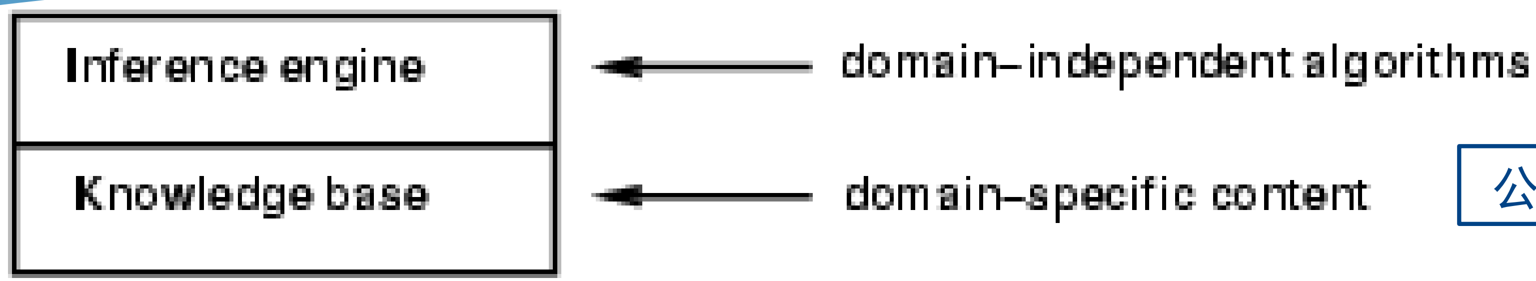
Chapter 7
Logical Agent

王子磊 (Zilei Wang)
Email: zlwang@ustc.edu.cn
http://vim.ustc.edu.cn

提纲

- 基于知识的 Agent
- Wumpus 世界
- 逻辑 (Logic)
- 命题逻辑 (Propositional/Boolean logic)
- 命题逻辑推理与定理证明 (Theorem proving)
  - 模型检验
  - 归结证明
  - 前向和反向链接

知识库



公理+定理

知识库 (Knowledge Base, *KB*) = 一种正式语言描述的语句集合

- 声明用来建立一个agent系统
  - Tell 知识库系统需要知道的内容
- 然后 Ask 知识库应该执行什么行动
  - answers should follow from the KB
- 从知识层面看
  - i.e., what they know, regardless of how implemented
- 从实现层面看
  - i.e., data structures in KB and algorithms that manipulate them

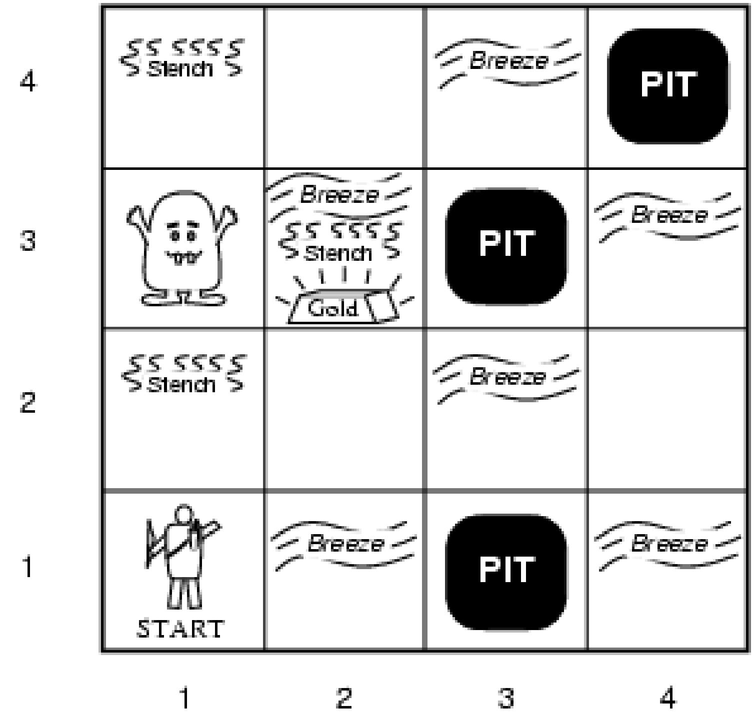
A simple knowledge-based agent

```
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
```

- Agent 必须能够：
  - 表达状态、行动等
  - 纳入新的感知内容
  - 更新世界的内部表示 (internal representations)
  - 推断世界的隐命题 (hidden properties)
  - 推断合理的行动

Wumpus 世界 —— PEAS描述

- 性能度量 (Performance measure)
  - gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- 环境 (Environment, 4 X 4)
  - Wumpus 相邻有臭气 Stench
  - 无底洞(PIT)相邻有微风 Breezy
  - 黄金所处方格闪闪发光 Glitter
  - 面向Wumpus射击时可以杀死它 Shoot
  - 只能射击一次
  - 在黄金方格内可以捡起它 Grab
  - 在黄金方格内可以丢下它 Release

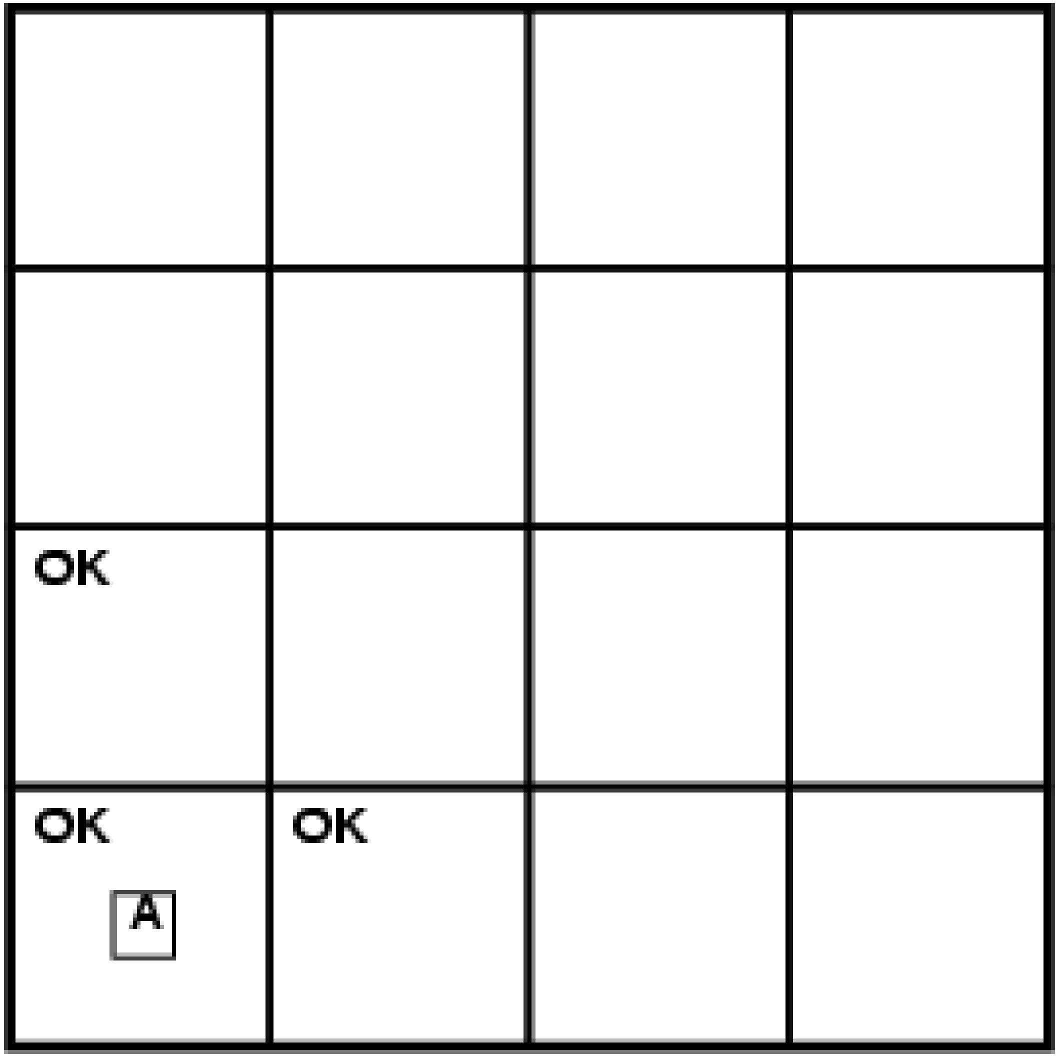


- 传感器 (Sensors): [Stench, Breeze, Glitter, Bump, Scream]
- 执行器 (Actuators): Left turn, Right turn, Forward, Grab, Release, Shoot

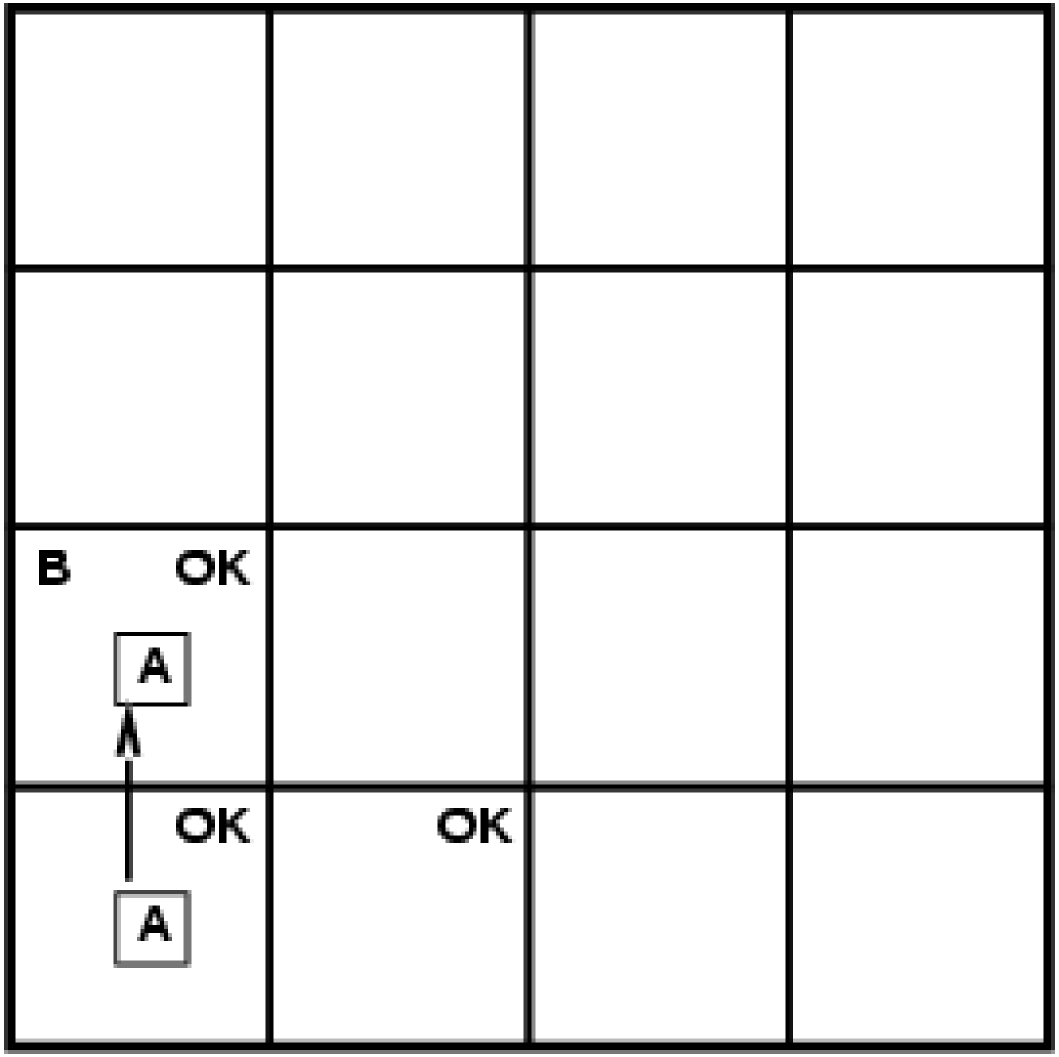
Wumpus 世界的特点

- 完全可观的 (Fully Observable)? No – only local perception
- 确定性的 (Deterministic)? Yes – outcomes exactly specified
- 片段的 (Episodic)? No – sequential at the level of actions
- 静态的 (Static)? Yes – Wumpus and Pits do not move
- 离散的 (Discrete)? Yes
- 单代理的 (Single-agent)? Yes – Wumpus is essentially a natural feature

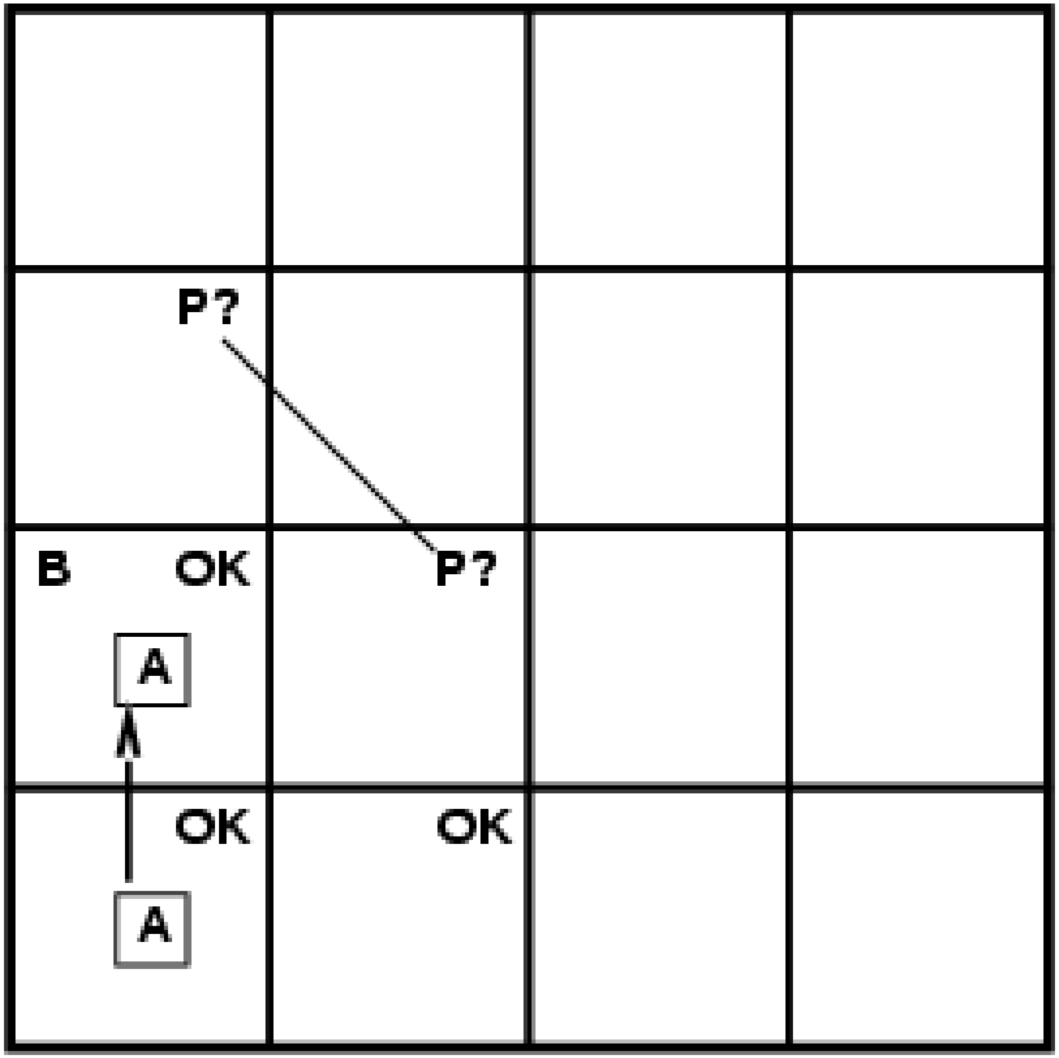
Wumpus 世界中探索



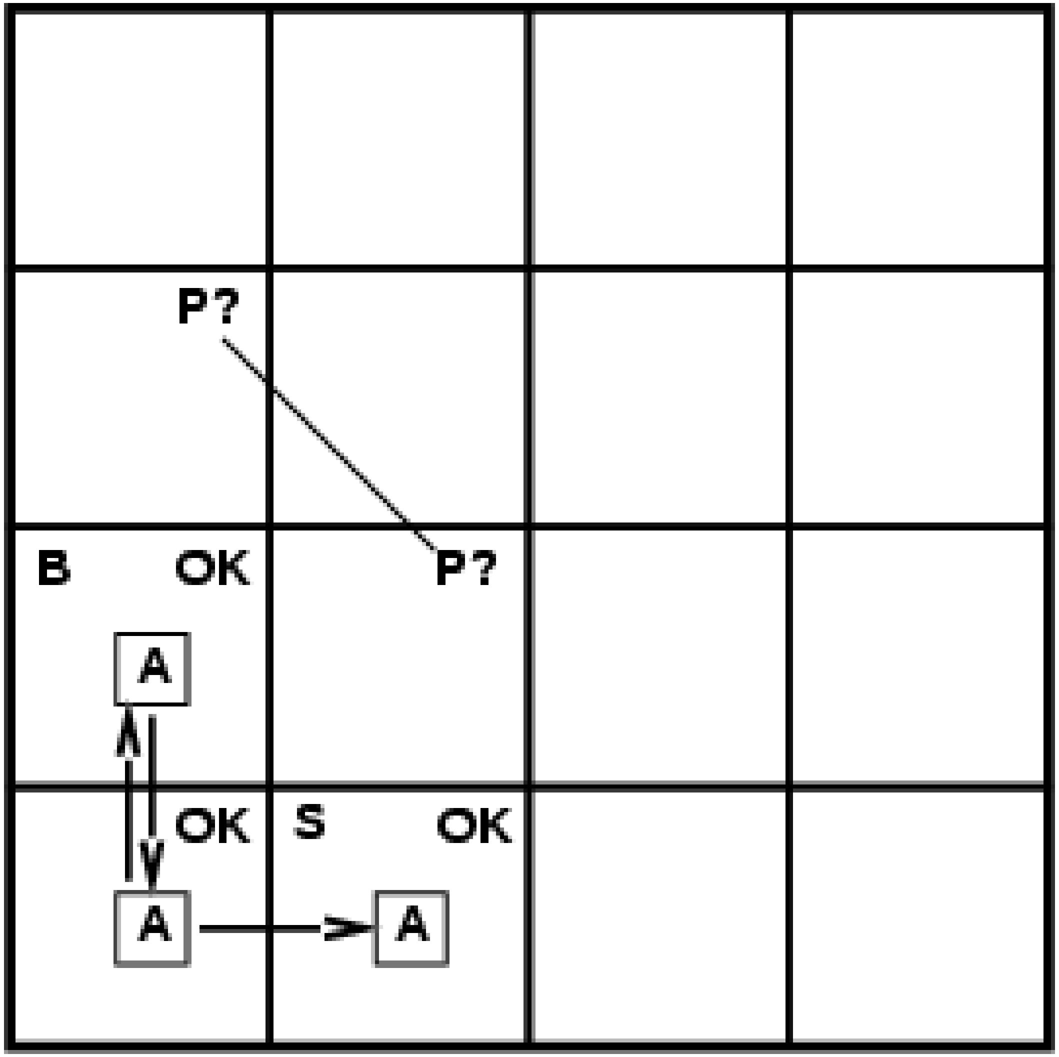
Wumpus 世界中探索



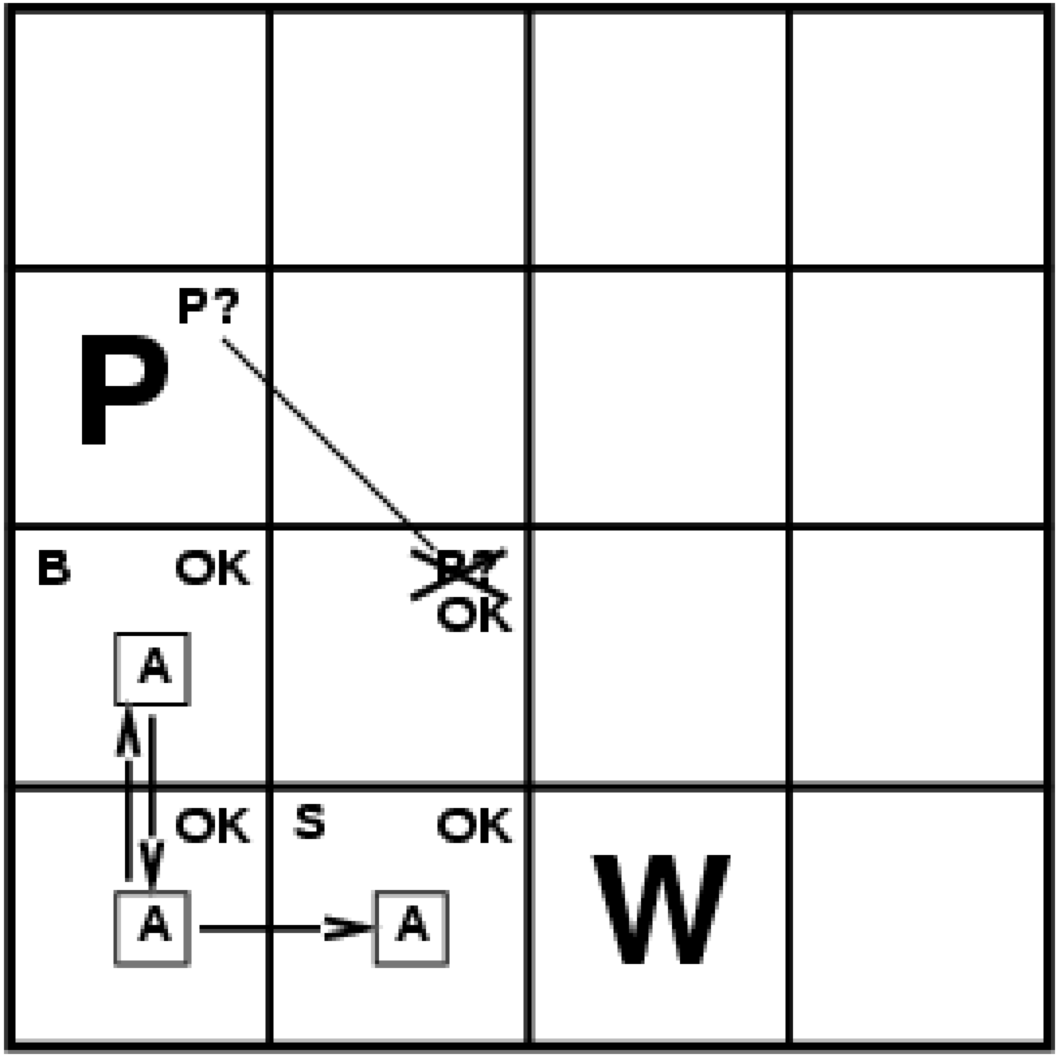
Wumpus 世界中探索



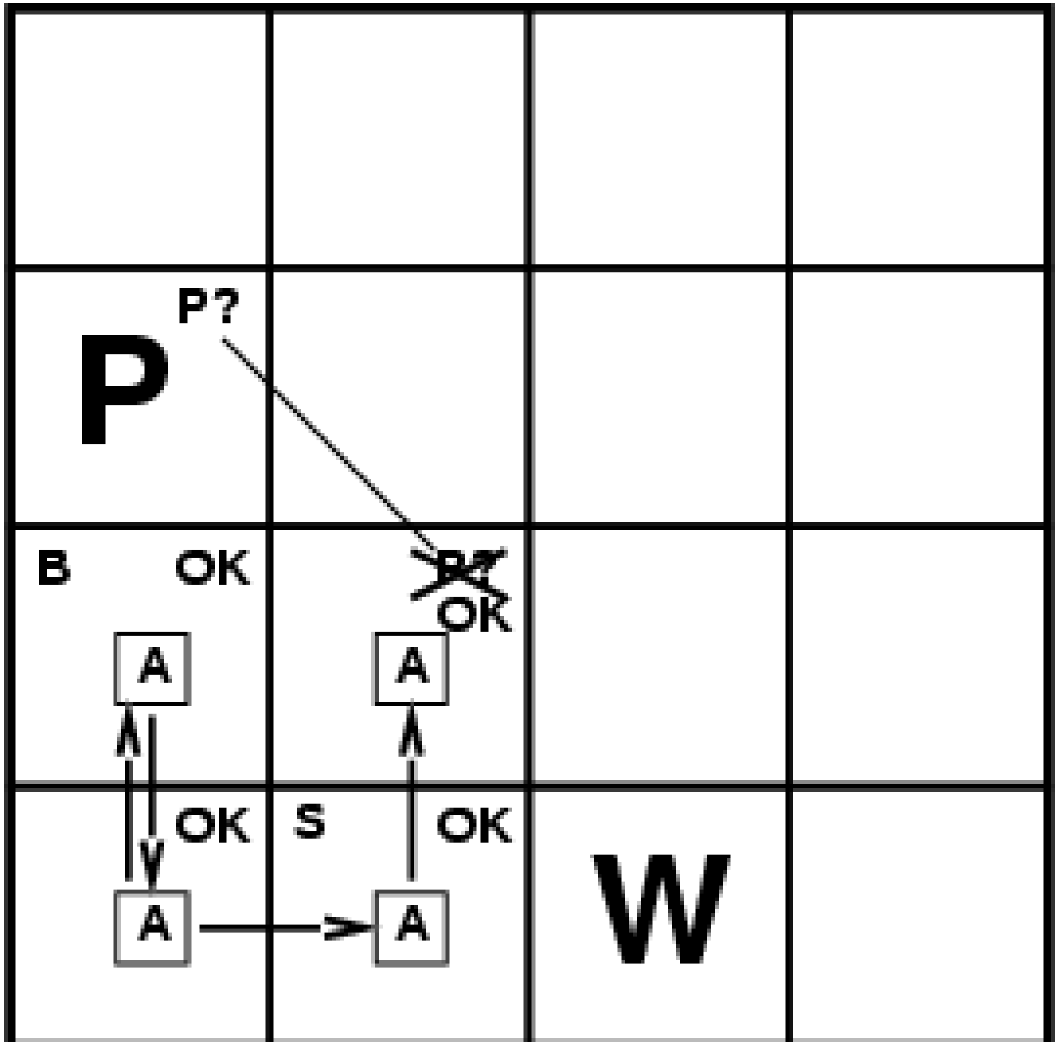
Wumpus 世界中探索



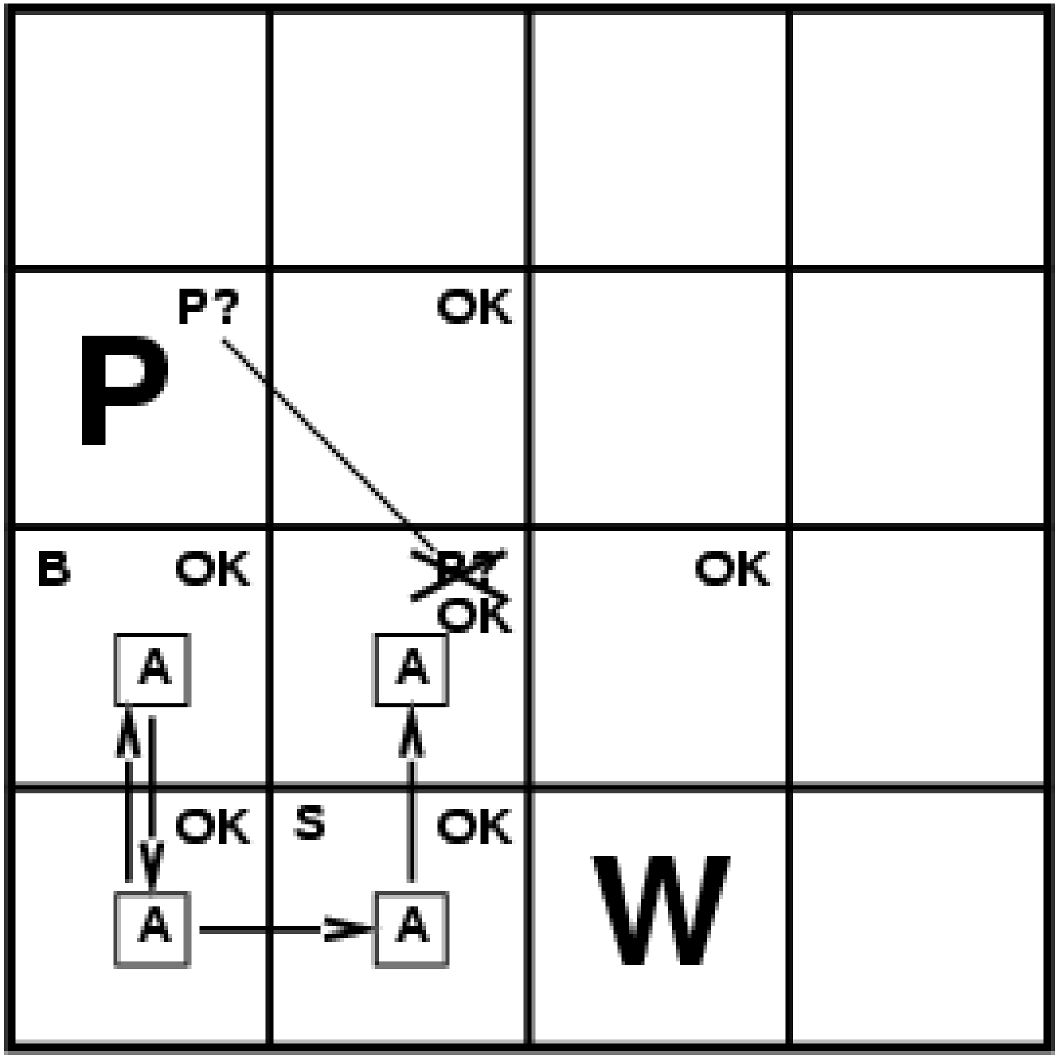
Wumpus 世界中探索



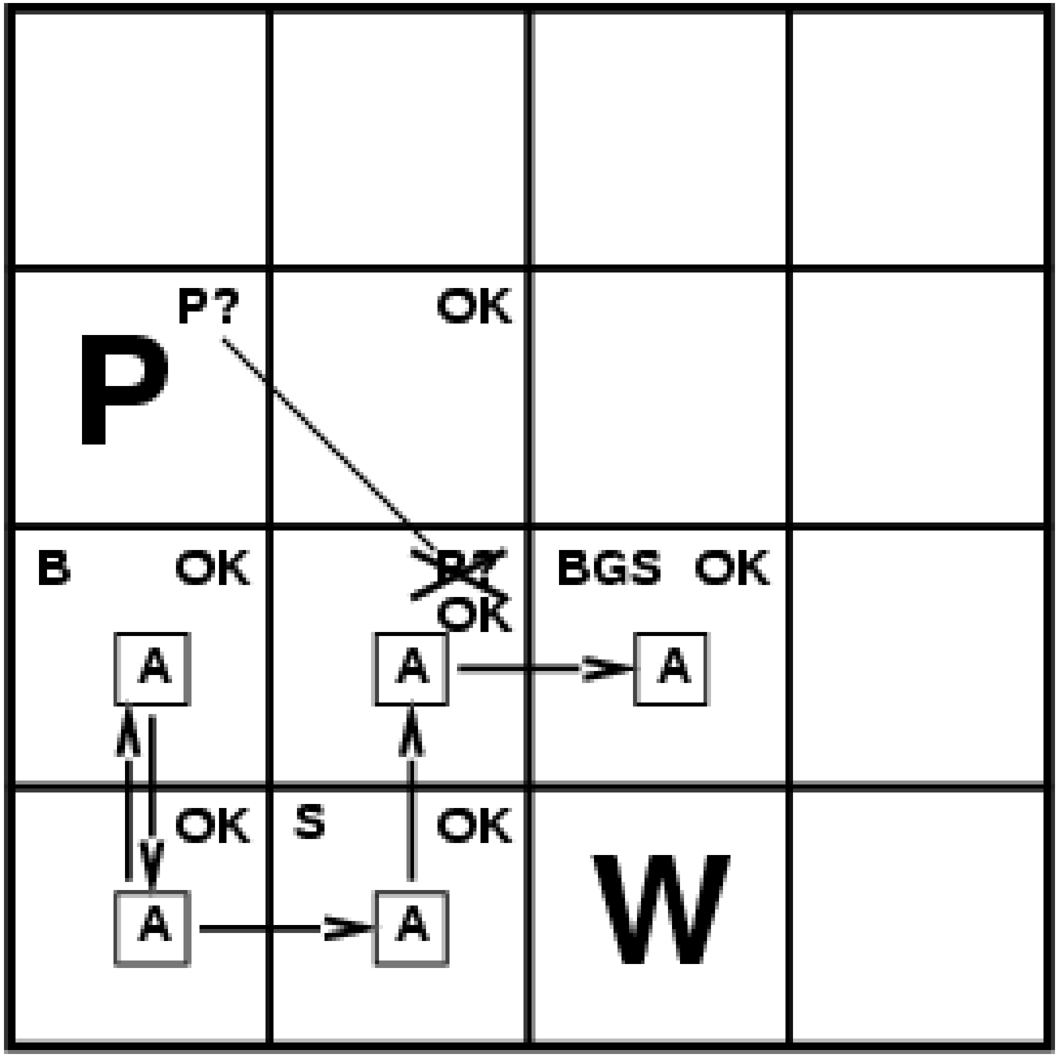
Wumpus 世界中探索



Wumpus 世界中探索



Wumpus 世界中探索



逻辑

- 逻辑 (Logics) 是表达信息的正式语言
  - 能够进行结论的推导
- 语法 (Syntax) 定义了合法语句的规范
- 语义 (Semantics) 表达了语句的含义
  - i.e., 定义了一个语句在可能世界中的真值 (truth)
- 例如， 在算术语言中
  - $x+2 \geq y$ is a sentence; $x2+y > \{\}$ 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 $x = 0, y = 6$

蕴涵 (Entailment)

- 蕴含：某语句逻辑上跟随 (follows from) 另外一个语句

$$KB \models \alpha$$

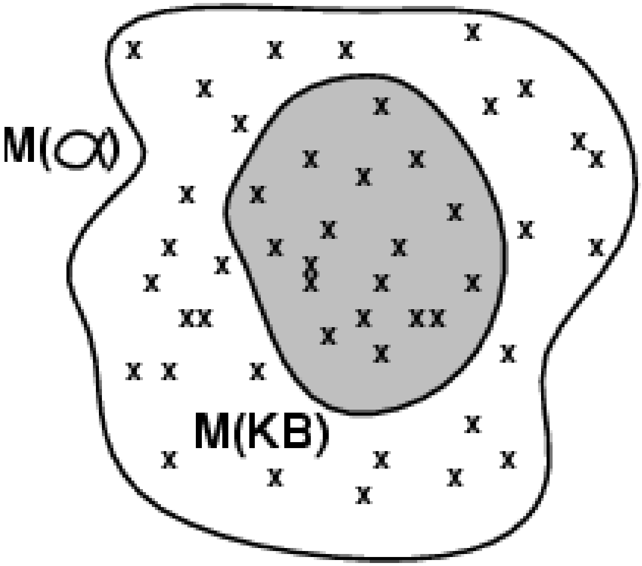
- 知识库 KB 蕴含语句 α
  - 当且仅当 α 在 KB 所有为真的世界中都为真
  - E.g., the *KB* containing "the Giants won" and "the Reds won" entails "Either the Giants won or the Reds won"
  - E.g., x+y = 4 entails 4 = x+y

- 蕴含表达了语句(i.e., syntax)之间的一种语义(semantics)关系

模型 (Models)

- 模型：逻辑学家的典型用语，用于评估真值的结构化世界
- 当 α 在 m 中为真时，m 是语句 α 的模型
- $M(\alpha)$ 表示 α 所有模型的集合
- $KB \models \alpha$ 当且仅当 $M(KB) \subseteq M(\alpha)$
  - E.g. KB = Giants won and Reds won

    α = Giants won

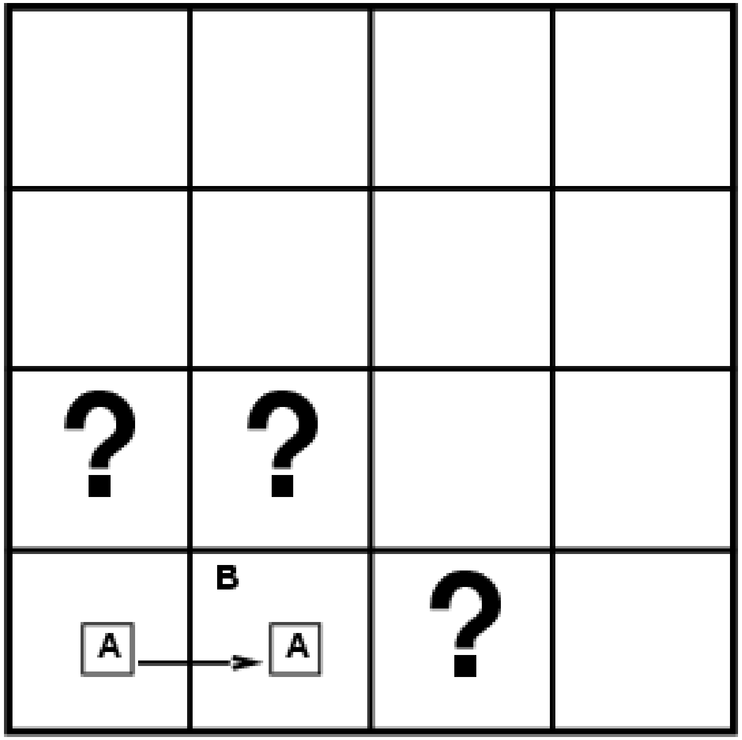


Wumpus 世界中的蕴含

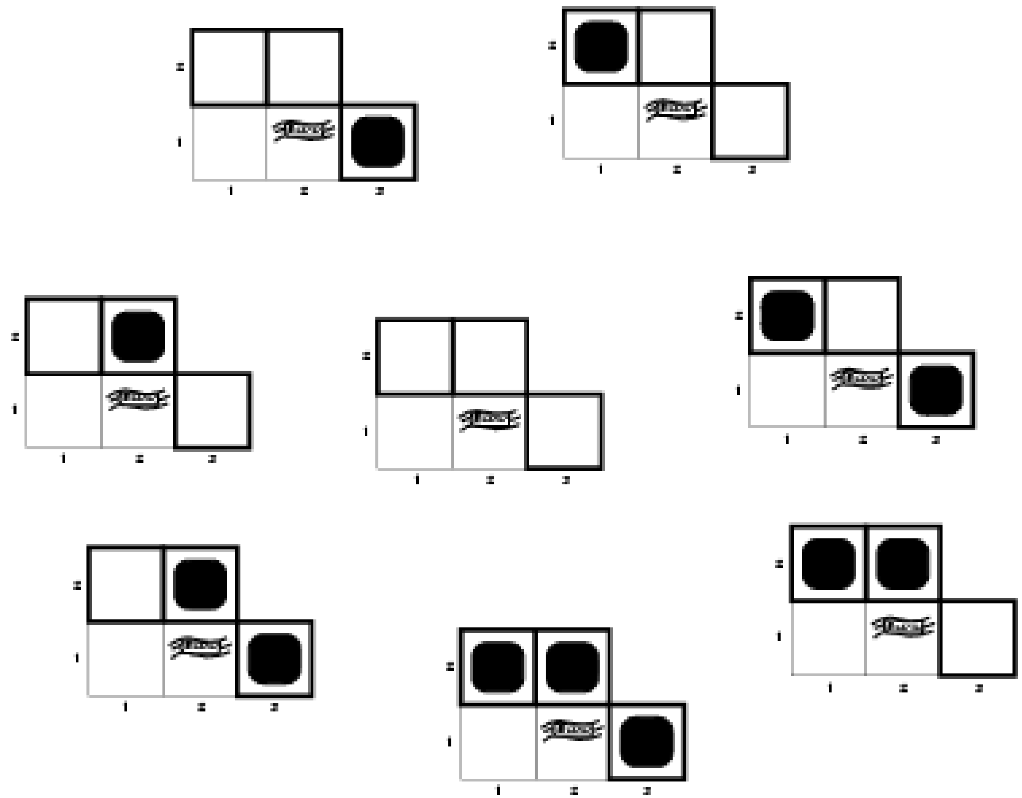
状况：在 [1,1]中什么都没有检测到，然后右移，在 [2,1]中检测到微风 (breeze)

考虑 *KB* 可能的模型（假设只考虑pits问题）

3 Boolean choices ⇒ 8 possible models

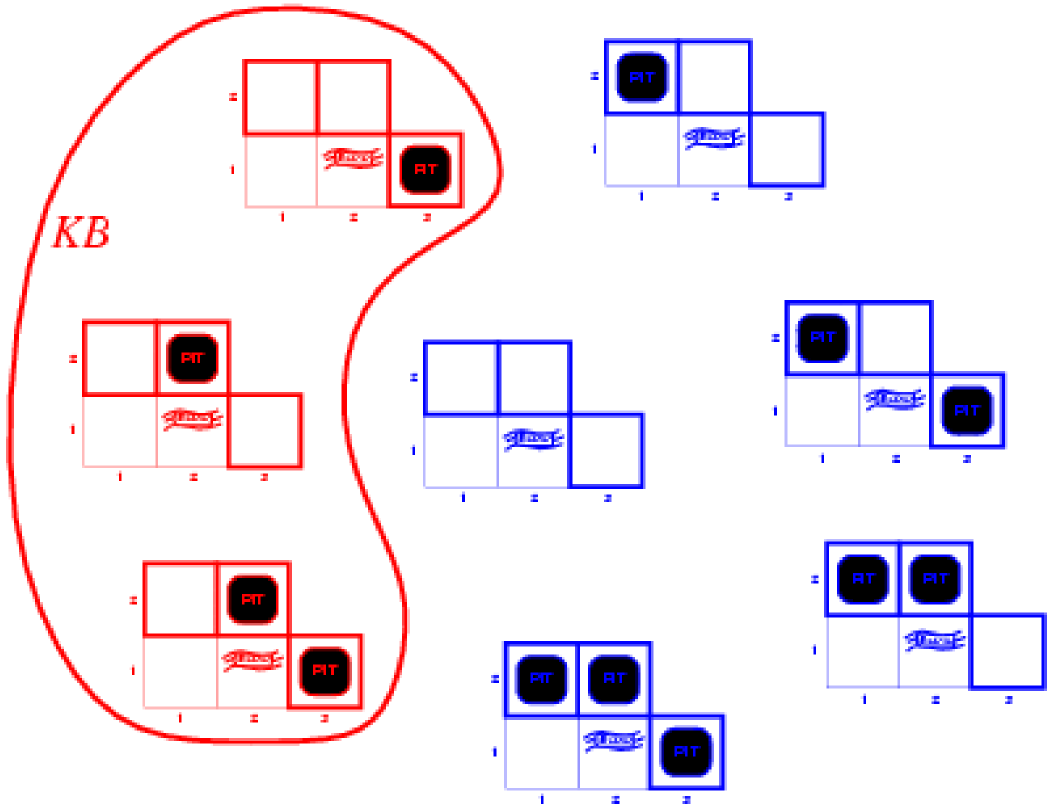


Wumpus 模型



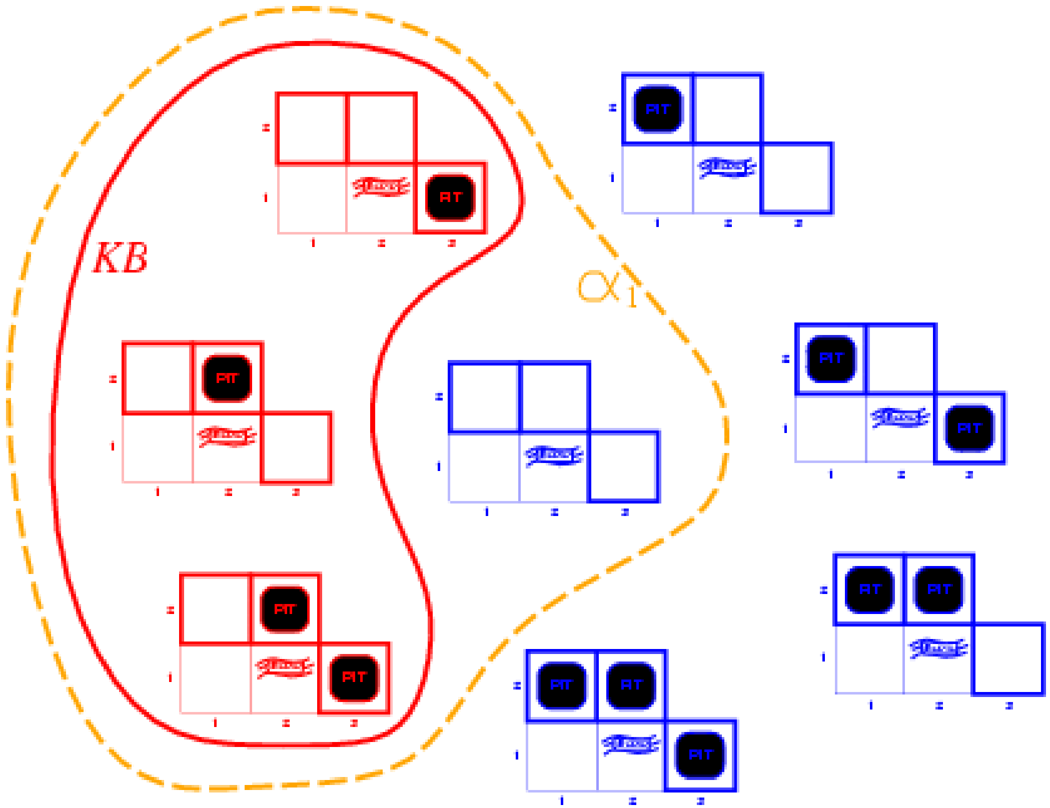
❖ [1,2], [2,2], [3,1] 中有无陷阱的部分模型

Wumpus 模型



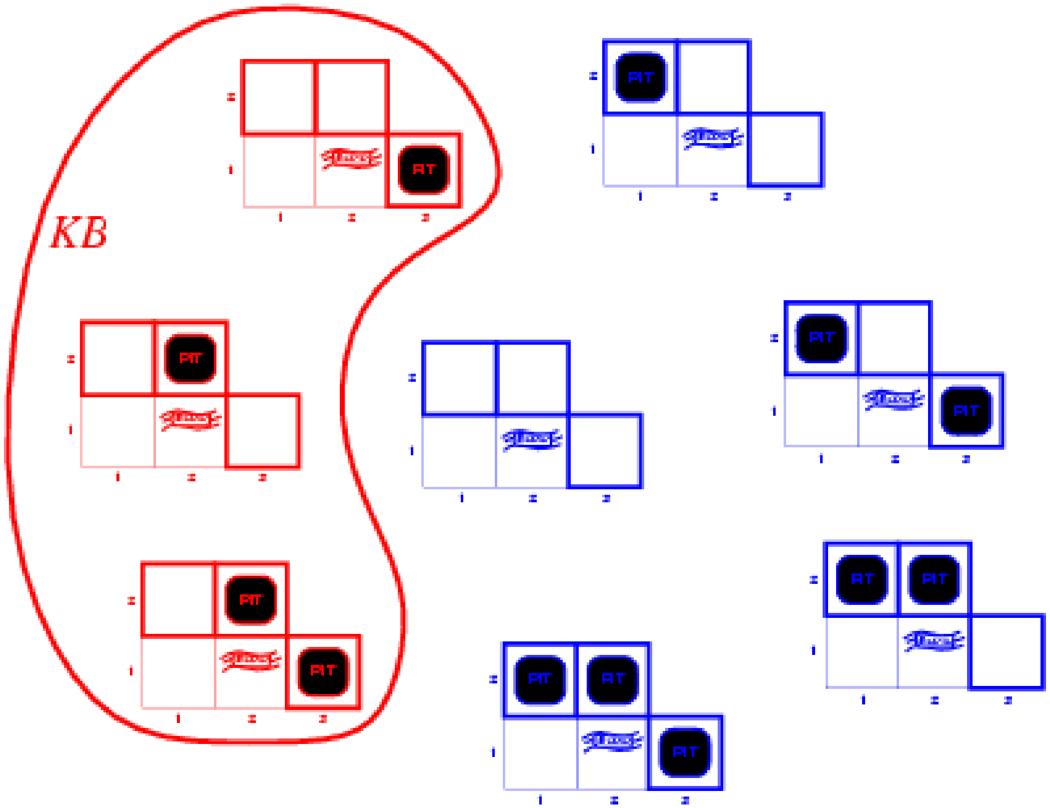
- *KB* = wumpus-world rules + observations (观测)
  - [1,1]什么都没有 + [2,1]有微风

Wumpus 模型



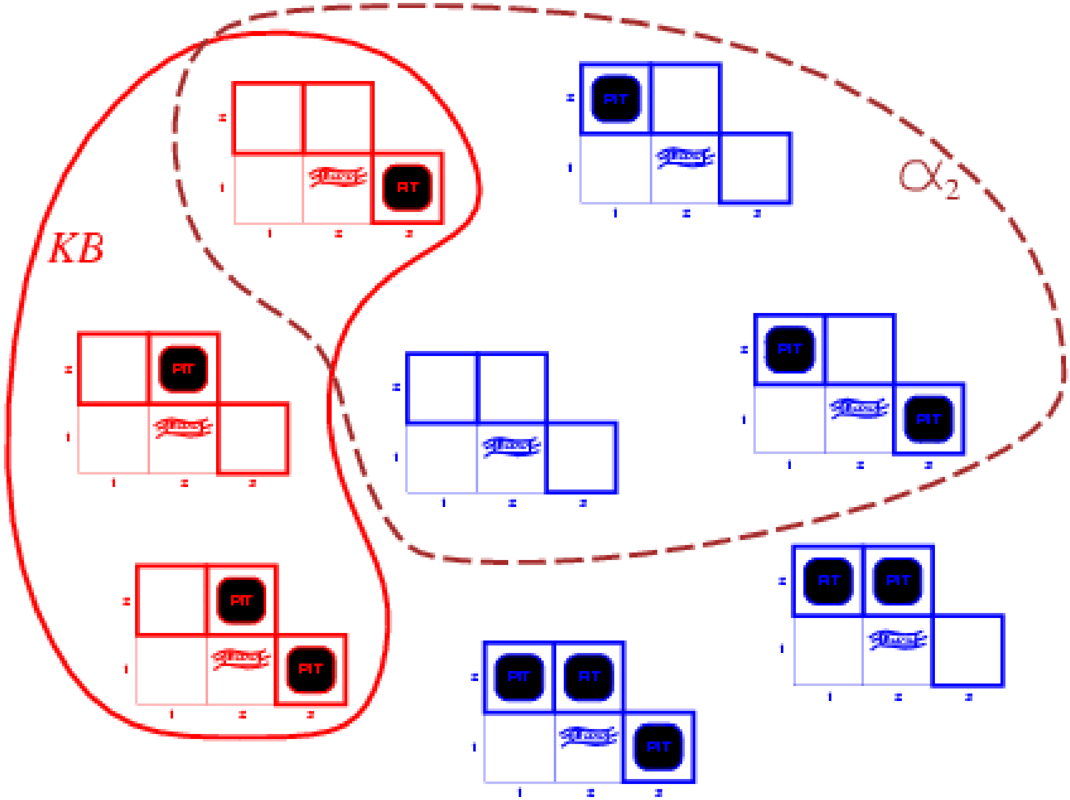
- *KB* = wumpus-world rules + observations (观测)
- 模型检验 (model checking)： α_1 = "[1,2] is safe", $KB \models \alpha_1$

Wumpus 模型



❖ *KB* = wumpus-world rules + observations (观测)

Wumpus 模型



- *KB* = wumpus-world rules + observations (观测)
- α_2 = "[2,2] is safe", $KB \not\models \alpha_2$

推理 (Inference)

$KB \vdash_i \alpha$ = sentence α can be derived from *KB* by procedure ***i***

- 推理算法 *i*
  - 可靠性(Soundness): 只导出蕴涵句
    - *i* is sound if whenever $KB \vdash_i \alpha$, it is also true that $KB \models \alpha$
  - 完备性(Completeness): 生成任一蕴涵句
    - *i* is complete if whenever $KB \models \alpha$, it is also true that $KB \vdash_i \alpha$

- **Preview**: we will define a logic (first-order logic) which is expressive enough to say almost anything of interest, and for which there exists a sound and complete inference procedure.
  - That is, the procedure will answer any question whose answer follows from what is known by the *KB*.

命题逻辑：语法

- 命题逻辑 (Propositional logic) 是一种最简单的逻辑
  - 这里主要用于说明逻辑及推理的基本思想

- 命题符号 S_1, S_2 等是语句
  - $\neg$非：If S is a sentence, $\neg S$ is a sentence (否定式，negation)
  - $\wedge$与：If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (合取式，conjunction)
  - $\vee$或：If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (析取式，disjunction)
  - $\Rightarrow$蕴含：If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (蕴含式，implication)
  - $\Leftrightarrow$当且仅当：If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence （双向蕴含式，biconditional）

命题逻辑：语义

每个命题符号在模型中指定了真值 true/false

E.g.	$P_{1,2}$	$P_{2,2}$	$P_{3,1}$
	false	*true*	*false*

(这些符号有 8 个可能的模型，能够自动列出）

针对任一模型 ***m***，真值计算规则如下：

$\neg S$	is true iff	S	is false			
$S_1 \wedge S_2$	is true iff	S_1	is true	and	S_2	is true
$S_1 \vee S_2$	is true iff	S_1	is true	or	S_2	is true
$S_1 \Rightarrow S_2$	is true iff	S_1	is false	or	S_2	is true
i.e.,	is false iff	S_1	is true	and	S_2	is false
$S_1 \Leftrightarrow S_2$	is true iff	$S_1 \Rightarrow S_2$	is true	and	$S_2 \Rightarrow S_1$	is true

简单递归就可以计算出任一语句的真值，如：

$$\neg P_{1,2} \wedge (P_{2,2} \vee P_{3,1}) = true \wedge (true \vee false) = true \wedge true = true$$

连接的真值计算表

P	Q	$\neg P$	$P \land Q$	$P \lor Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
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*

Truth tables for connectives

Wumpus 世界的语句

Let $P_{i,j}$ be true if there is a pit in $[i, j]$.

Let $B_{i,j}$ be true if there is a breeze in $[i, j]$.

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

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

$$B_{2,1}$$

- 规则："Pits cause breezes in adjacent squares“

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

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

“A square is breezy if and only if there is an adjacent pit"

推理真值表

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	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*

- 枚举每一行 (对各符号进行不同赋值)
  - 如果 KB 在该行为真，检查 α 是否也为真

枚举推理 (Inference by enumeration)

- 模型检验：深度优先的模型枚举是可靠和完备的

```
function TT-ENTAILS?(KB, α) returns true or false
   inputs: KB, the knowledge base, a sentence in propositional logic
           α, the query, a sentence in propositional logic

   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   递归算法
   if EMPTY?(symbols) then
      if PL-TRUE?(KB, model) then return PL-TRUE?(α, model)
      else return true      达到边界——模型检测
   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))   递归调用
```

- 对 n 个符号，时间复杂度为 $O(2^n)$，空间复杂度为 $O(n)$

逻辑等价 (Logical equivalence)

- 两个语句 α, β 是逻辑等价的，当且仅当在相同模型集中为真

  $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \land \beta)$	$\equiv$	$(\beta \land \alpha)$	commutativity of $\land$
$(\alpha \lor \beta)$	$\equiv$	$(\beta \lor \alpha)$	commutativity of $\lor$
$((\alpha \land \beta) \land \gamma)$	$\equiv$	$(\alpha \land (\beta \land \gamma))$	associativity of $\land$
$((\alpha \lor \beta) \lor \gamma)$	$\equiv$	$(\alpha \lor (\beta \lor \gamma))$	associativity of $\lor$
$\neg(\neg\alpha)$	$\equiv$	α	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv$	$(\neg\alpha \lor \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv$	$((\alpha \Rightarrow \beta) \land (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \land \beta)$	$\equiv$	$(\neg\alpha \lor \neg\beta)$	de Morgan
$\neg(\alpha \lor \beta)$	$\equiv$	$(\neg\alpha \land \neg\beta)$	de Morgan
$(\alpha \land (\beta \lor \gamma))$	$\equiv$	$((\alpha \land \beta) \lor (\alpha \land \gamma))$	distributivity of $\land$ over $\lor$
$(\alpha \lor (\beta \land \gamma))$	$\equiv$	$((\alpha \lor \beta) \land (\alpha \lor \gamma))$	distributivity of $\lor$ over $\land$

有效性与可满足性

一个语句是有效的 (valid)，如果它在所有模型中都为真

e.g., $True$, $A \vee \neg A$, $A \Rightarrow A$, $(A \wedge (A \Rightarrow B)) \Rightarrow B$

有效性通过演绎定理 (Deduction Theorem)与推理相关联：

$KB \models \alpha$ if and only if $(KB \Rightarrow \alpha)$ is valid

一个语句是可满足的 (satisfiable)，如果它在某些模型中为真

e.g., $A \vee B$, C

一个语句是不可满足的 (unsatisfiable)，如果没有模型能够使它为真

e.g., $A \wedge \neg A$

可满足性与推理是关联的：

$KB \models \alpha$ if and only if $(KB \wedge \neg\alpha)$ is unsatisfiable

证明(推理)方法

- 大体上分为两类
  - 应用推理规则
    - 用旧语句产生合理的新语句
    - Proof = a sequence of inference rule applications
      [Use inference rules as operators in a standard search algorithm]
    - 典型地，需要将语句转换为标准范式 (normal form)
  - 模型检验
    - 真值表的枚举 (总是 n 的指数级)
    - 改进的回溯法， e.g., Davis--Putnam-Logemann-Loveland (DPLL)
    - 模型空间的启发式搜索 (可靠的但可能不完备)
      e.g., min-conflicts-like hill-climbing algorithms

归结 (Resolution)

合取范式 (Conjunctive Normal Form, CNF)

conjunction of $\underbrace{\text{disjunctions of literals}}_{\text{子句 (clauses)}}$

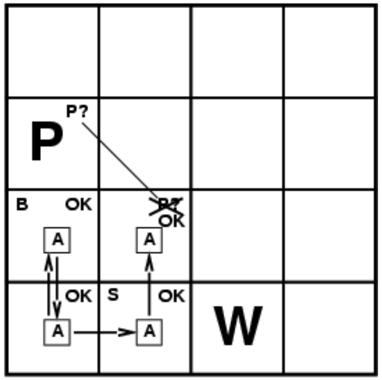
E.g., $(A \vee \neg B) \wedge (B \vee \neg C \vee \neg D)$

❖ 归结推理规则 (for CNF)：

$$\frac{\ell_1 \vee \cdots \vee \ell_k, \qquad m_1 \vee \cdots \vee m_n}{\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k \vee m_1 \vee \cdots \vee m_{j-1} \vee m_{j+1} \vee \cdots \vee m_n}$$

这里，ℓ_i 和 m_j 是互补常量，如：

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



❖ 对命题逻辑，归结是可靠的和完备的

转换为 CNF

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

1. 消去等价词 $\Leftrightarrow$, 用 $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$ 替换 $\alpha \Leftrightarrow \beta$

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

2. 消去蕴涵词 $\Rightarrow$, 用 $\neg\alpha \vee \beta$ 替换 $\alpha \Rightarrow \beta$

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

3. 将否定词 $\neg$ 移到文字前面 (应用 Morgan‘s 和 double-negation律)

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

4. 使用分配律 (distributivity) 并扁平化

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

归结算法

❖ 采用反证法：证明 $KB \wedge \neg\alpha$ 是不可满足的

```
function PL-RESOLUTION(KB, α) returns true or false
    inputs: KB, the knowledge base, a sentence in propositional logic
            α, the query, a sentence in propositional logic

    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
```

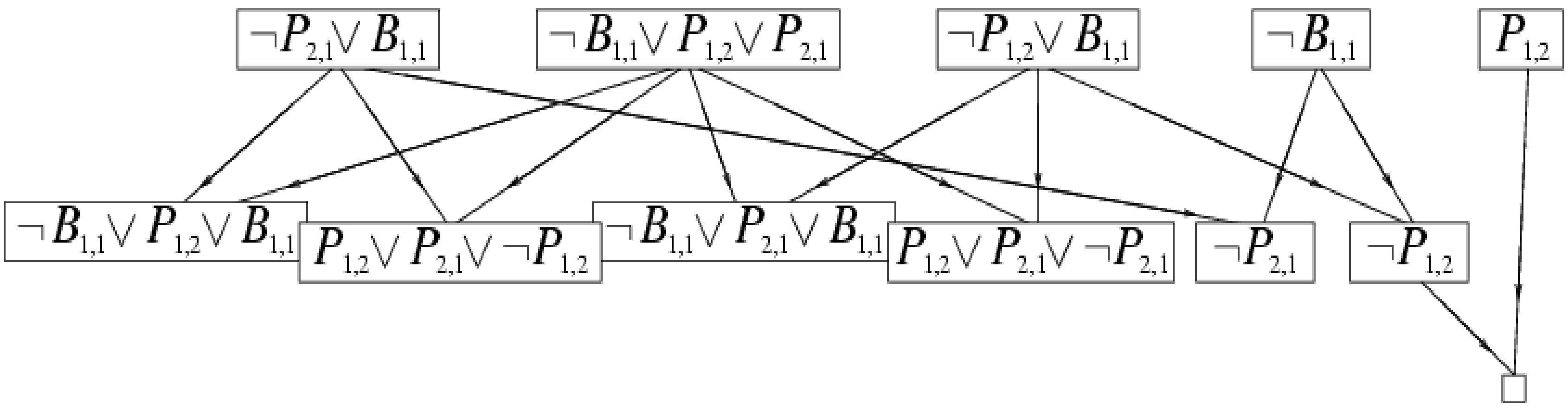
穷尽遍历

发生矛盾

不再产生新子句

归结推理举例

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



前向和反向链接

- Horn 形式 (restricted)

  KB = conjunction of Horn clauses（Horn子句的合取）

  Horn clause =
  - 命题符号；或者
  - 蕴涵式：(conjunction of symbols) ⇒ symbol

  - E.g., $C \wedge (B \Rightarrow A) \wedge (C \wedge D \Rightarrow B)$

- Horn形式的假言推理 (Modus Ponens)：对Horn知识库是完备的

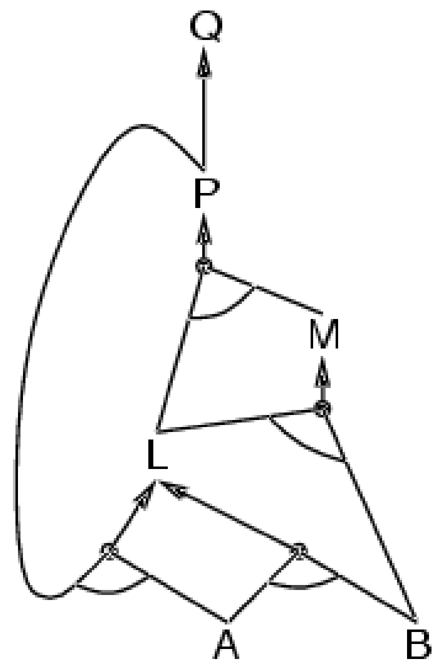
$$\frac{\alpha_1, \ldots, \alpha_n, \qquad \alpha_1 \wedge \cdots \wedge \alpha_n \Rightarrow \beta}{\beta}$$

- 可以使用前向链接(forward chaining)或反向链接(backward chaining)算法
- 这些算法是自然的，时间复杂性与知识库大小成线性关系

前向链接

- 基本思想：触发 *KB* 中前提得到满足的所有规则
  - 将结论添加到 *KB* 中，直到 查询(query) 能够得到

$$
\begin{array}{l}
P \Rightarrow Q \\
L \wedge M \Rightarrow P \\
B \wedge L \Rightarrow M \\
A \wedge P \Rightarrow L \\
A \wedge B \Rightarrow L \\
\hline
A \\
B
\end{array}
$$

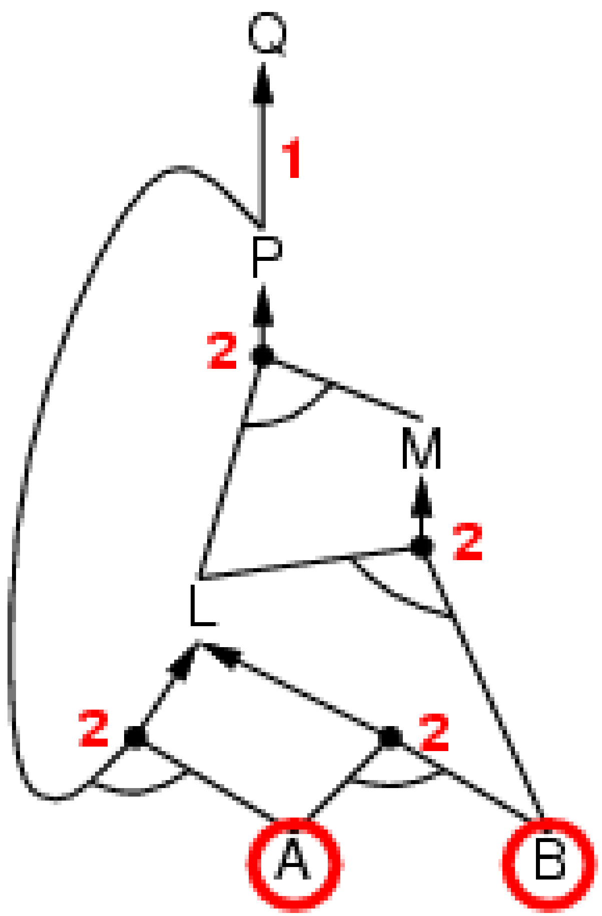


前向链接算法

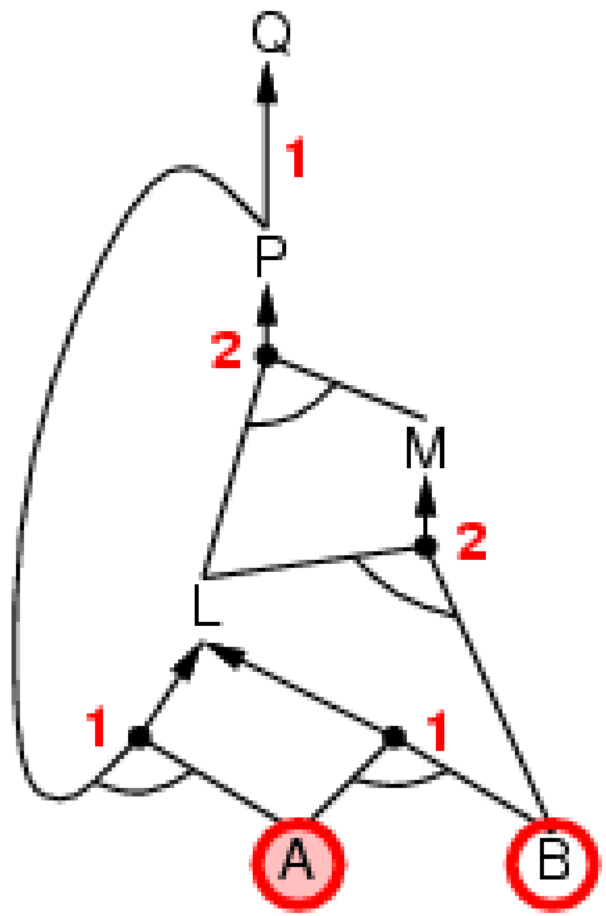
```
function PL-FC-ENTAILS?(KB, q) returns true or false
   inputs: KB, the knowledge base, a set of propositional Horn clauses
           q, the query, a proposition symbol
   local variables: count, a table, indexed by clause, initially the number of premises
                    inferred, a table, indexed by symbol, each entry initially false
                    agenda, a list of symbols, initially the symbols known in KB

   while agenda is not empty do                         穷尽执行
       p ← POP(agenda)
       unless inferred[p] do
           inferred[p] ← true
           for each Horn clause c in whose premise p appears do
               decrement count[c]
               if count[c] = 0 then do                  蕴含条件满足
                   if HEAD[c] = q then return true
                   PUSH(HEAD[c], agenda)                得到查询语句
   return false
```

- 对 Horn 知识库，前向链接算法是可靠的和完备的

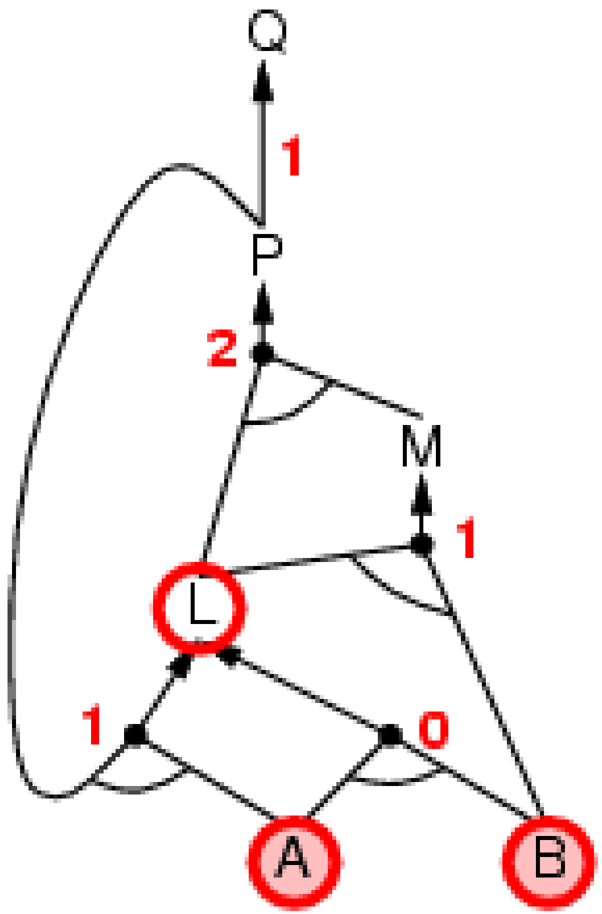
前向链接举例



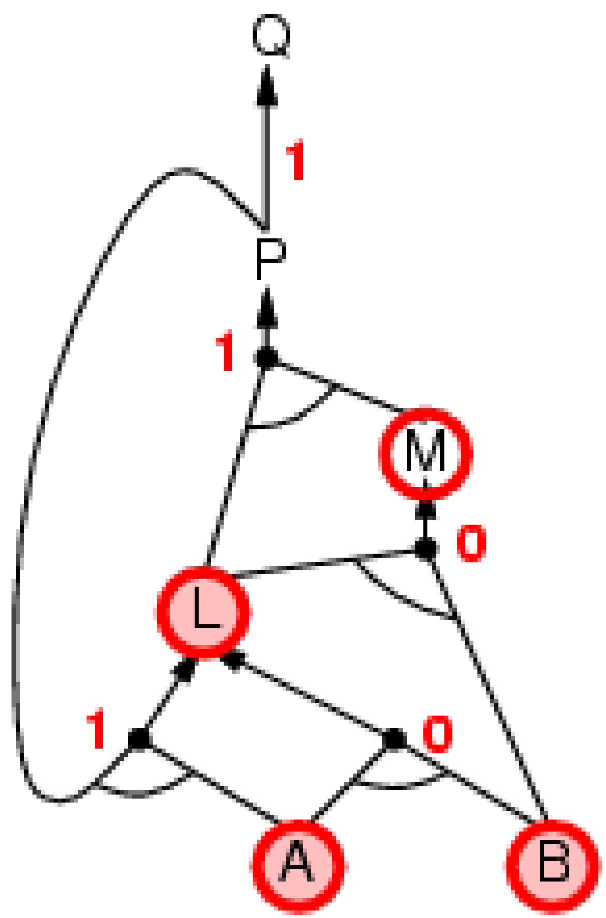
前向链接举例



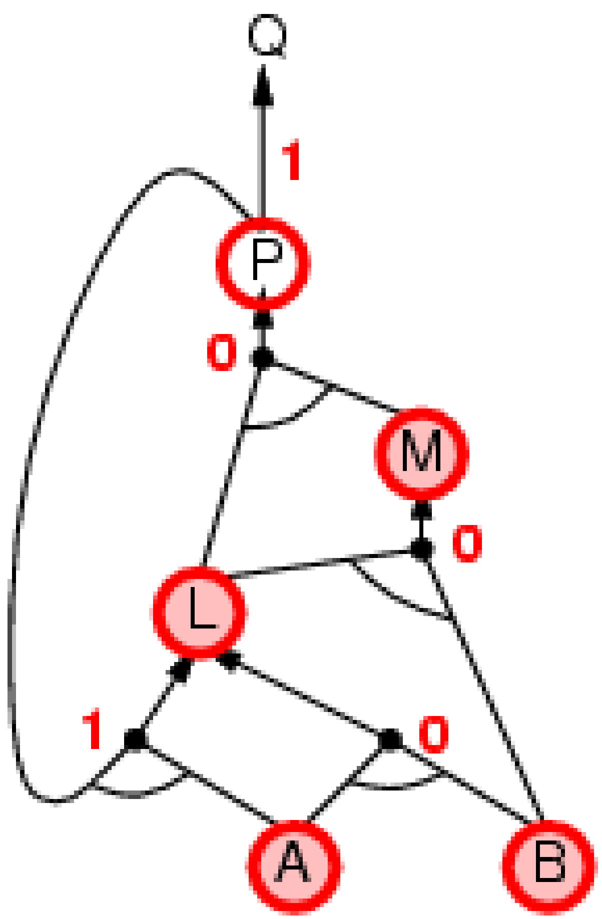
前向链接举例



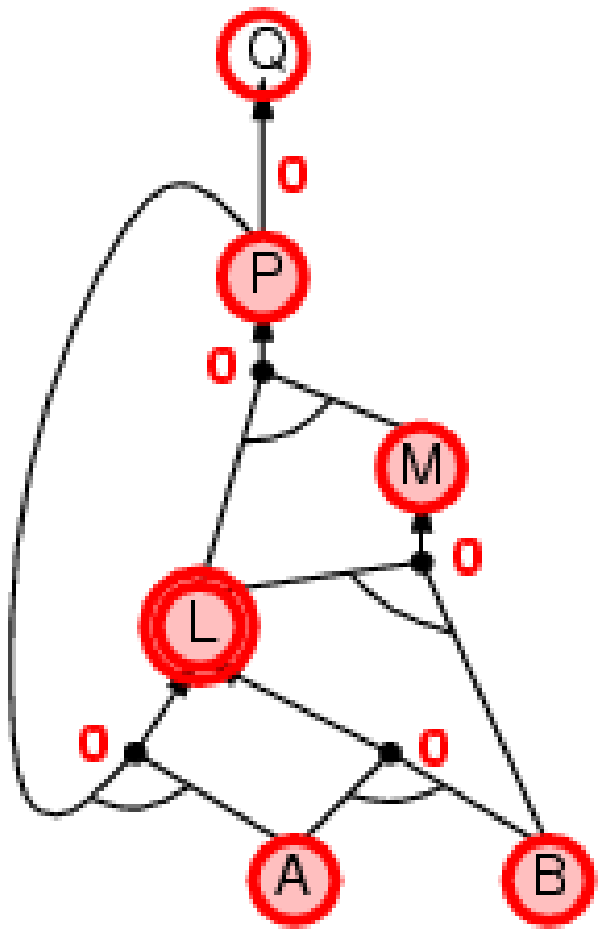
前向链接举例



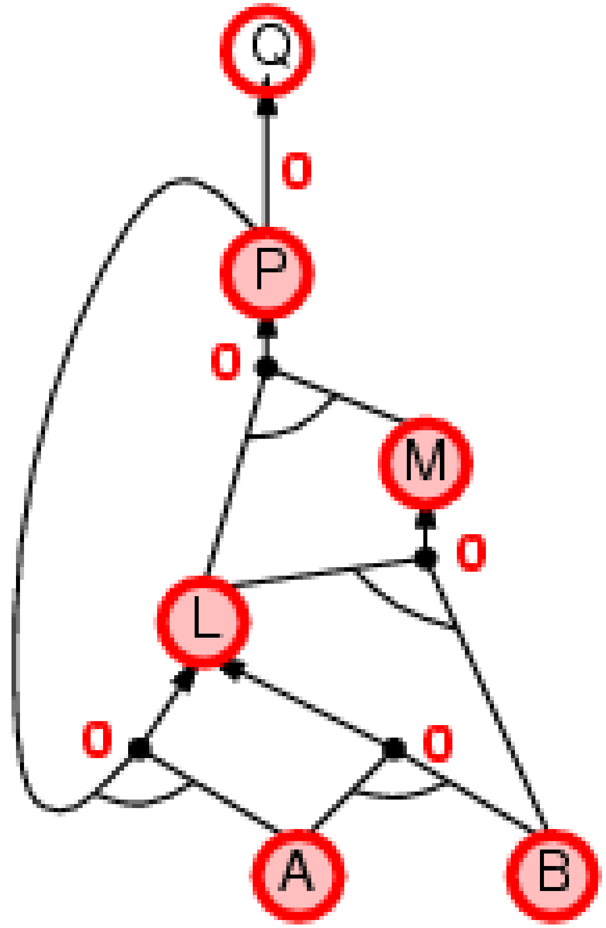
前向链接举例



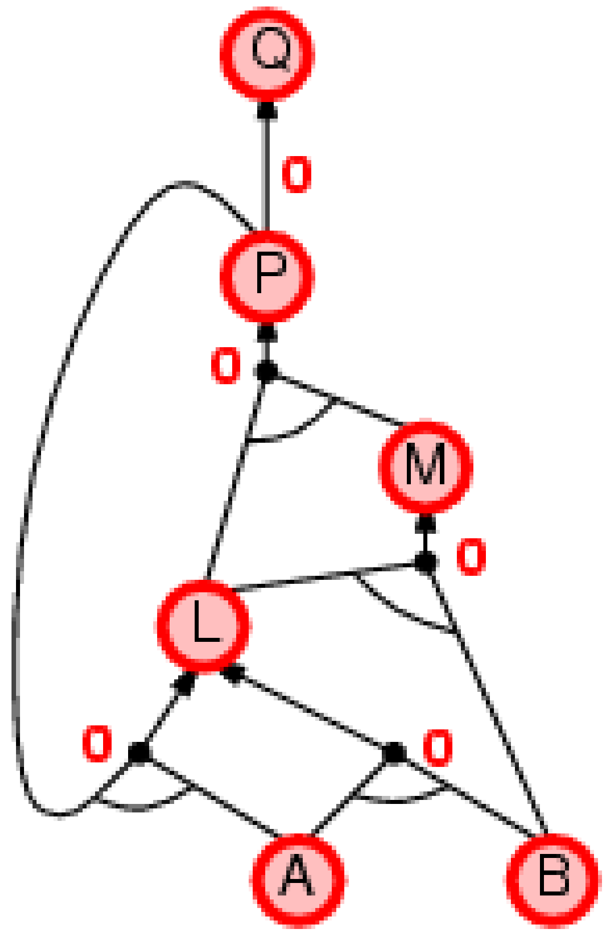
前向链接举例



前向链接举例



前向链接举例



完备性证明

❖ 前向链接能够推导出知识库 *KB* 蕴涵的任一原子语句

1. FC 达到一个稳定点 (fixed point)——没有新的原子语句
2. 考虑最终状态的模型 *m*，每个符号都赋值了 *true/false*
3. 原知识库 *KB* 中的每个子句在 *m* 中都是 *true*

Proof: Suppose a clause $a_1 \wedge \ldots \wedge a_k \Rightarrow b$ is false in m
Then $a_1 \wedge \ldots \wedge a_k$ is true in m and b is false in m
Therefore the algorithm has not reached a fixed point!

4. 因而，*m* 是 *KB* 的一个模型
5. 如果 $KB \models q$, q 在 *KB* 的每个模型中都为真，包括 *m*

反向链接

基本思想：从查询 q 开始进行后向处理

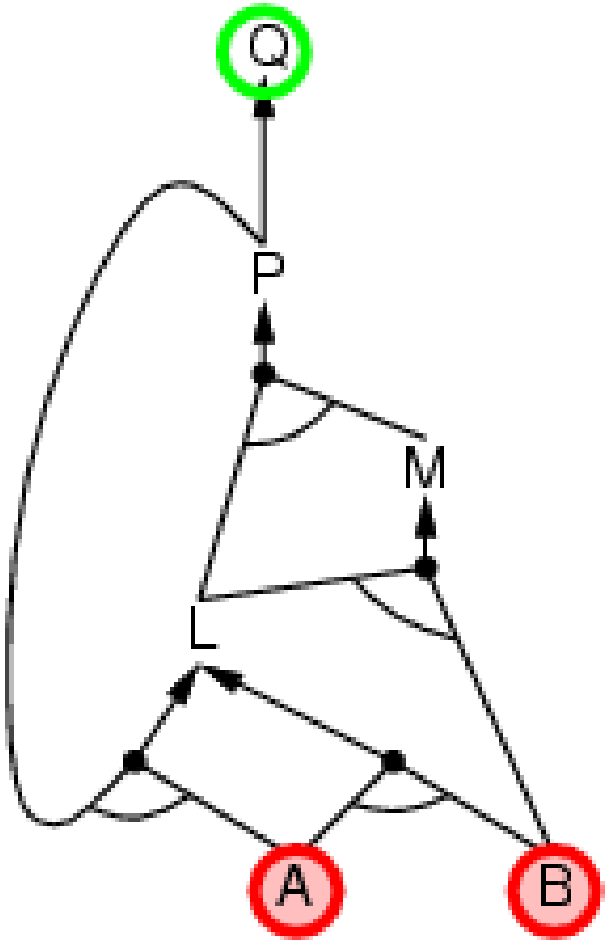
> 用反向链接 (BC) 证明 q ：
>
> 检查 q 是否已知了，或者
>
> 用 BC 证明 q 的所有前提 (递归新的子目标）

避免循环：检查新的子目标是否已经在目标堆栈中

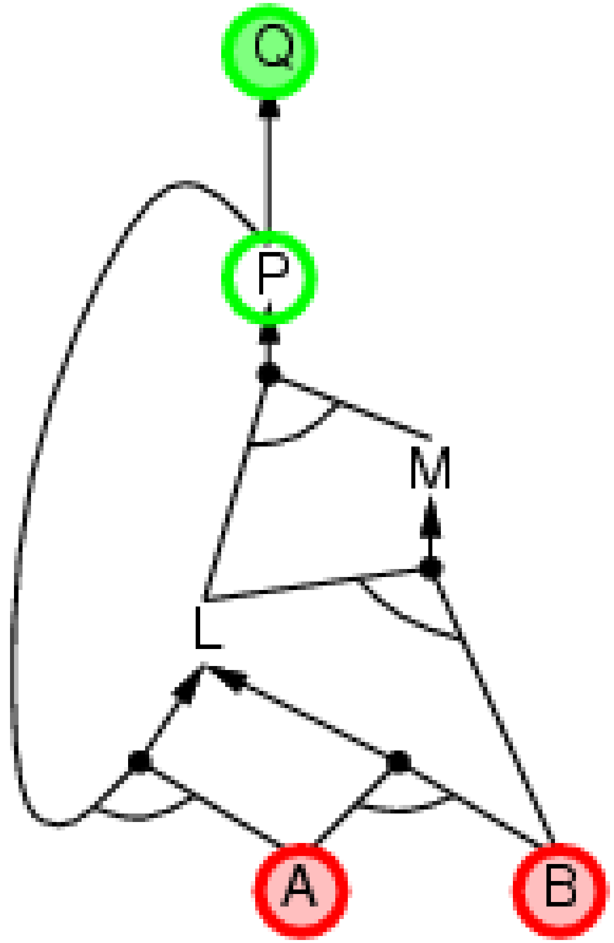
避免重复工作：检查新的子目标是否

1. 已经被证明，或者
2. 已经失败了

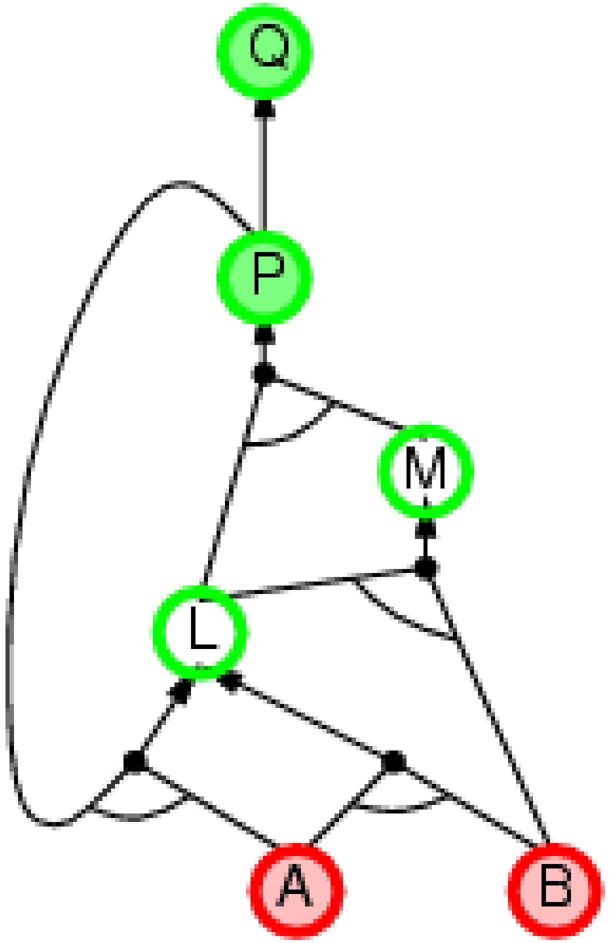
反向链接举例



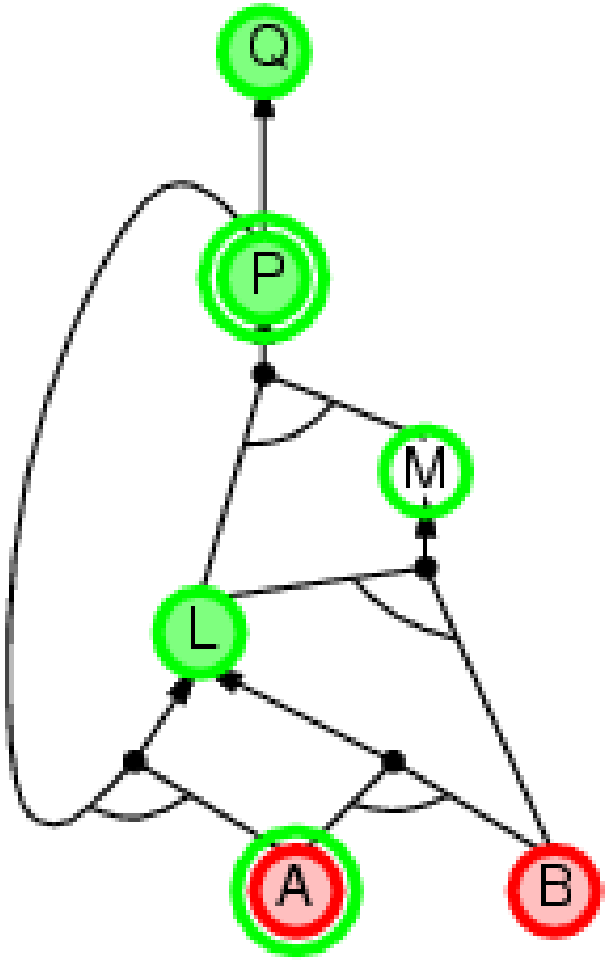
反向链接举例



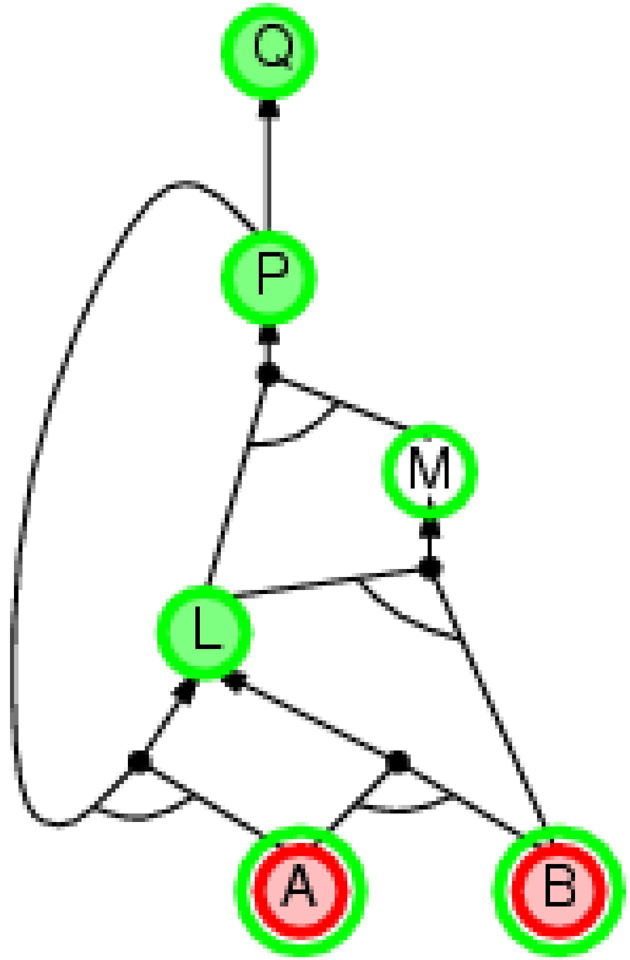
反向链接举例



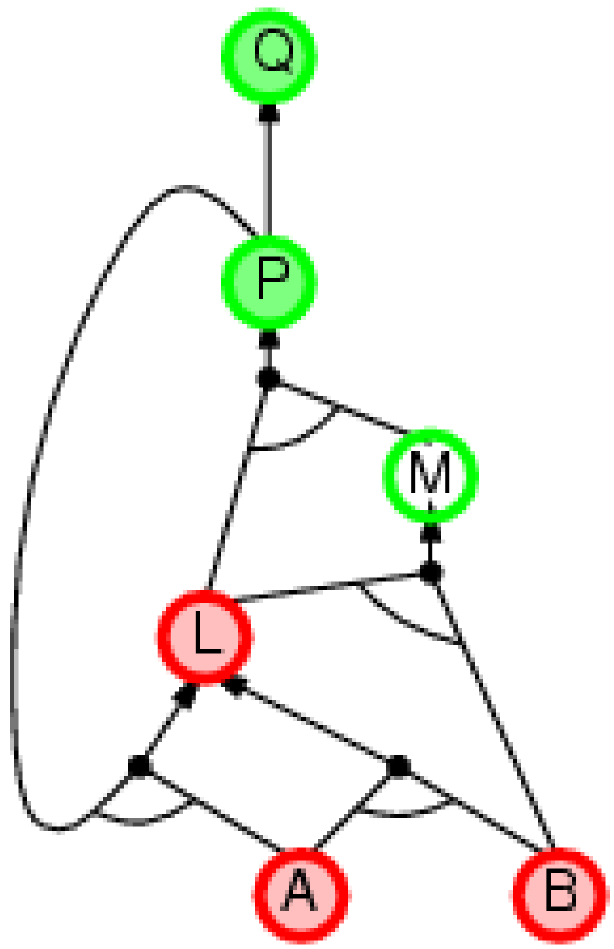
反向链接举例



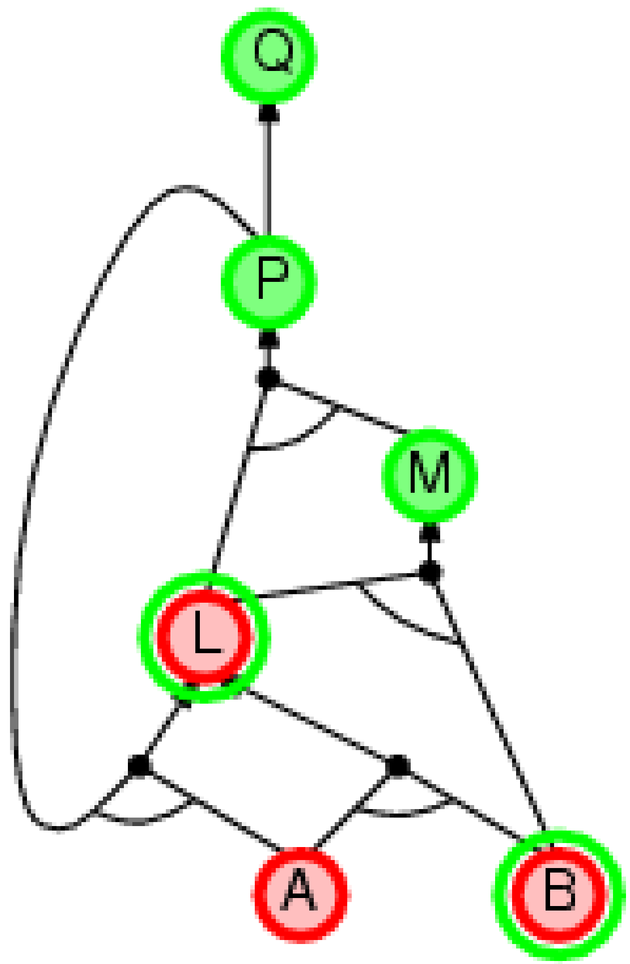
反向链接举例



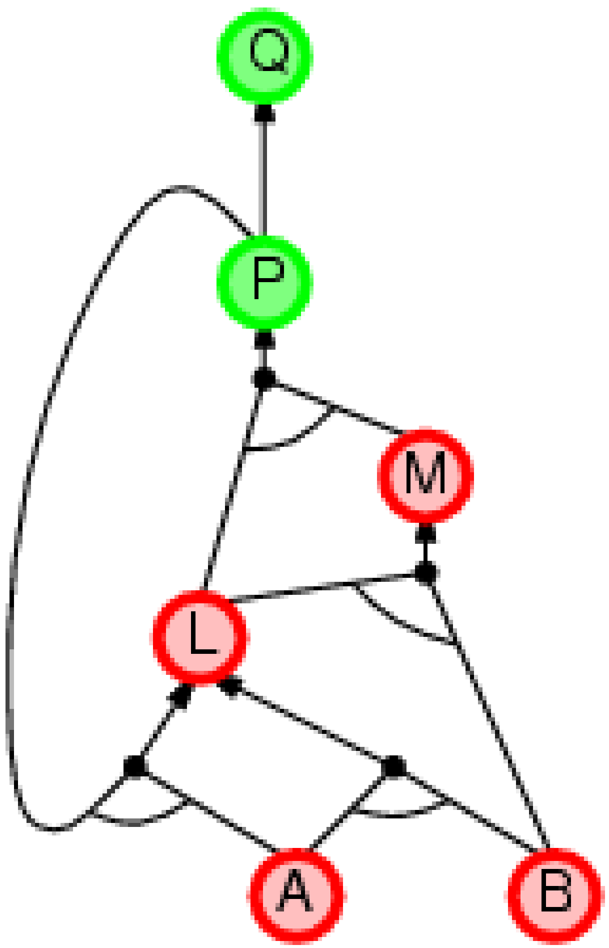
反向链接举例



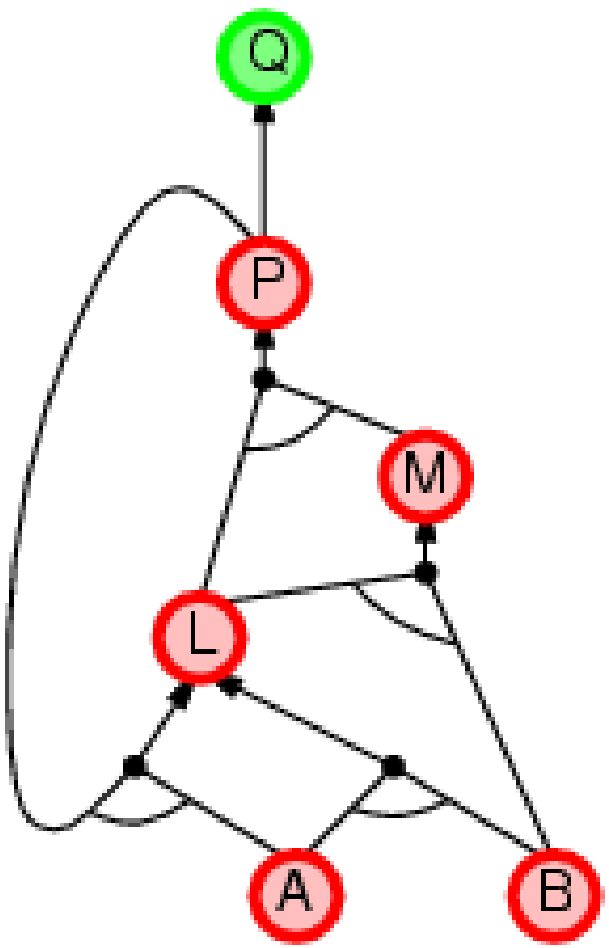
反向链接举例



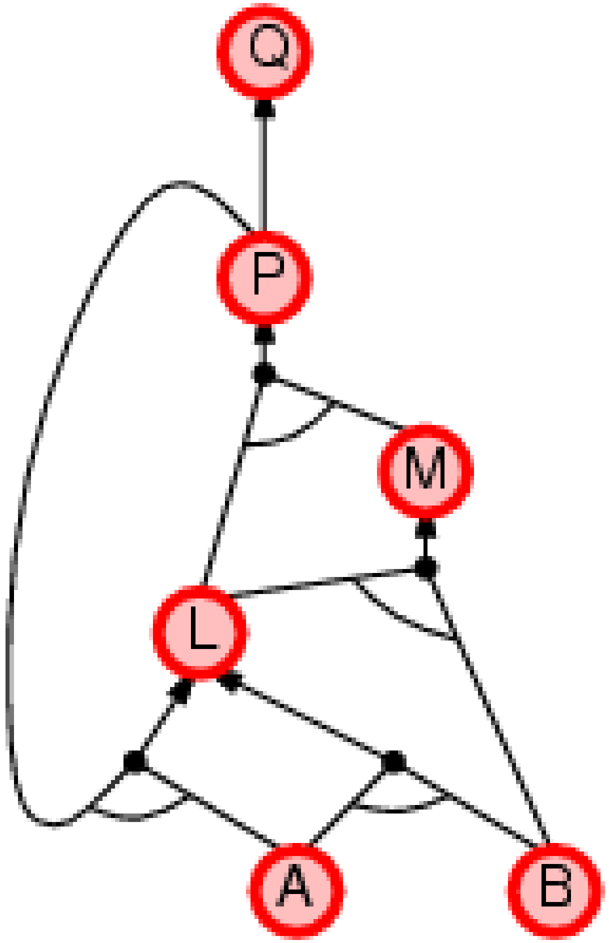
反向链接举例



反向链接举例



反向链接举例



前向与反向链接对比

- 前向链接 (FC) 是数据驱动的，自动无意识地处理
  - e.g., object recognition, routine decisions
  - 可能做很多与目标无关的工作

- 反向链接 (BC) 是目标驱动的，适合于问题求解
  - e.g., Where are my keys? How do I get into a PhD program?
  - BC 的时间复杂性比数据库大小的线性还要低

总结

- 逻辑 agents 在知识库 *KB* 上进行推理，推导出新信息或作出决策
- 逻辑的基本概念
  - 语法 (syntax): formal structure of sentences
  - 语义 (semantics): truth of sentences wrt models
  - 蕴涵 (entailment): necessary truth of one sentence given another
  - 推理 (inference): deriving sentences from other sentences
  - 可靠性 (soundness): derivations produce only entailed sentences
  - 完备性 (completeness): derivations can produce all entailed sentences
- Wumpus 世界要求能够表达部分或否定信息、实例推理等
- 命题逻辑的归结推理是完备的
- 对Horn子句，前向和反向链接是线性时间的和完备的
- 命题逻辑表达能力不足

谢谢聆听！