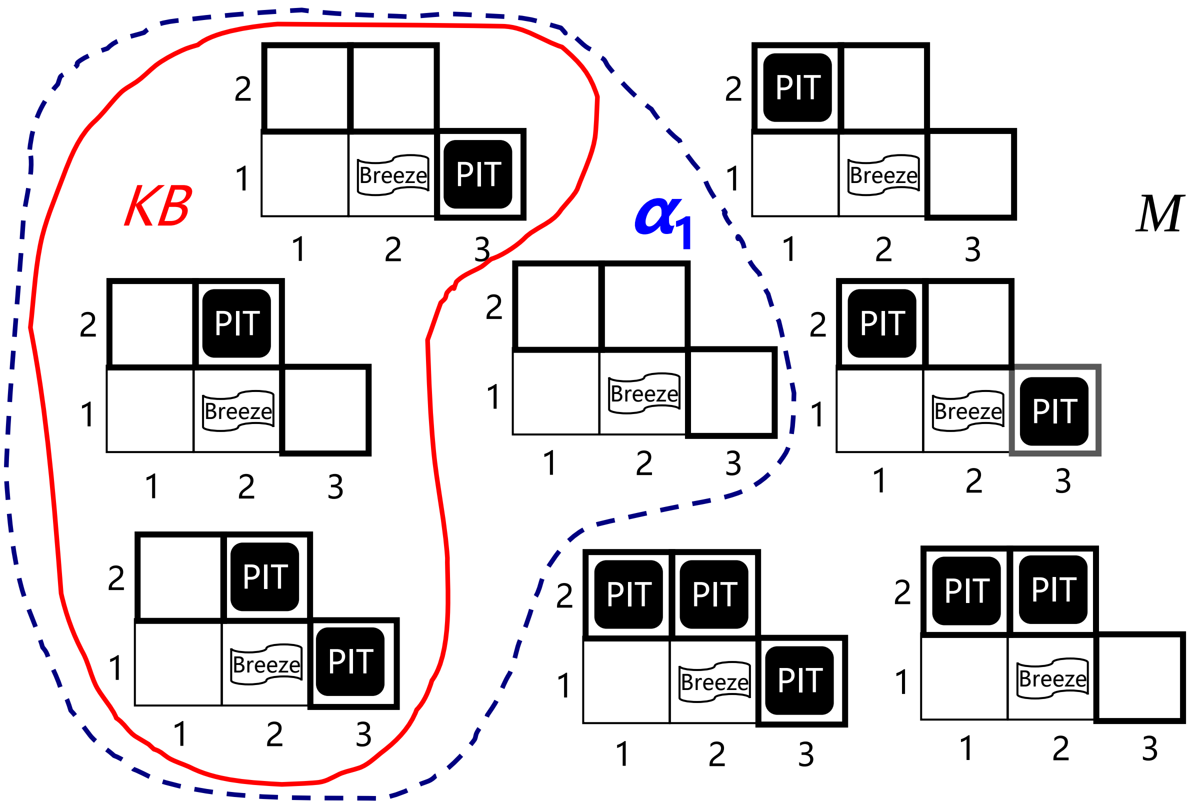
只考虑标有“?”方格中是否有陷阱的可能模型

3个布尔选择 ⇒ 8种可能的模型



可能的模型

α1：[1, 2]中没有陷阱

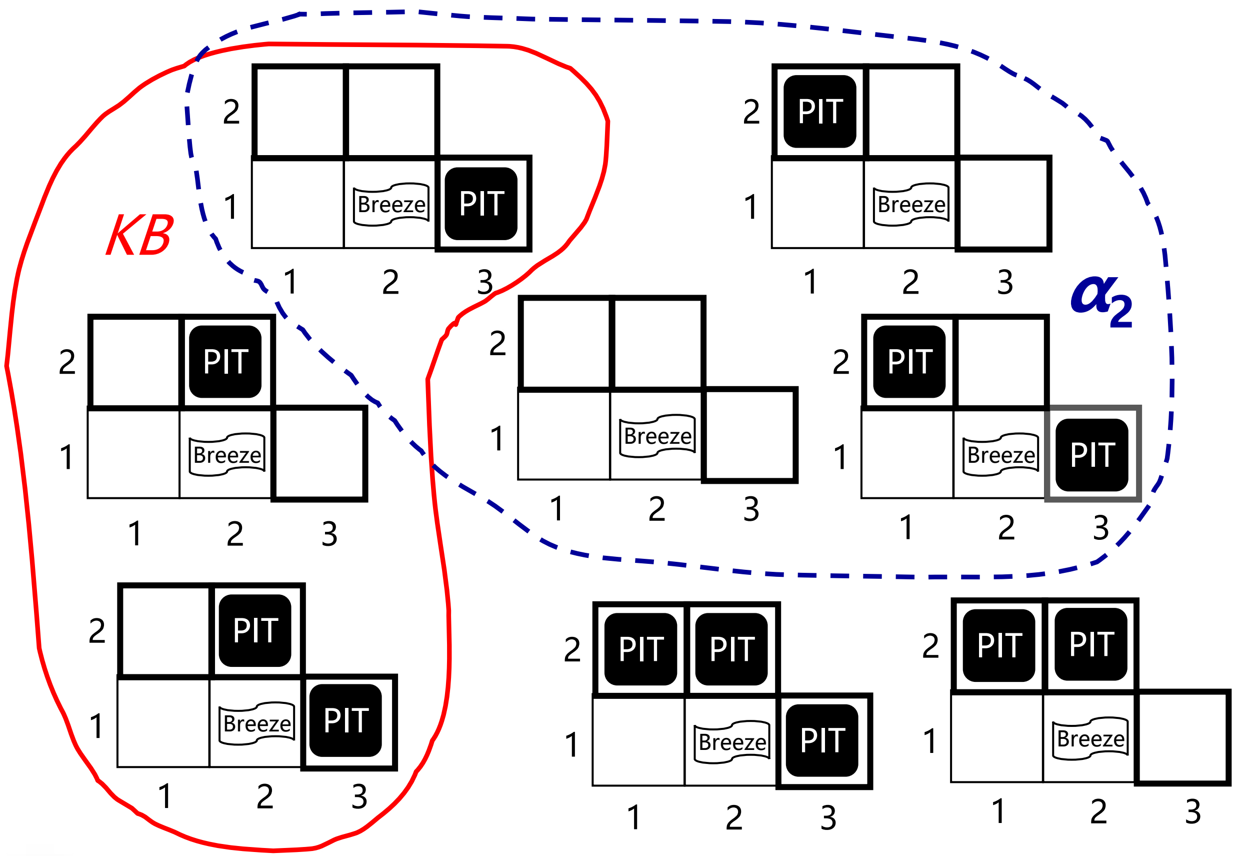


$$M(KB) \subseteq M(\alpha_1)?$$

结论：
$KB \models \alpha_1$

妖兽世界模型（续）

α_2：[2, 2]中没有陷阱



$$M(KB) \subseteq M(\alpha_2)?$$

结论：

KB$|\neq\alpha_2$

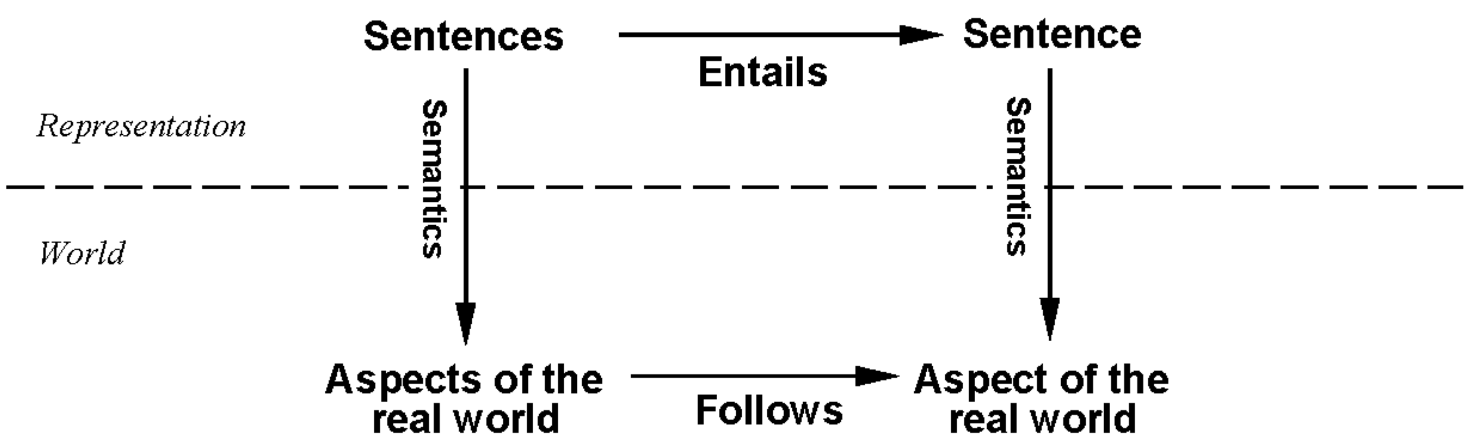
推理和蕴涵/蕴含

- **逻辑推理**：用蕴含推导出结论
- **模型检验**：通过枚举所有可能的模型来检验KB为真的情况下α都为真，即M(KB)含于M（α)
- **可靠的**：只导出蕴含句的推理算法被称为可靠的
- **完备性**：如果推理算法可以生成任一蕴含句，则完备。（在有限的情况下确实完备，但是常常不完备）

- **比喻**：KB的所有推论集合视为一个大干草堆，而语句α则视为一根针
- 如果推理算法 i 可以从KB导出α，我们表示为：$KB \vdash_i \alpha$，读为“i从KB导出α”
- 只要$KB \vdash_i \alpha$，$KB \models \alpha$就为真，则i是**合理**的
- 只导出蕴涵句的推理算法称为**可靠的**或**真值保持的**推理
- 只要$KB \models \alpha$，$KB \vdash_i \alpha$ 就为真，则i是**完备**的
  - 推理算法是**完备**的，如果它可以生成任一蕴涵句

知识表示和世界之间的关系

- **前提为真的任何世界中可以保证结论为真**
- **如果语句**是智能体的物理结构
- **推理**是从旧结构中创立新结构的过程
- **逻辑推理**应该确保新结构所代表的那部分世界的确是旧结构所代表的那部分的必然结论



落地（筑基，Grounding）

逻辑推理过程和智能体生存的正式环境之间的联系

- Q：如何知道KB在现实世界中为真?
- (simple) Answer：智能体的传感器创造了这一联系
  - 例如：妖兽世界的智能体有一个嗅觉传感器，只要有气味，智能体就创建一个合适的语句来表述
  - 只要该语句存在于知识库，它在现实世界中就为真
  - 感知语句：真值是通过产生它们的感知和语句构造过程来定义的
  - 一般规则：根据感知经验学习得到，但不是经验的直接陈述，由**学习**的语句构造过程产生

6.4 命题逻辑：一种简单逻辑——简单，但很强大

- 语法
- 语义
- 蕴涵

陈述句 (Declaration Sentences)

陈述句 (declarative sentences) 或者叫**命题** (propositions) 是一个可以判断对错的句子, 例如：

- p ：3 加 5 等于 8
- q ：铁可以导电

我们使用符号 $p, q, r, \ldots$ 来表示这种原子 (atomic) 的句子, 并使用一些方法将它们组成更复杂的句子, 例如：

- $\neg$ ：如 $\neg p$ 是 "3 加 5 **不**等于 8" .
- $\vee$ ：如 $p \vee q$ 是 "3 加 5 等于 8 **或**铁可以导电" .
- $\wedge$ ：如 $p \wedge q$ 是 "3 加 5 等于 8 **且**铁可以导电" .
- $\rightarrow$ ：如 $p \rightarrow q$ 是 "**如果** 3 加 5 等于 8 **则**铁可以导电" .

在下边, 我们暂时先不去关心 p 和 q 本身的含义, 而仅去关注这些符号所组成的逻辑结构.

6.4.1 命题逻辑：语法

- **原子语句**
  - **命题词**：不可分割的句法元素。
    - 每个命题词代表一个或为真或为假的命题。
    - 用大写字母表示，可能包括其他符号或下标。
      - 如P, Q, R, $W_{1,3}$, North
      - 可以任意命名，但通常选择有实际意义的名称，

      如用$W_{1,3}$表示“Wumpus位于[1, 3]”
  - 有两个命题词有固定的含义：
    - True 是永真命题
    - False是永假命题

6.4.1 命题逻辑：语法

- **复合句**：由**原子语句**用**逻辑连接符**构造而成。
  - ¬ **(非)** ：如$\neg W_{1,3}$这样的语句称为 $W_{1,3}$ 的**否定式**
    - 文字指原子语句（正文字）或否定的原子语句（ 负文字）
  - ∧ **(与)** ：主要连接词为∧的语句，如$W_{1,3} \wedge P_{1,3}$，称为“**合取式**”，其各部分称为“合取子式”
  - ∨ **(或)** ：采用连接词为∨的语句，如$W_{1,3} \vee P_{1,3}$，称为“**析取式**”，其各部分称为“析取子式”
  - ⇒ **(蕴含)** ：采用连接词为⇒ (有的书中用⊃或→)的语句，称为“**蕴含式**”或“条件式”，也称为“规则”或“if-then语句”。
    - $(W_{1,3} \wedge P_{1,3}) \Rightarrow \neg W_{2,2}$
    - **前提/前项**：$W_{1,3} \wedge P_{1,3}$
    - **结论/后项**： $\neg W_{2,2}$
  - ⇔ **(当且仅当)** ：语句$W_{1,3} \Leftrightarrow \neg W_{2,2}$**是双向蕴含式**，有些书记为≡

命题逻辑语句的BNF（巴克斯-瑙鲁范式）语法，及逻辑运算的优先级，从高到低

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{True} \mid \mathit{False} \mid P \mid Q \mid R \mid \ldots \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
 & \mid & \neg\ \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \land \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \lor \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
 & \mid & \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
 & & \\
\text{OPERATOR PRECEDENCE} & : & \neg, \land, \lor, \Rightarrow, \Leftrightarrow
\end{array}
$$

- 命题逻辑的优先级次序为（高到低）： $\neg$， $\land$， $\lor$， $\Rightarrow$， $\Leftrightarrow$
- **$\neg P \lor Q \land R \Rightarrow S$**等价于（（$\neg P$）$\lor$（$Q \land R$））$\Rightarrow S$

6.4.2 命题逻辑：语义

语义定义了用于判断特定模型的语句真值的规则

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

P⇒Q：使得该语句为假的唯一条件是如果P为真而Q为假！

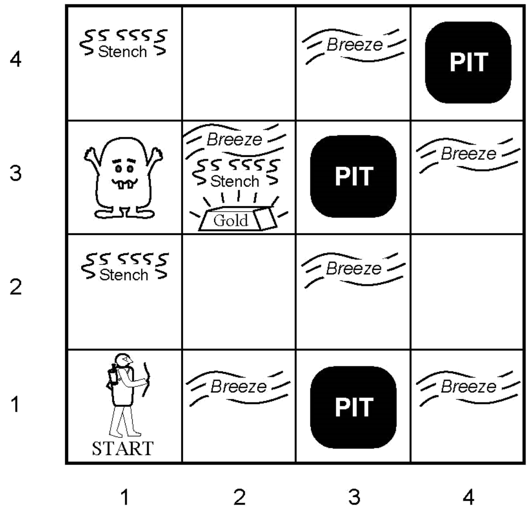
P⇔Q：只要P⇒Q与Q⇒P同时为真，即为真

1、命题逻辑不要求P和Q之间存在相关性或者因果关系。“5是奇数蕴含东京是日本的首都”是命题逻辑的真语句
2、前提为假的任何蕴含都为真；M(P) M(Q)，空集是任何集合的子集
3、如果P为真，我主张Q为真；否则无可奉告

6.4.3 知识库的例子：Wumpus世界中的陷阱

基于知识的智能体

- 核心构件是其**知识库（KB）**
  - 知识库是一个语句集合，语句由知识表示语言来表达，表示了关于世界的某些断言，包括问题的一些背景知识
- 将新语句添加到知识库以及查询目前所知内容的途径
  - **TELL和ASK**：可能涉及推理
  - 当ASK知识库一个问题时，答案必须遵循（follow）事先被告知的知识库的内容
- 基于知识的智能体也是用**感知信息**作为输入，并返回一个**行动**



对于每个i，j：

- 如果[i, j]中有陷阱，令$P_{i,j}$为真。
- 如果[i, j]中有微风，令$B_{i,j}$为真。
- 如果[i,j]有怪兽，则$W_{i,j}$为真
- 如果[i,j]有臭气，则$S_{x,y}$为真

知识库中包含

- [1, 1]中没有陷阱，
  - **R_1**：　$\neg P_{1,1}$
- 在与陷阱直接相邻的方格内，智能体可以感知到微风（领域知识信息）
  - **R_2**:　$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - **R_3**:　$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- 感知信息：
  - **R_4**:　$\neg B_{1,1}$
  - **R_5**:　$B_{2,1}$
- 知识库由R_1到R_5的语句构成。
  - $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

6.4.4 简单推理过程：通过模型检验来判断蕴涵（枚举法）

推理：判断对于某些语句α，KB|=α是否成立

知识库推理的真值表

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	KB	α_1
false	*false*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*false*	*false*	*false*	*false*	*false*	*true*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*false*	*true*
false	*true*	*false*	*false*	*false*	*false*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*false*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*false*	*true*	*true*	<u>*true*</u>	<u>*true*</u>
false	*true*	*false*	*false*	*true*	*false*	*false*	*false*	*true*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*false*

共有**2^7=128个**可能模型，仅有**3个模型中KB为真。**

在这3个模型中，$P_{1,2}$为假，说明[1,2]中没有陷阱，
[2,2]中有陷阱吗？

命题逻辑：推理

推理：判断对于某些语句α，KB|=α是否成立

```
function TT-ENTAILS?(KB, α) returns true or false
    symbols ← a list of the proposition symbols in KB and α
    return TT-CHECK-ALL(KB, α, symbols, [])

function TT-CHECK-ALL(KB, α, symbols, model) returns true or false
    if EMPTY?(symbols) then
        if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
        else return true
    else do
        P ← FIRST(symbols); rest ← REST(symbols)
        return TT-CHECK-ALL(KB, α, rest, EXTEND(P, true, model)) and
                   TT-CHECK-ALL(KB, α, rest, EXTEND(P, false, model))
```

判断命题蕴涵的真值表枚举算法

TT-ENTAILS? 执行对变量赋值有限空间的**递归枚举**，算法直接实现了蕴涵的定义，是**可靠的、完备的**，因为它可以用于任意KB和α，而且总能够终止（因为只存在**有限多个**需要检验的模型）。

（TT代表真值表）如果语句包含在模型中，PL-TRUE?返回真。变量MODEL表示一个不完全模型（部分模型），只对某些变量赋值。函数调用EXTEND返回一个新的不完全模型，在该模型中P为真。

- 真值表枚举：时间复杂度为$O(2^n)$, 空间复杂度$O(n)$

6.5 命题逻辑定理证明：通过定理证明来判断蕴涵

- 在知识库语句上直接应用推理规则以构建目标语句的证明，而无须关注模型
- 适用于模型数目庞大而证明很短

与蕴涵相关的附加概念：逻辑等价

如果两个语句在同样的模型集合中为真，则二者逻辑等价。

定义：任意两个语句α和β是等价的当且仅当它们互相蕴涵时。

$$\alpha \equiv \beta \ \textbf{\textit{iff}} \ \alpha \models \beta \ \textbf{\textit{and}} \ \beta \models \alpha$$

标准的逻辑等价

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$	commutativity of $\wedge$	可交换性
$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$	commutativity of $\vee$	
$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$	结合律
$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$	
$\neg(\neg\alpha) \equiv \alpha$	double-negation elimination	双重否定消去
$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition	逆否命题
$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta)$	implication elimination	蕴涵消去
$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination	双向蕴涵消去
$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$	de Morgan	摩根律
$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$	de Morgan	
$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$	∧对∨的分配率
$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$	∨对∧的分配率

与蕴涵相关的附加概念：有效性（合法性）和可满足性

- **有效性：**定义：一个语句是有效的，如果在**所有的模型**中它都为真。
  - 有效语句也叫重言式——必定为真：每个有效语句都**逻辑等价**于True。
  - 演绎定理：对于任意语句α和β，α|=β 当且仅当 语句（α⇒β）是有效的。
  - 每个有效的（合法的）蕴涵语句都描述了一个有效的（合法的）推理。
- **可满足性：**定义：一个语句是可满足的，如果对于**部分模型**它是真值。
  - 一个句子是不可满足的如果它在任何模型中都不为真。
- **有效性与可满足性的关联**
  - α是有效的 当且仅当 ¬α不可满足【¬α不可满足，即在任何模型中¬α都不为真，即在任何模型中α都为真，即α是有效的】
  - α是可满足的当且仅当¬α不是有效的【¬α不是有效的，即不是在所有模型中¬α都是真的，即在某些模型中¬α是假的，即在某些模型中α是真的，即α是可满足的】
  - α|=β 当且仅当语句 (α∧¬β)是不可满足的。
    - 通过归谬来证明（反证法、矛盾法）：假定语句β为假，将推导出和已知公理α的矛盾。该矛盾正好说明语句(α∧¬β)是不可满足的。

6.5.1 推导和证明

应用**推理规则**得到一个证明：一系列结论直到目标语句。

1）**假言推理规则**（Modus Ponens，拉丁文）

$$\frac{\alpha \Rightarrow \beta,\ \ \alpha}{\beta}$$

只要给定任何形式为$\alpha \Rightarrow \beta$和α的语句，就可以推导出语句β

例： 已知 (WumpusAhead∧WumpusAlive)⇒Shoot

和 (WumpusAhead ∧ WumpusAlive)

就可以推导出 Shoot

2）**消去合取词**

$$\frac{\alpha \wedge \beta}{\alpha}$$

可以从合取式推导出任何合取子式。

例：(WumpusAhead ∧ WumpusAlive) 可以推导出WumpusAlive

如图所有逻辑等价都可以作为推理规则

$$
\begin{aligned}
(\alpha \land \beta) &\equiv (\beta \land \alpha) && \text{commutativity of } \land \\
(\alpha \lor \beta) &\equiv (\beta \lor \alpha) && \text{commutativity of } \lor \\
((\alpha \land \beta) \land \gamma) &\equiv (\alpha \land (\beta \land \gamma)) && \text{associativity of } \land \\
((\alpha \lor \beta) \lor \gamma) &\equiv (\alpha \lor (\beta \lor \gamma)) && \text{associativity of } \lor \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \lor \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \land \beta) &\equiv (\neg\alpha \lor \neg\beta) && \text{de Morgan} \\
\neg(\alpha \lor \beta) &\equiv (\neg\alpha \land \neg\beta) && \text{de Morgan} \\
(\alpha \land (\beta \lor \gamma)) &\equiv ((\alpha \land \beta) \lor (\alpha \land \gamma)) && \text{distributivity of } \land \text{ over } \lor \\
(\alpha \lor (\beta \land \gamma)) &\equiv ((\alpha \lor \beta) \land (\alpha \lor \gamma)) && \text{distributivity of } \lor \text{ over } \land
\end{aligned}
$$

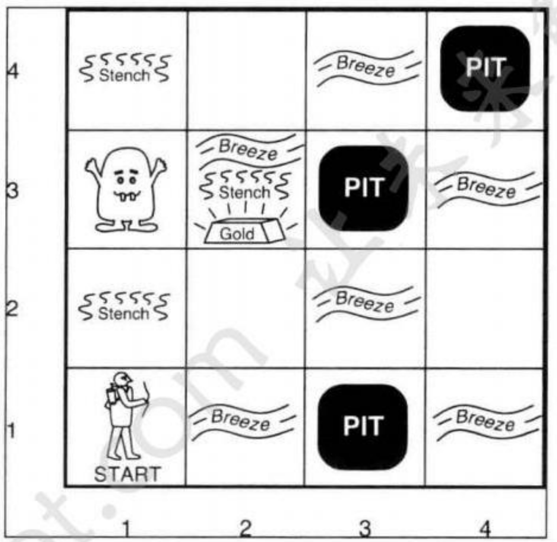
$$\frac{\alpha \Leftrightarrow \beta}{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}$$

$$\frac{(\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha)}{\alpha \Leftrightarrow \beta}$$

推理规则和等价如何应用于Wumpus世界

知识库中包含

- [1, 1]中没有陷阱，
  - R_1:　$\neg P_{1,1}$
- 在与陷阱直接相邻的方格内，智能体可以感知到微风（领域知识信息）
  - R_2:　$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - R_3:　$B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- 感知信息：
  - R_4:　$\neg B_{1,1}$
  - R_5:　$B_{2,1}$
- 知识库由R_1到R_5的语句构成。
  - $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$



如何证明$\neg P_{1,2}$即[1,2]中没有陷阱?

1. 将**双向蕴含消去**应用于R_2，得到R_6：　$(B_{1,1} \Rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
2. 对R_6**消去合取词**，得到R_7：　$((P_{1,2} \vee P_{2,1}) \Rightarrow B_{1,1})$
3. R_7**逆否命题**的逻辑等价，得到R_8：　$(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \vee P_{2,1}))$
4. 对R_8和R_4运用**假言推理规则**，得到R_9：　$\neg(P_{1,2} \vee P_{2,1})$
5. 应用**De Morgan律**，得出结论 R_{10}：　$\neg P_{1,2} \wedge \neg P_{2,1}$

　　也就是[1,2]和[2,1]都不包含陷阱。

通过搜索算法来找出证明序列

（这叫搜索证明，是模型枚举的一个替代方法）

1. **初始状态：** 初始知识库KB
2. **行动：** 行动集合由应用于语句的所有推理规则组成，要匹配推理规则的上半部分。
3. **结果：** 将推理规则下半部分的语句实例加入知识库KB
4. **目标：** 要证明的语句状态

- 推理规则的应用
  - （**正向推理**）从初始知识库**正向**出发，应用推理规则以生成目标语句，或（**反向推理**）从目标语句**反向**出发，试图找到由初始知识库引出的推理规则链
    - 通常需要将句子转换成范式
  - 寻找证明的过程与搜索问题中寻找解的过程非常类似：定义后继函数以便生成推理规则所有可能的应用
  - 命题逻辑的推理是NP-Hard的

逻辑系统的单调性

"

逻辑系统的最后一个概念是**单调性**，单调性意味着逻辑蕴涵语句集会随着添加到知识库的信息增长而增长[1]。对于任意语句α和β,

如果 $KB \models \alpha$，那么 $KB \wedge \beta \models \alpha$

例如，假设知识库包含附加断言β，β 宣称世界中正好有 8 个无底洞。这条知识可能有助于 Agent 推导出附加结论，但是它无法推翻任意已经推导出的结论α——如[1，2]中没有无底洞的结论。单调性意味着只要在知识库中发现了合适的前提，就可以应用推理规则——规则的结论"与知识库中的其余内容无关"。

"

可靠：假言推理和消去合取词。
完备？

1. 将**双向蕴含消去**应用于R_2，得到R_6：　$(B_{1,1} \Rightarrow (P_{1,2} \lor P_{2,1})) \land ((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$
2. 对R_6**消去合取词**，得到R_7：　$((P_{1,2} \lor P_{2,1}) \Rightarrow B_{1,1})$
3. R_7**逆否命题**的逻辑等价，得到R_8：　$(\neg B_{1,1} \Rightarrow \neg(P_{1,2} \lor P_{2,1}))$
4. 对R_8和R_4运用**假言推理规则**，得到R_9：　$\neg(P_{1,2} \lor P_{2,1})$
5. 应用**De Morgan律**，得出结论 R_{10}：　$\neg P_{1,2} \land \neg P_{2,1}$

   也就是[1,2]和[2,1]都不包含陷阱。

取消 2中的消去合取词规则？

如果推理规则不够充分，那么目标将不可达，即只用那些推理规则找不到证明

6.5.2 归结证明

归结（resolution）规则

当它和任何一个完备的搜索算法相结合时，可以得到完备的推理算法。

单元归结（unit resolution）推理规则

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

l_i和m是互补文字

互补文字：
一个文字是另一个文字的否定式。

- **单元归结：**选取一个子句（文字的**析取式**）和一个文字，生成一个新的子句，该新子句包含除了两个互补文字以外的原始子句中的所有文字。

例1：
$$\frac{P_{1,3} \vee P_{2,2}, \quad \neg P_{2,2}}{P_{1,3}}$$

全归结（full resolution）推理规则

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

l_i和m_j是互补文字

- **全归结：**选取两个子句，生成一个新的子句，该新子句包含除了两个互补文字以外的原始子句中的所有文字。

例2：

$$\frac{P_{1,1} \vee P_{3,1}, \quad \neg P_{1,1} \vee \neg P_{2,2}}{P_{3,1} \vee \neg P_{2,2}}$$

单元归结（unit resolution）推理规则

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k}$$

l_i和m是互补文字

互补文字：
一个文字是另一个文字的否定式。

全归结（full resolution）推理规则

$$\frac{l_1 \vee \cdots \vee l_k, \qquad m_1 \vee \cdots \vee m_n}{l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

l_i和m_j是互补文字

合取规则（消解规则）

$p \vee q$ $\underline{\neg p \vee r}$ $\therefore q \vee r$	$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$	消解

消解规则的证明

命题（消解规则） $(p \vee q) \wedge (\sim p \vee r) \Rightarrow q \vee r$

即，已知$\{p \vee q, \sim p \vee r\}$ 证明:$q \vee r$

证明一 反证法. 假设结论$q \vee r$不成立，则$\sim(q \vee r)$成立，即$\sim q \wedge \sim r$成立，即$\sim q, \sim r$成立.

由$p \vee q$与$\sim q$, 可知p成立，那么$\sim p$不成立.

由$\sim p \vee r$ 与$\sim r$, 可知$\sim p$成立, 推出矛盾.

故$q \vee r$成立. ■

$p \vee q$ $\neg p \vee r$ $\therefore q \vee r$	$[(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$	消解

证明二 $(p \vee q) \wedge (\sim p \vee r)$

$$\Leftrightarrow (p \wedge \sim p) \vee (p \wedge r) \vee (q \wedge \sim p) \vee (q \wedge r)$$
$$\Leftrightarrow (p \wedge r) \vee (q \wedge \sim p) \vee (q \wedge r)$$

（以下$A \vee B$简写为AB）

$$\Leftrightarrow pqq \wedge pqr \wedge p{\sim}pq \wedge p{\sim}pr \quad // \text{ 分配律}$$
$$\wedge rqq \wedge rqr \wedge r{\sim}pq \wedge r{\sim}pr$$
$$\Leftrightarrow r \vee q \vee q \quad // \text{ 化简律}$$
$$\Leftrightarrow q \vee r \quad \blacksquare$$

结果子句中每个文字只能出现1次，去除文字的多余副本叫做“归并”（factoring），如A∨A可简化为A，即为**化简律。**

合取范式（CNF）

上节的归结规则只应用于子句（文字的析取式），如何对于所有的命题逻辑，实现完备推理过程?

合取范式（CNF)

合取范式（CNF)：以子句的合取式表达的语句

命题逻辑的每个语句逻辑上等价于**文字析取式**的**合取式**（conjunction of disjunctions of literals)

例子：$(A \lor \neg B) \land (B \lor \neg C \lor \neg D)$

CNF语句可作为归结过程的输入

怎样将语句转换为CNF?

1、消除等价词 $\Leftrightarrow$

2、消去蕴涵词 $\Rightarrow$

3、否定词 $\neg$ 内移

4、消去嵌套（在可能的位置上将$\lor$对$\land$进行分配)

实例：合取范式

R_2：$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

1. 消去 $\Leftrightarrow$, 用 $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$取代$\alpha \Leftrightarrow \beta$

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

2. 消去 $\Rightarrow$, 用 $\neg\alpha \vee \beta$取代$\alpha \Rightarrow \beta$

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

3. 用摩根律和双重否定将 $\neg$ 移到内部

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

4. 应用分配律($\wedge$ 或 $\vee$)

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

51

6.5.2 归结证明

基于归结的推理过程使用的是反证法

归结法（也称消解法）的本质是一种反证法。

为了证明一个命题A恒真，要证明其反命题~A恒假。所谓恒假就是不存在模型，即在所有的可能解释中，~A均取假值。但一命题的解释通常有无穷多种，不可能一一测试。为此，Herbrand建议使用一种方法：从众多的解释中，选择一种代表性的解释，并严格证明：任何命题，一旦证明为在这种解释中取假值，即在所有的解释中取假值，这就是**Herbrand解释**。

这里用~表示¬

要证明：$A_1 \wedge A_2 \wedge A_3 \rightarrow B$ 是定理（重言式）

等价于

$\Leftrightarrow A_1 \wedge A_2 \wedge A_3 \wedge \sim B$ 是矛盾(永假)式

归结推理方法就是从$A_1 \wedge A_2 \wedge A_3 \wedge \sim B$ 出发，使用归结推理规则来寻找矛盾，最后证明定理成立。归结法（消解法）的本质是数学中的**反证法**，称为“**反演推理方法**”。

归结算法

- KB操作归结为可满足性，KB|=α当且仅当(KB∧¬α)是不可满足的
- 为了证明KB|=α，只需证明KB∧¬α不可能为真

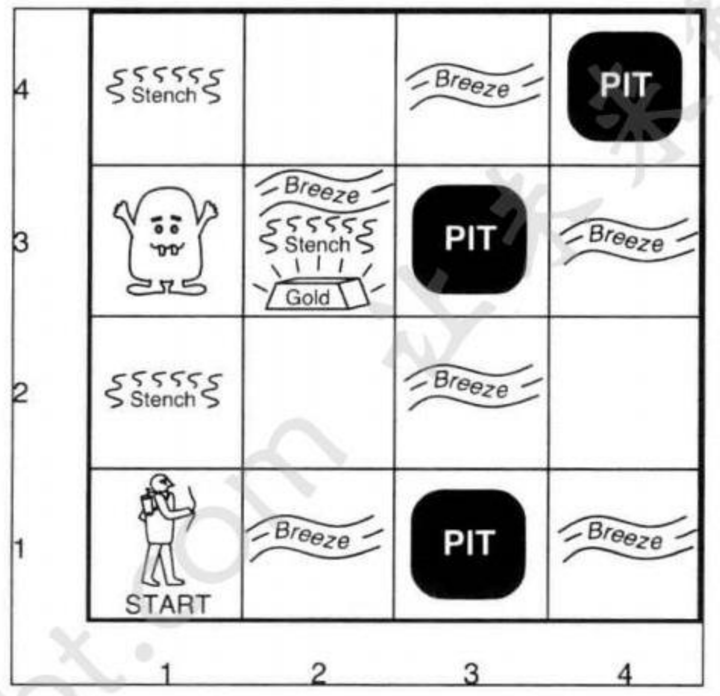
归结实例：wumpus世界中的陷阱

知识库中包含

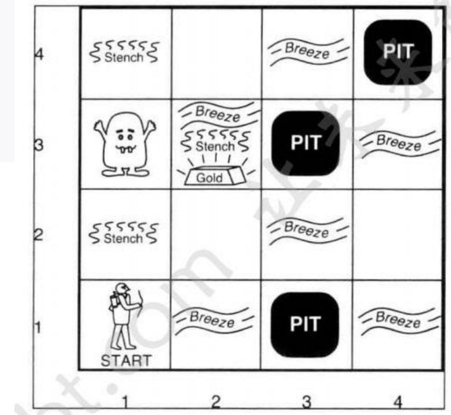
- [1, 1]中没有陷阱，
  - **R_1**: $\neg P_{1,1}$
- 在与陷阱直接相邻的方格内，智能体可以感知到微风（领域知识信息）
  - **R_2**: $B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$
  - **R_3**: $B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1})$
- 感知信息：
  - **R_4**: $\neg B_{1,1}$
  - **R_5**: $B_{2,1}$

- 知识库由R_1到R_5的语句构成。
  - $R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5$

32



归结实例



当Agent位于[1,1]时，那里没有微风，

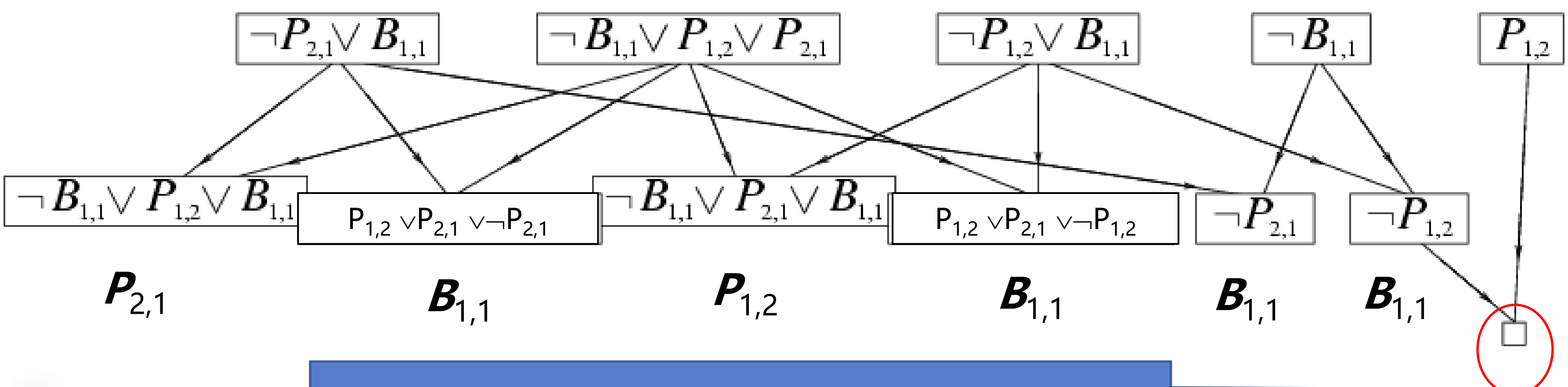
因此，相邻的方格中没有陷阱。相关的知识库为：

$KB = R_2 \wedge R_4 = (B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})) \wedge \neg B_{1,1}$

希望证明 α，即 $\neg P_{1,2}$

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

将(KB∧¬α)转换为CNF，得到的子句如下图顶部



在一种解释中取假值，即在所有的解释中取假值

归结算法

- KB操作归结为可满足性，KB|=α当且仅当(KB∧┐α)是不可满足的
- 为了证明KB|=α，只需证明KB∧¬α不可能为真
- PL-resolution是完备的（引进归结闭包）

Resolution algorithm

Proof by contradiction, i.e., show $KB \wedge \neg\alpha$ unsatisfiable

```
function PL-Resolution(KB, α) returns true or false
    clauses ← the set of clauses in the CNF representation of KB ∧ ¬α
    new ← { }
    loop do
        for each Ci, Cj in clauses do
            resolvents ← PL-Resolve(Ci, Cj)
            if resolvents contains the empty clause then return true
            new ← new ∪ resolvents
        if new ⊆ clauses then return false
        clauses ← clauses ∪ new
```

基于归结的推理
- 转换所有式子为CNF
- 重复使用归结规则
- 如果推导失败，返回不可满足性

PL-resolve返回对它的两个输入进行归结得到的所有可能的数据的集合

6.5.2 归结证明

$$\begin{array}{l} p \vee q \\ \underline{\neg p \vee r} \\ \therefore q \vee r \end{array} \qquad [(p \vee q) \wedge (\neg p \vee r)] \rightarrow (q \vee r)$$

- 任何完备的搜索算法，只使用归结规则，就可以生成命题逻辑中被任何知识库蕴涵的任何结论（反证法完备性）：
  - 归结总是可以用于确认或者反驳某个语句，但无法用于真值语句的枚举。

7.5.3 Horn子句和限定子句

在很多实际情况下无需用到归结的全部能力

现实世界的知识库通常满足它们所包含的语句形式的特定限制，这使得它们可以使用更受限也更有效的推理算法。

受限子句: 指恰好**只含有一个正文字**的析取式

$(\neg L_{1,1} \vee \neg Breeze \vee B_{1,1})$ √

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

霍恩子句-Horn Clause

- 定义：至多只有**一个正文字**的文字析取式，例如：$\neg P_{1,1} \vee \neg P_{2,2} \vee P_{3,1}$
- 所有限定子句都是Horn Clause
- 没有正文字的析取式也是Horn Clause，被称为 **目标子句**
- Horn子句在归结下是封闭的：如果对两个Horn子句进行归结，结果依然是Horn子句。

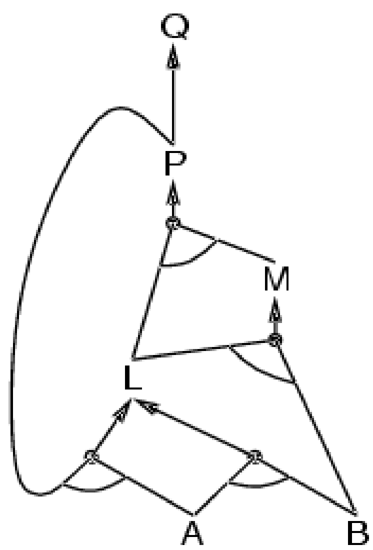
只包含限定子句的知识库的意义 (Horn Clause的重要性)

- 每个Horn Clause都可写成一个蕴涵式
  - 只有一个正文字的Horn Clause称为限定子句
  - $\neg P_{1,1} \vee \neg P_{2,2}$等价于$P_{1,1} \wedge P_{2,2} \Rightarrow false$，这样的语句称为数据库世界的完整性约束
  - 假定知识库只包含限定子句，而且没有完整性约束（Horn Form）
- 使用Horn Clause的推理可在前向和反向链接中进行
- 用Horn Clause判定蕴涵所需的时间与数据库大小成线性关系 （前向和反向）

6.5.4 前向链接

- 前向链接算法用于判定（查询）单个命题词是否被限定子句的知识库所蕴涵
- 前向链接：从**知识库已知事实**开始，如果蕴涵的所有前提已知，那么把它结论添加到已知事实集，直到无法进行更一步的推理或者目标被发现。 是**数据驱动的推理**
- 前向链接是**可靠的**：每个推理本质上是假言推理规则的一个应用
- **完备的**：每个被蕴涵的原子语句都可以生成

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



假言推理规则（分离规则）

$$\frac{\alpha \Rightarrow \beta, \ \alpha}{\beta}$$

消去合取词（与消去）

$$\frac{\alpha \wedge \beta}{\alpha}$$

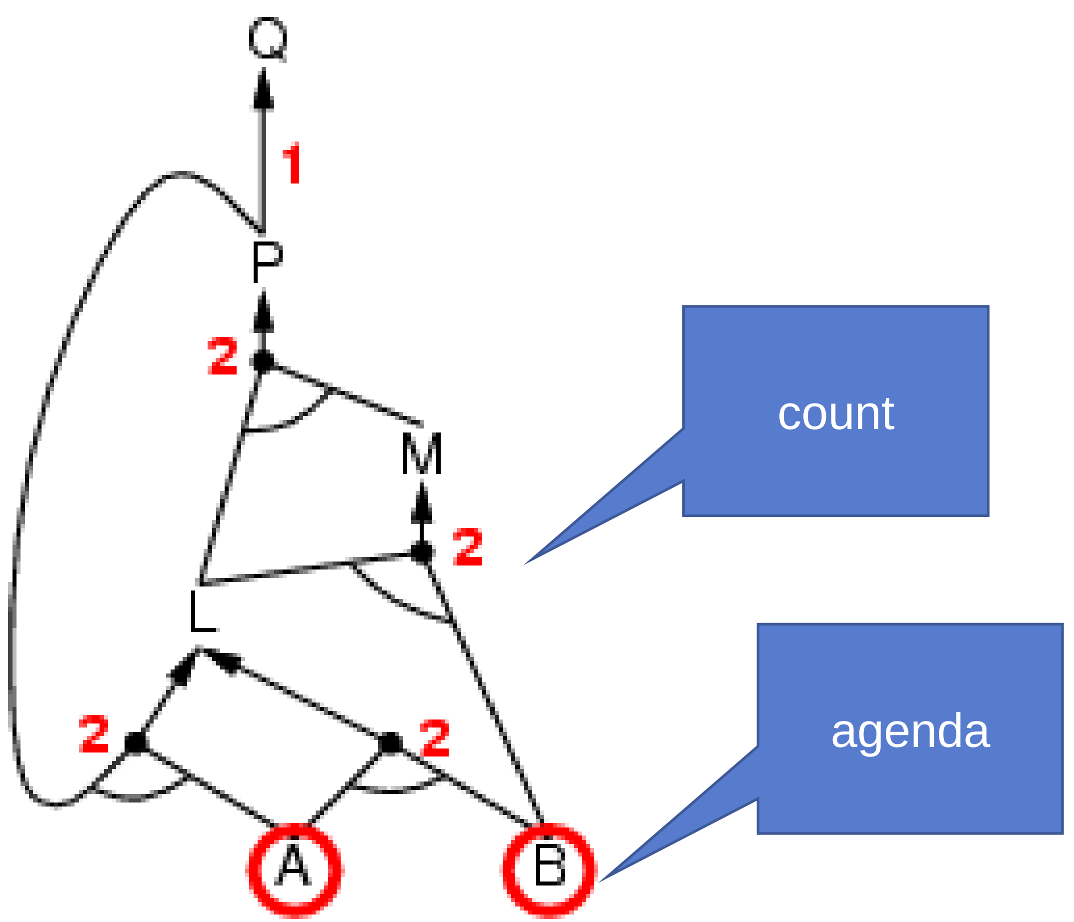
- 从**知识库已知事实**开始，如果蕴涵的所有前提已知，那么把它结论添加到已知事实集，直到无法进行更一步的推理或者目标被发现。

```
function PL-FC-ENTAILS?(KB, q) returns true or false
  inputs: KB, the knowledge base, a set of propositional definite clauses
          q, the query, a proposition symbol
  count ← a table, where count[c] is the number of symbols in c's premise
  inferred ← a table, where inferred[s] is initially false for all symbols
  agenda ← a queue of symbols, initially symbols known to be true in KB

  while agenda is not empty do
     p ← POP(agenda)
     if p = q then return true
     if inferred[p] = false then
        inferred[p] ← true
        for clause c in KB where p is in c.PREMISE do
           decrement count[c]
           if count[c] = 0 then add c.CONCLUSION to agenda
  return false
```

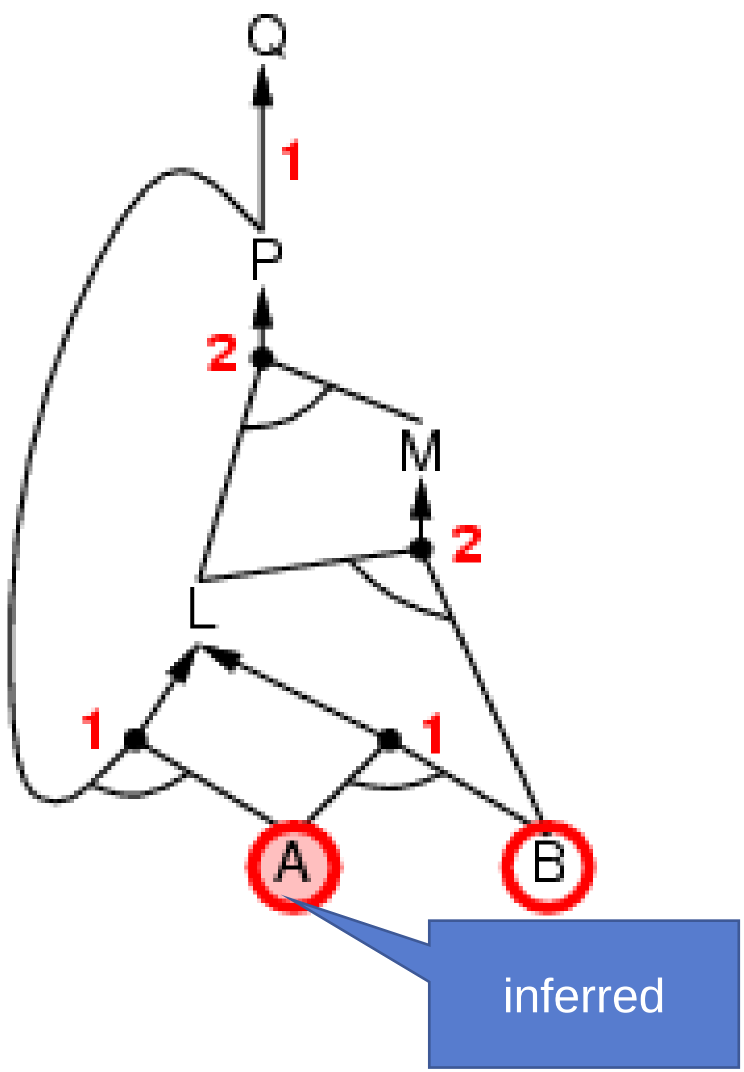
前向链接

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



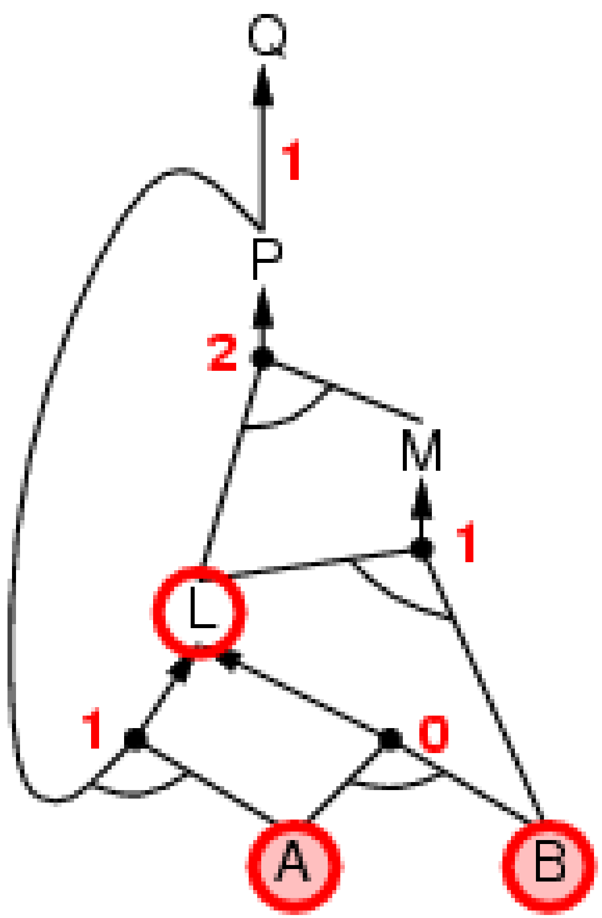
前向链接

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$



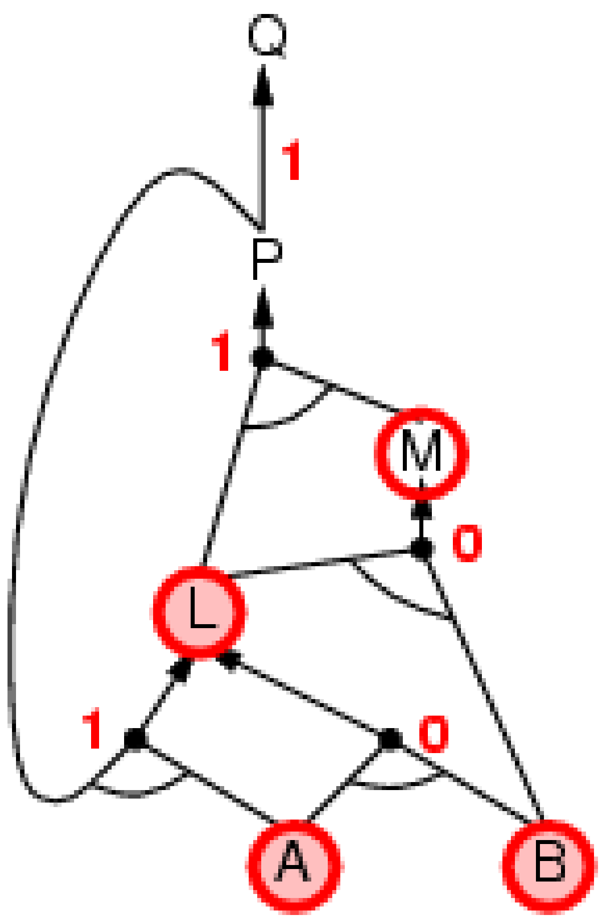
前向链接

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



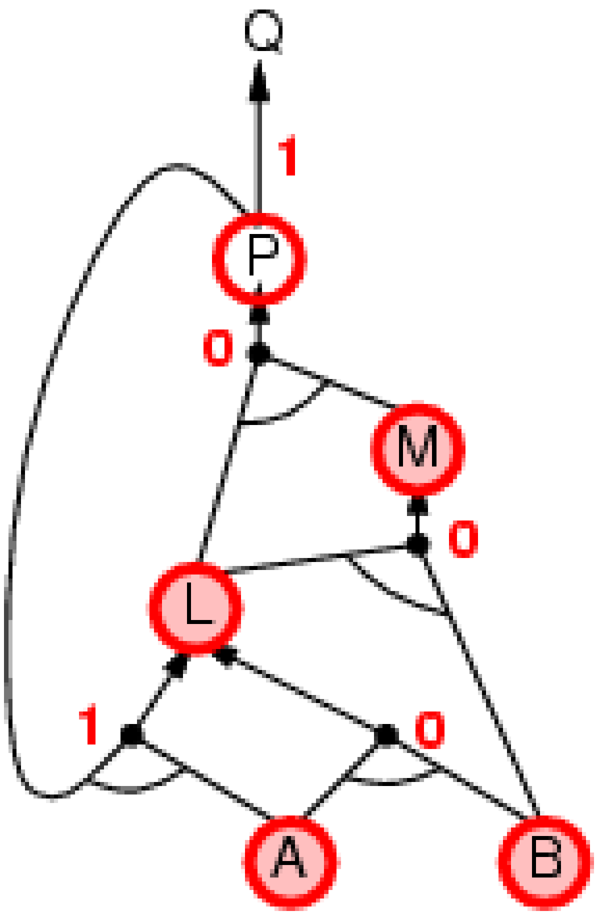
前向链接

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



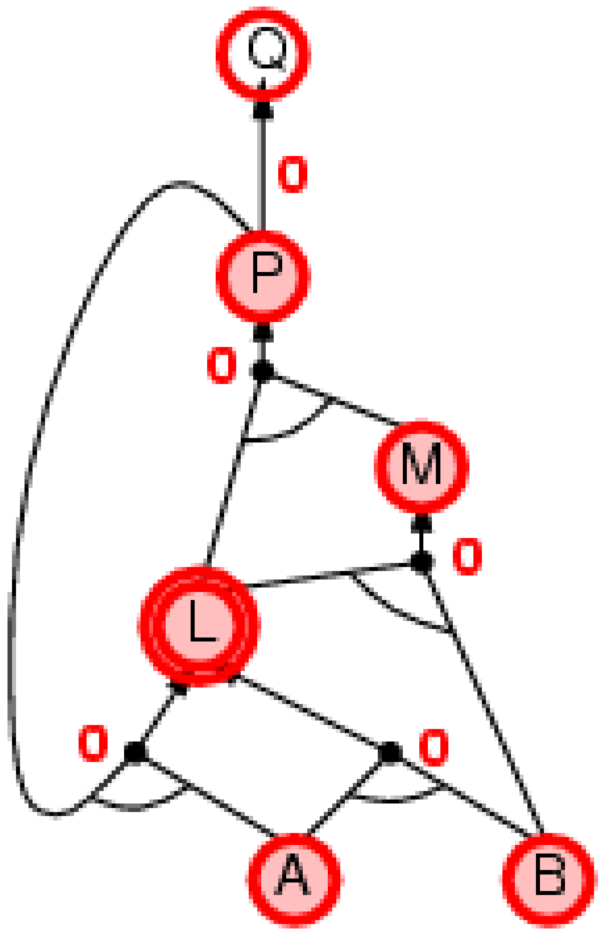
前向链接

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$



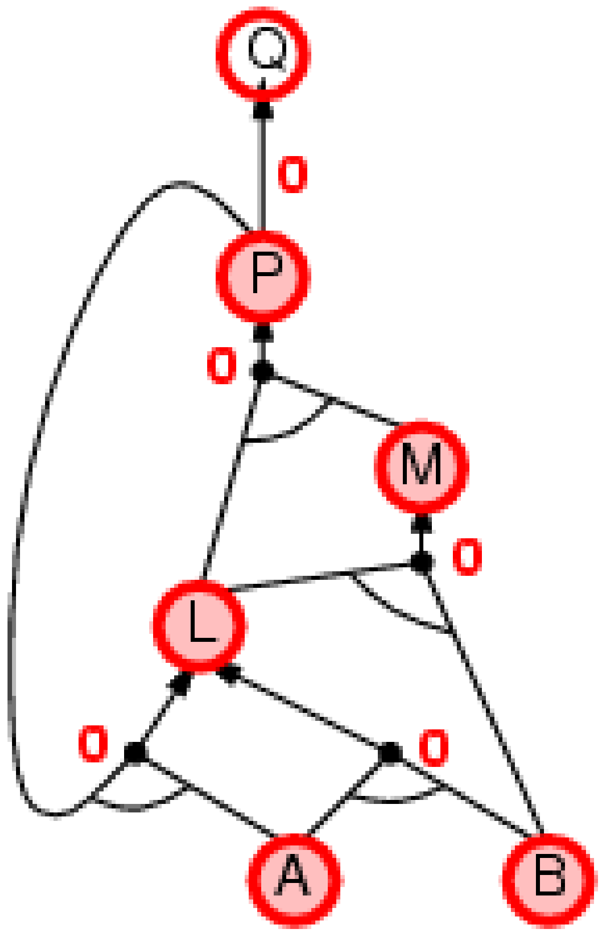
前向链接

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$



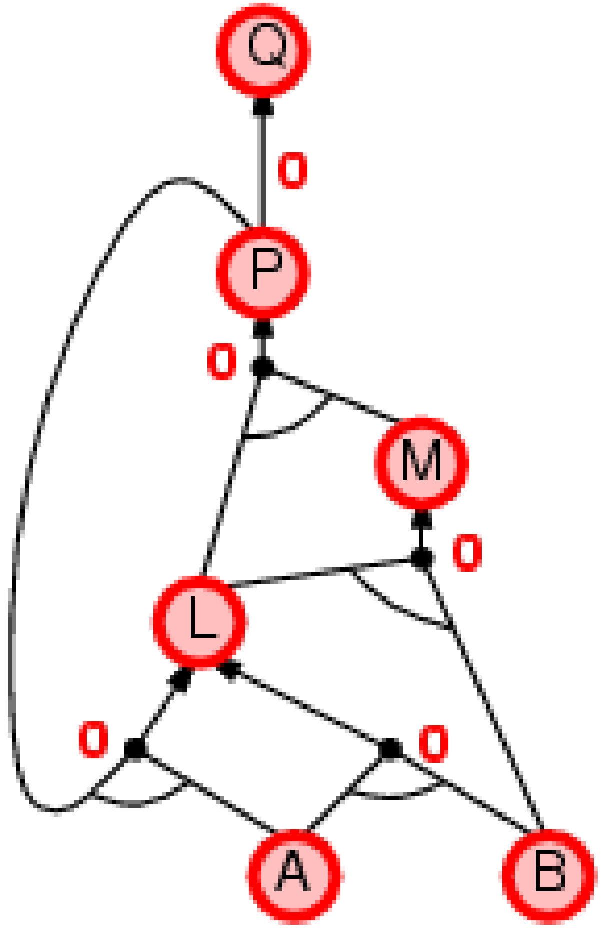
前向链接

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$



前向链接

$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$



反向链接

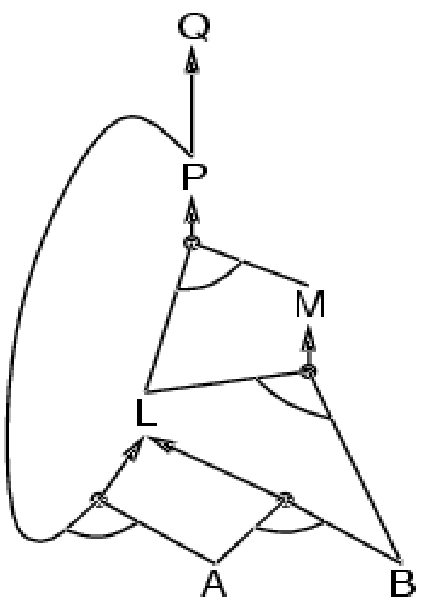
- 反向链接从查询反向开始(目标驱动)

  考虑前页的推理

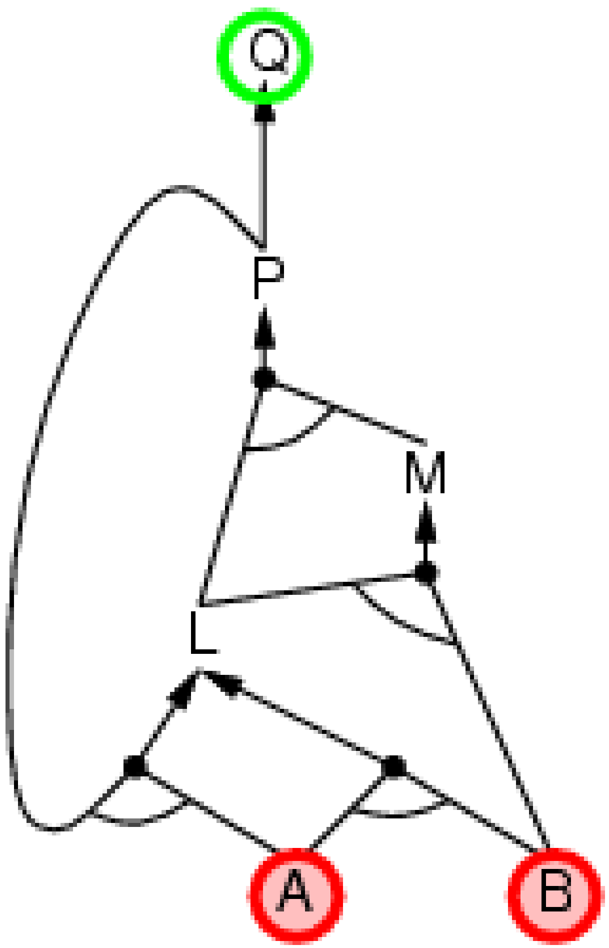
  KB: P⇒Q, L∧M⇒P, B∧L⇒M, A∧P⇒L, A∧B⇒L, A, B

  Q: Is KB |= ¬ Q？

$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

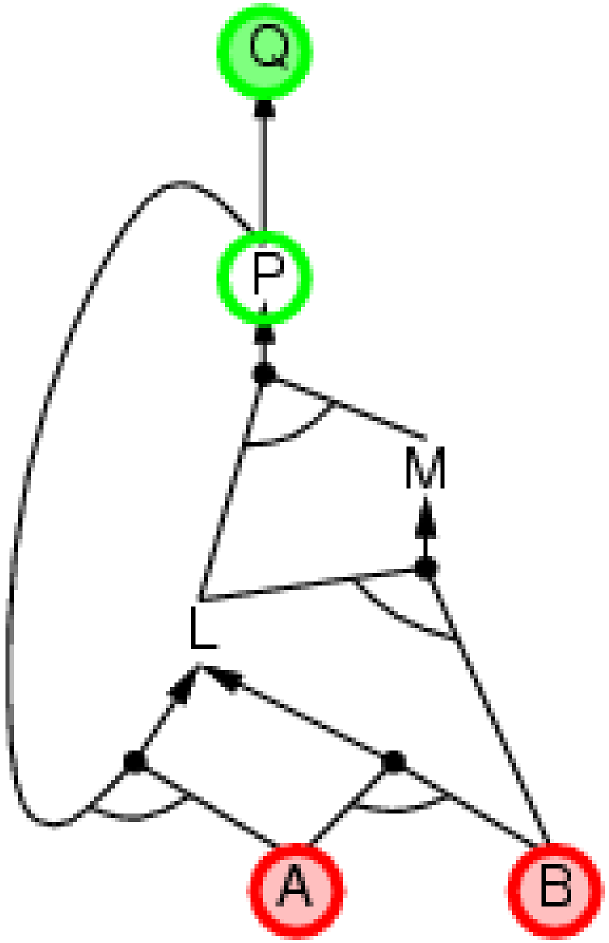


反向链接



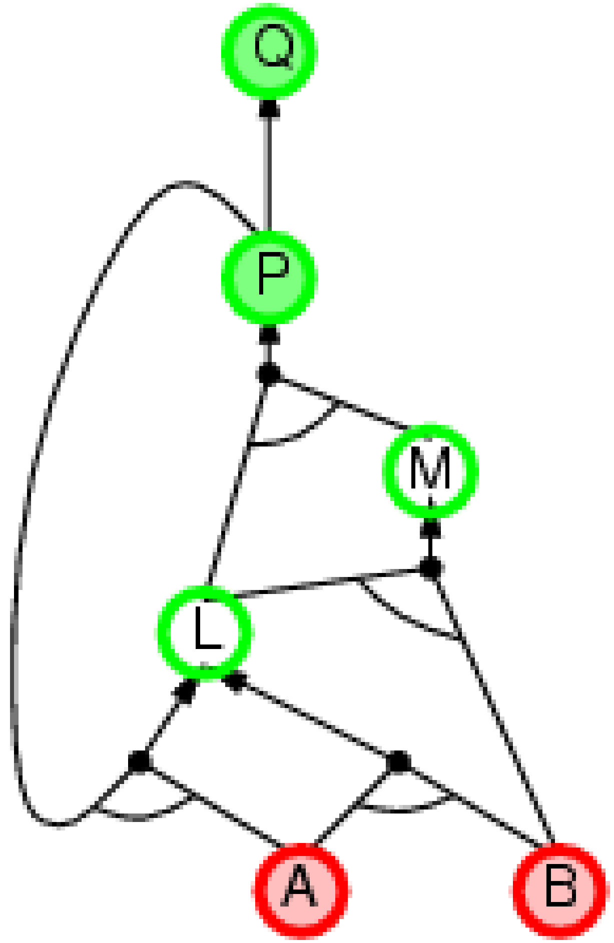
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

反向链接



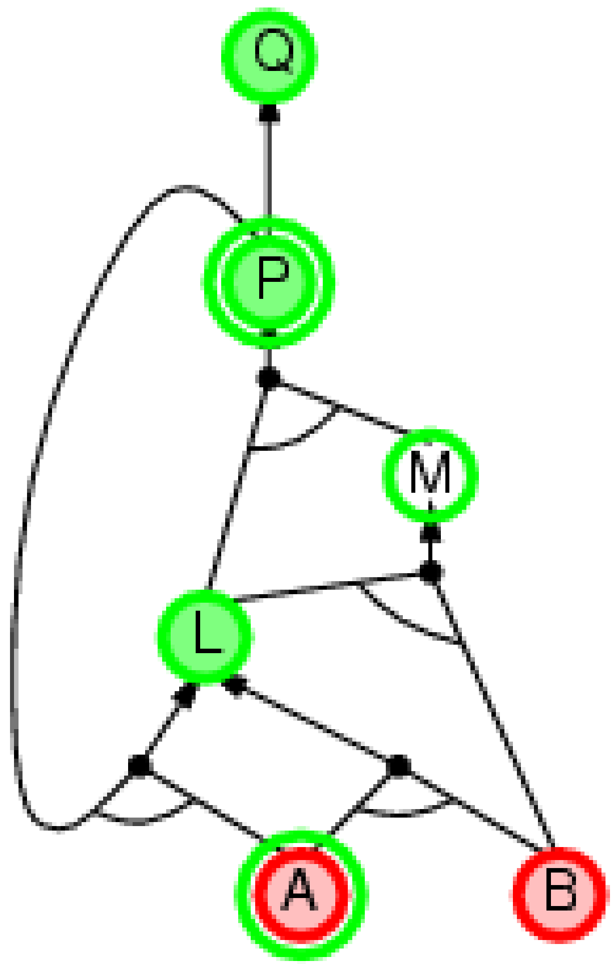
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

反向链接



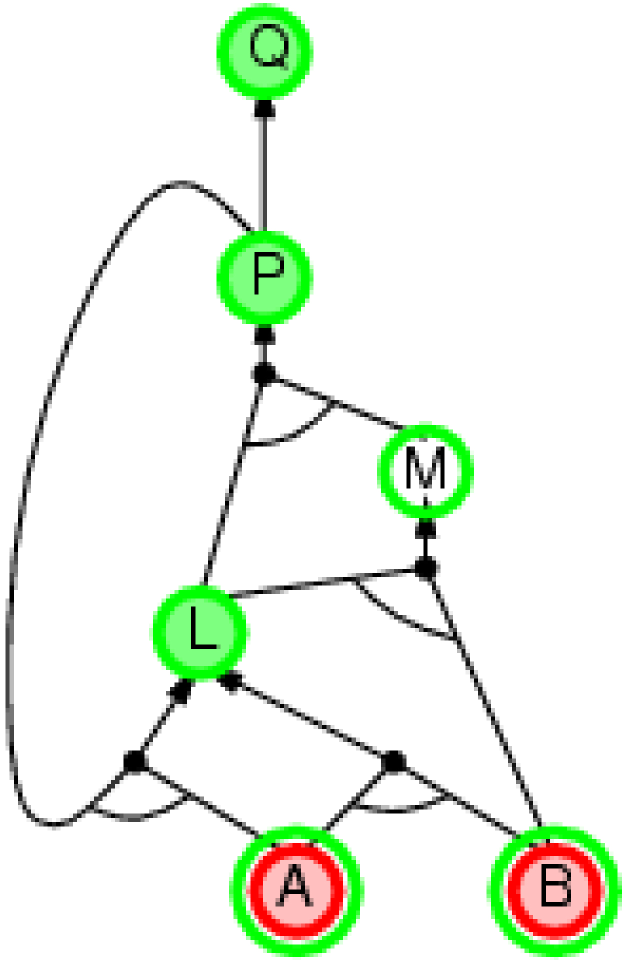
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

反向链接



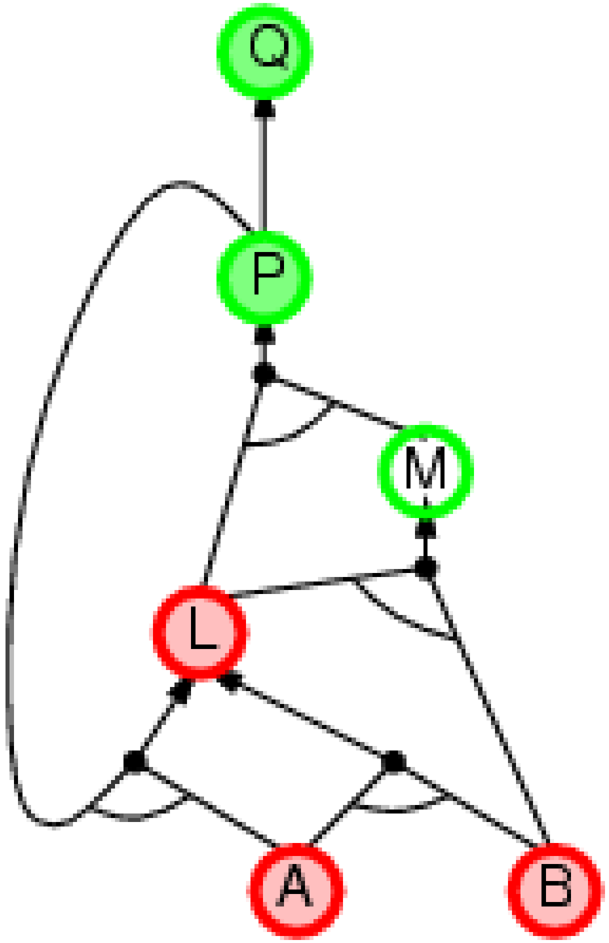
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

反向链接



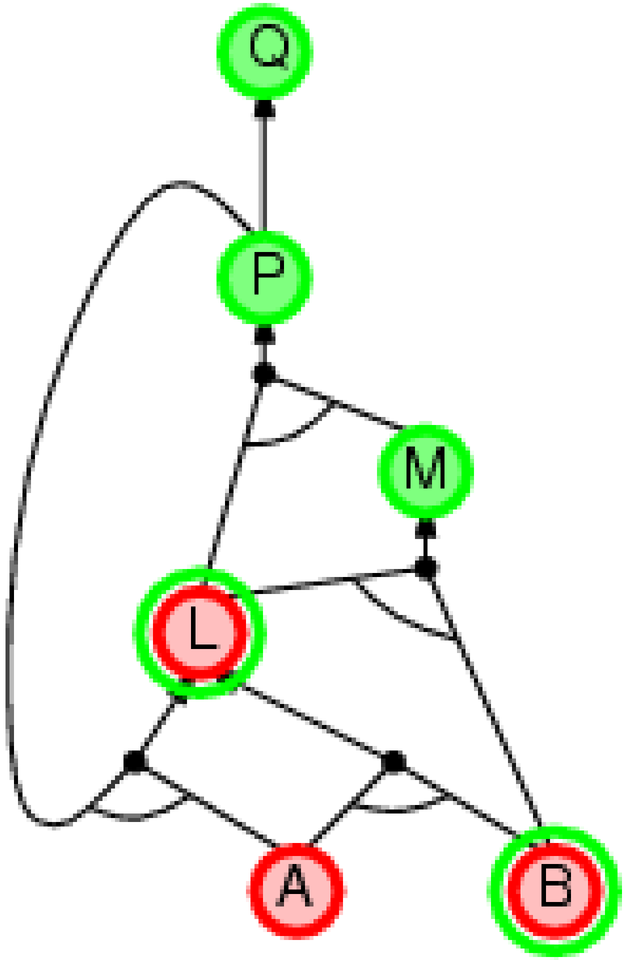
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

反向链接



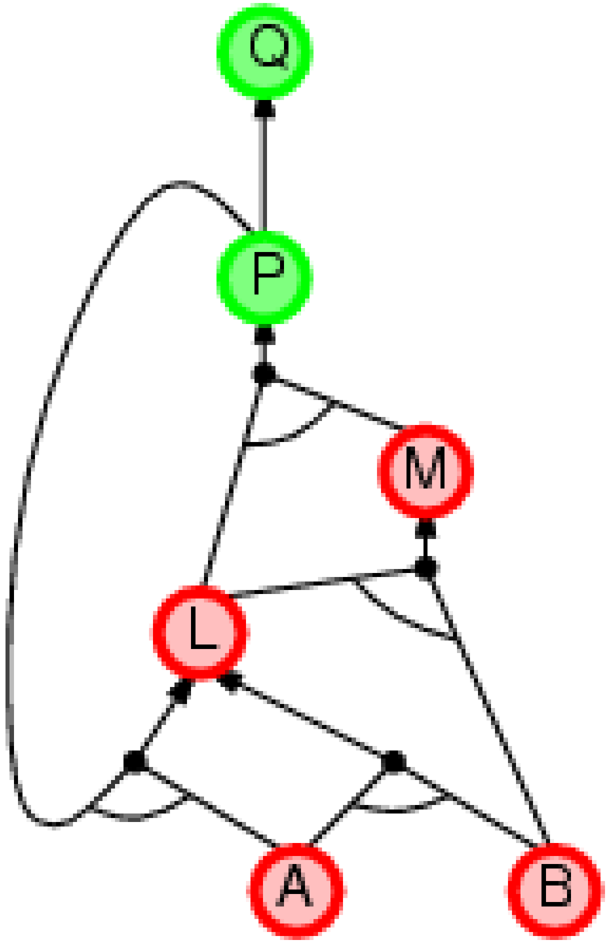
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

反向链接



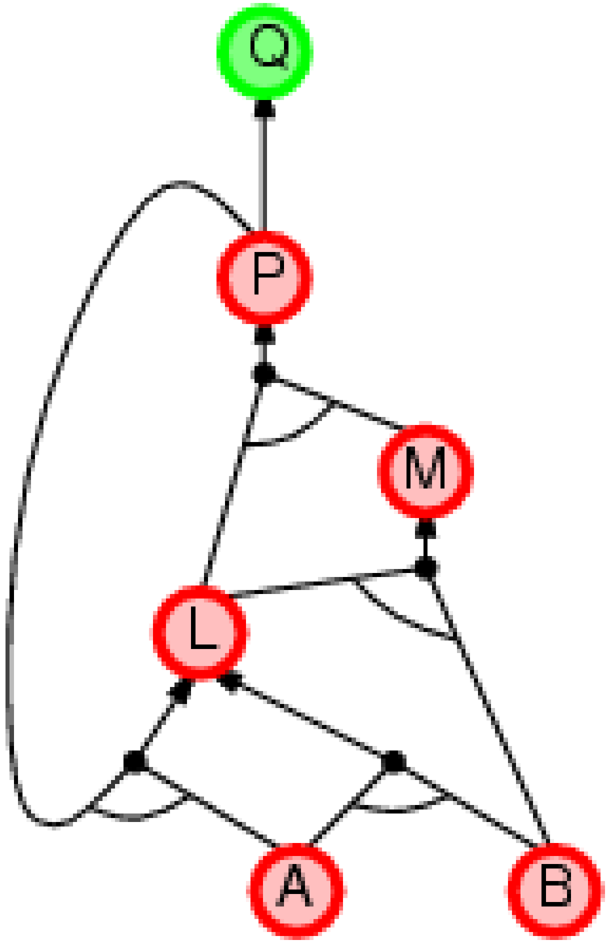
$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

反向链接



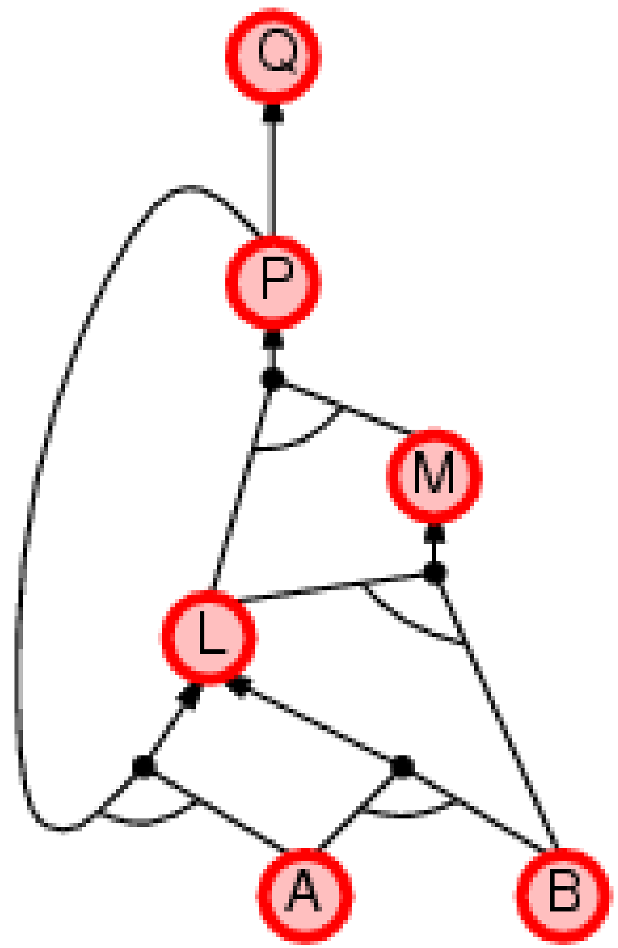
$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

反向链接



$$P \Rightarrow Q$$
$$L \wedge M \Rightarrow P$$
$$B \wedge L \Rightarrow M$$
$$A \wedge P \Rightarrow L$$
$$A \wedge B \Rightarrow L$$
$$A$$
$$B$$

反向链接



$$
\begin{aligned}
P &\Rightarrow Q \\
L \wedge M &\Rightarrow P \\
B \wedge L &\Rightarrow M \\
A \wedge P &\Rightarrow L \\
A \wedge B &\Rightarrow L \\
A & \\
B &
\end{aligned}
$$

前向链接和反向链接的对比

- 前向链接：data-driven，适合于目标识别、路径规划等，但会做与目标无关的事
- 反向链接：goal-driven，适合于问题求解
- 通常，反向链接的耗散远小于KB大小的线性值
- 对于智能体来说，应该**共享前向和反向推理的工作**，将前向推理限制在生成与要用反向链接求解的查询有关的事实上

6.6 命题推理的有效算法

- 用于检验可满足性：一个句子是可满足的，如果它在某些模型中为真
  - 基于回溯搜索
  - 基于爬山法搜索
- 计算机科学中许多组合问题可以简化为检验命题语句的可满足性

6.7 基于命题逻辑的Agent：用逻辑推理寻找陷阱和wumpus

- 陈述“物理规则”的知识库：

$\neg P_{1,1}$

$\neg W_{1,1}$

$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y})$ (1)

$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y})$ (2)

$W_{1,1} \vee W_{1,2} \vee \ldots \vee W_{4,4}$

$\neg W_{1,1} \vee \neg W_{1,2}$

$\neg W_{1,1} \vee \neg W_{1,3}$

…

$\Rightarrow$64个符号和155个语句

```
function PL-WUMPUS-AGENT(percept) returns an action
    inputs: percept, a list, [stench, breeze, glitter]
    static: KB, initially containing the "physics" of the wumpus world
            x, y, orientation, the agent's position (init. [1,1]) and orient. (init. right)
            visited, an array indicating which squares have been visited, initially false
            action, the agent's most recent action, initially null
            plan, an action sequence, initially empty

    update x,y,orientation, visited based on action
    if stench then TELL(KB, S_{x,y}) else TELL(KB, ¬ S_{x,y})
    if breeze then TELL(KB, B_{x,y}) else TELL(KB, ¬ B_{x,y})
    if glitter then action ← grab
    else if plan is nonempty then action ← POP(plan)
    else if for some fringe square [i,j], ASK(KB, (¬ P_{i,j} ∧ ¬ W_{i,j})) is true or
            for some fringe square [i,j], ASK(KB, (P_{i,j} ∨ W_{i,j})) is false then do
        plan ← A*-GRAPH-SEARCH(ROUTE-PB([x,y], orientation, [i,j], visited))
        action ← POP(plan)
    else action ← a randomly chosen move
    return action
```

用命题逻辑来辨别陷阱、wumpus和安全方格的智能体

命题逻辑的表达能力的局限性

- KB包含以(1)和(2)中给出形式表示的每个方格的“物理规则”语句，环境规模越大，要求的初始KB就越大。

$$B_{x,y} \Leftrightarrow (P_{x,y+1} \vee P_{x,y-1} \vee P_{x+1,y} \vee P_{x-1,y}) \quad (1)$$

$$S_{x,y} \Leftrightarrow (W_{x,y+1} \vee W_{x,y-1} \vee W_{x+1,y} \vee W_{x-1,y}) \quad (2)$$

- 如果要记录位置与方向，KB需包含以下形式的语句：

$$L_{x,y}^{t} \wedge FacingRight^{t} \wedge Forward^{t} \Rightarrow L_{x+1,y}^{t+1}$$

- 随着时间步t的增加，知识库的子句将激增

7.8 小结

- 智能Agent 需要关于世界的知识以达到良好的决策
  - 逻辑智能体对知识库进行推理，以导出新的语句，并用它们来决策。
- 逻辑的基本概念
  - 表示语言通过语法和语义来定义
  - 语句之间的蕴涵关系
  - 推理：可靠性和完备性
- 归结规则产生一个用于表示为合取范式的知识库的完备推理算法。
- 前向链接和反向链接都是用于以Horn Form表示的知识库的非常自然的推理算法
- 命题逻辑缺乏表达能力

作业

7.2　假定智能体已经前进到图 7.4（a）所示的位置，感知到的情况为：[1, 1]什么也没有，[2, 1]有微风，[1, 2]有臭气。它现在想知道[1, 3]、[2, 2]和[3, 1]的情况。这 3 个位置中的每一个都可能包含陷阱，而最多只有一个可能有 wumpus。按照图 7.5 的实例，构造出可能世界的集合。（你应该找到 32 个。）把 KB 为真以及下列每个语句都为真的世界标出来：

α_2 = "[2, 2]中没有陷阱。"

α_3 = "[1, 3]中有一只 wumpus。"

据此证明 $KB \models \alpha_2$ 和 $KB \models \alpha_3$。

习题7.2

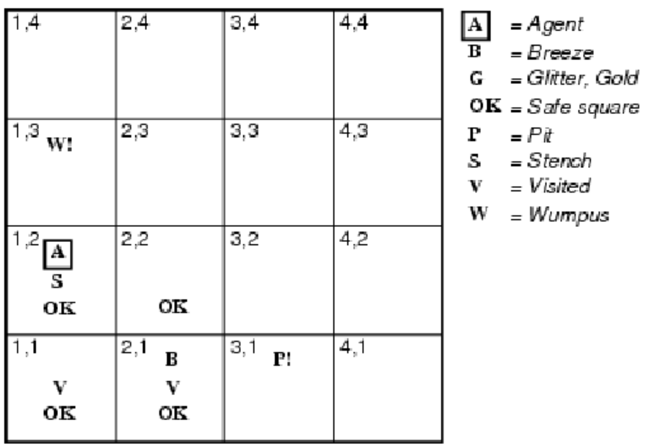


图7.4　(a)

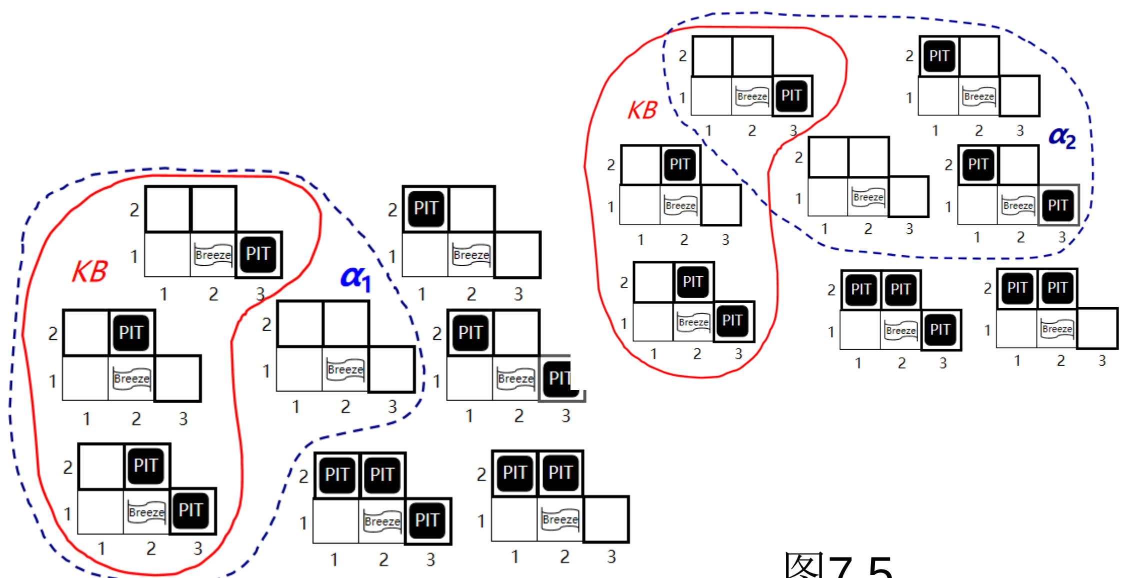


图7.5