人工智能

中国科学院计算技术研究所
Institute Of Computing Technology Chinese Academy Of Sciences

罗平 luop@ict.ac.cn
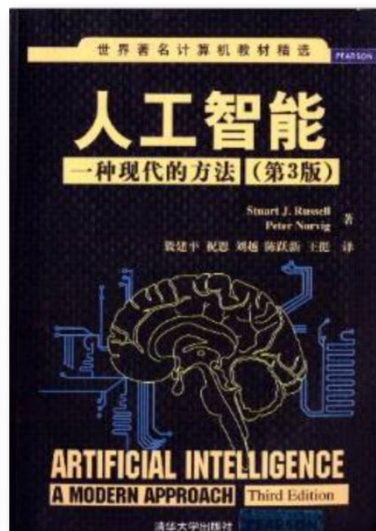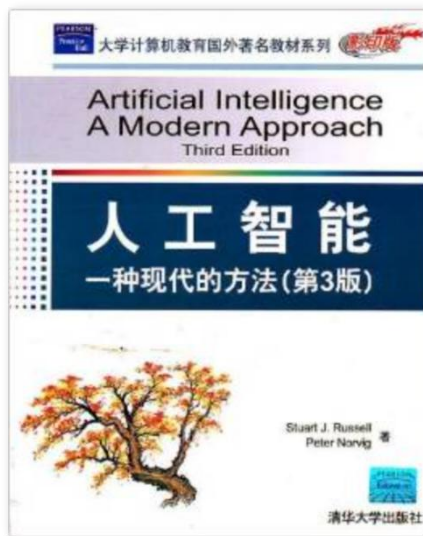
Knowledge 1

教材

http://www.ai-fundamental.com/
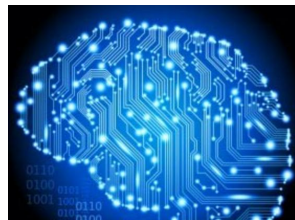
Materials on Knowledge

课程第二部分：
台湾大学 于天立教授 "人工智慧" 课程

链接: http://pan.baidu.com/s/1miec9jM 密码: 4fd4

http://aima.cs.berkeley.edu/

中英文版本对照阅读

数理逻辑

- 《计算机科学的数理逻辑》
  陆钟万 著

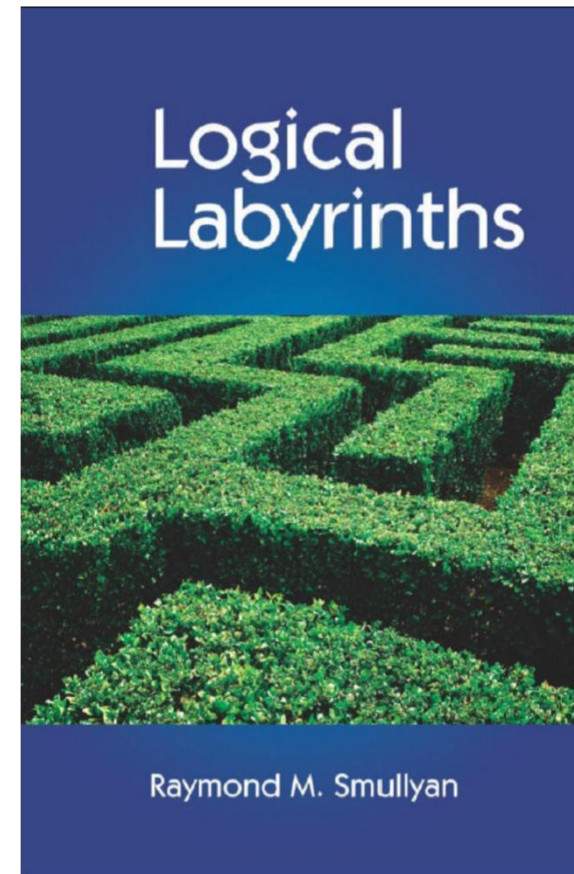- 授课视频：
  http://www.1ketang.com/course/2025.html

什么是 "逻辑"

One important purpose of mathematical logic is to make precise the notion of a *proof*.
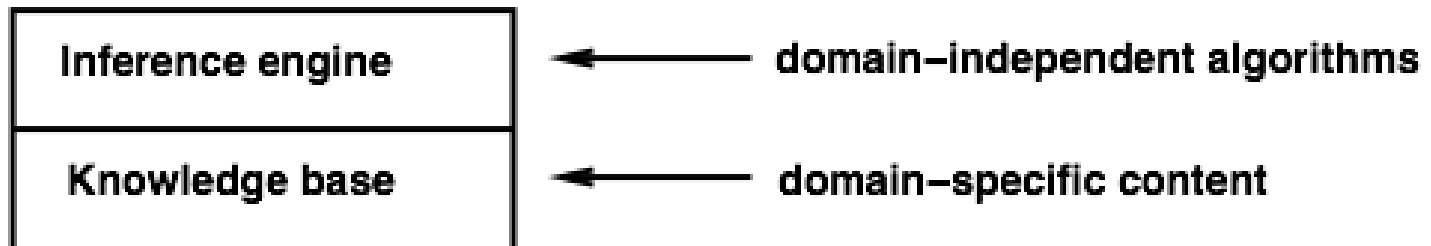
Just what is meant by the word "proof"? One person suggested to me that a proof is simply an **argument** that convinces somebody. Not bad, but the trouble is that different people are convinced by different arguments, so the concept of *proof* in that sense is quite subjective.

Can't one, then, find a more *objective* definition of the word "proof"? Well, so far no such definition has been found for the *general* notion of proof, but symbolic logic has done an admirable job in making absolutely precise the notion of "proof" for given axiom systems!

- Raymond Smullyan

Knowledge bases

```
┌─────────────────────────┐
│   Inference engine      │ ◄────── domain−independent algorithms
├─────────────────────────┤
│   Knowledge base        │ ◄────── domain−specific content
└─────────────────────────┘
```

- **Knowledge base** =set of **sentences** in a **formal** language
- Declarative approach to building an agent(or other system):
          TELL it what it needs to know
- Then it can ASK itself what to do—answers should follow from the KB

推理变成一个"机械"的自动过程。

Logic in general

- **Logics** are formal languages for representing information such that conclusions can be drawn

- **Syntax** defines the sentences in the language

- **Semantics** define the "meaning" of sentences;

  i.e., define truth of a sentence in a world

- E.g., the language of arithmetic

- $x + 2 \geq y$ 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 $x = 7, y = 1$

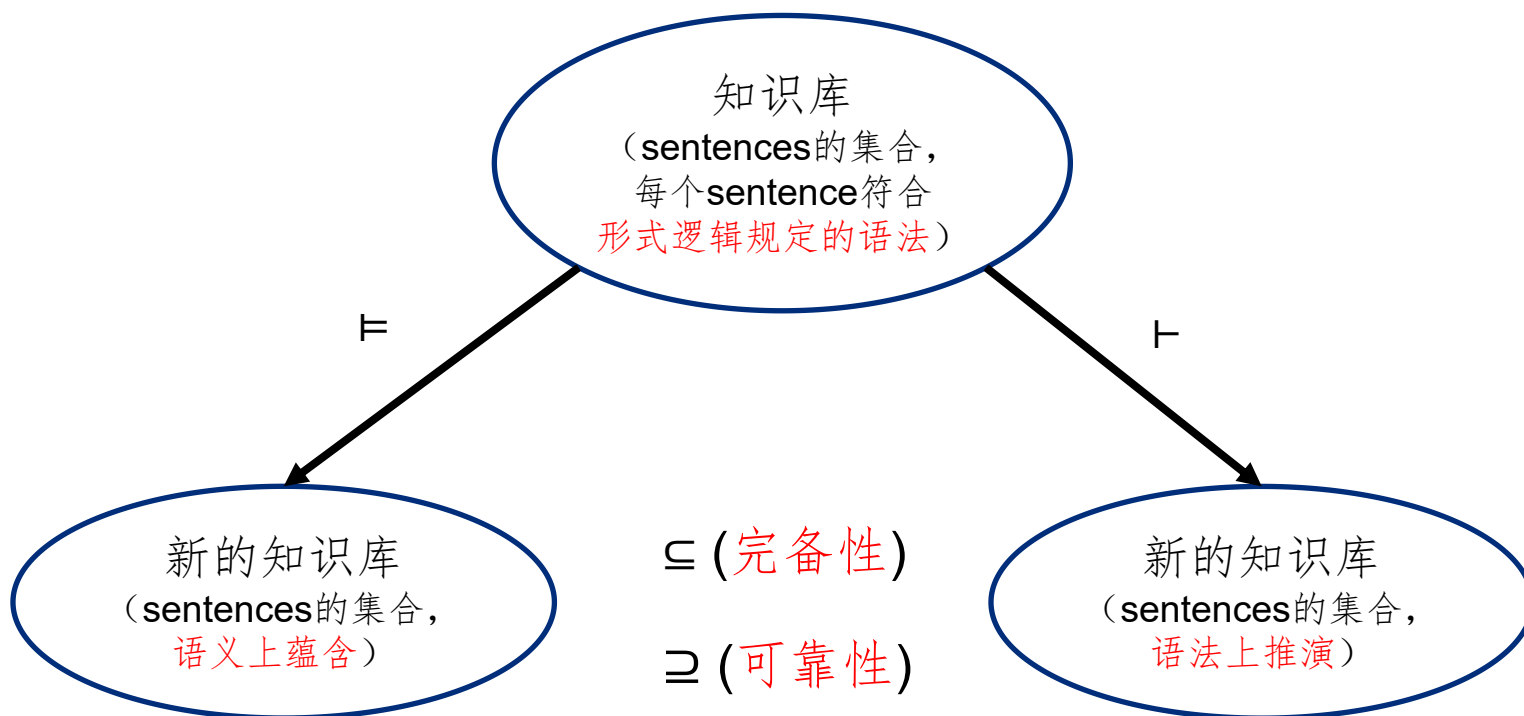- $x + 2 \geq y$ is false in a world where $x = 0, y = 6$

逻辑研究的内容

- 研究形式化定义的**sentences**之间的关系

- 两个角度：
  语义：**entailment** 蕴含，逻辑推导
  语法：**inference** 演绎，形式推演

知识库
（sentences的集合，
每个sentence符合
形式逻辑规定的语法）

$\vDash$

$\vdash$

新的知识库
（sentences的集合，
语义上蕴含）

新的知识库
（sentences的集合，
语法上推演）

$\subseteq$（完备性）

$\supseteq$（可靠性）

语义 ENTAILMENT （一般的定义）
什么是"正确的推理"？

Model（模型），Truth Assignment（真值指派）

- Logicians typically think in terms of **models**, which are formally structured worlds respect to which can be evaluated

- We say m is a model of a sentence α if α is true in m

- $M(\alpha)$ is the set of all models of α

X+Y=4
X=0,Y=4是这个句子的model

Entailment（蕴涵 / 蕴含）

语义：逻辑推导

- Entailment means that one thing follows from another：

    $$KB \vDash \alpha$$

- Knowledge base *KB* entails sentence α

    if and only if

- α is true in all worlds where *KB* is true

- 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 $4 = x + y$

- Entailment is a relationship between sentences (i.e., syntax) that is based on semantics

- Note: brains process syntax (of some sort)

Model（模型）

- Logicians typically think in terms of **models**, which are formally structured worlds respect to which can be evaluated

- We say m is a model of a sentence α if α is true in m

- $M(\alpha)$ is the set of all models of α

- Then $KB \vDash \alpha$ if and only if $M(KB) \subseteq M(\alpha)$

- E.g. *KB* = Giants won and Reds won

  α = Giants won

Formal proof
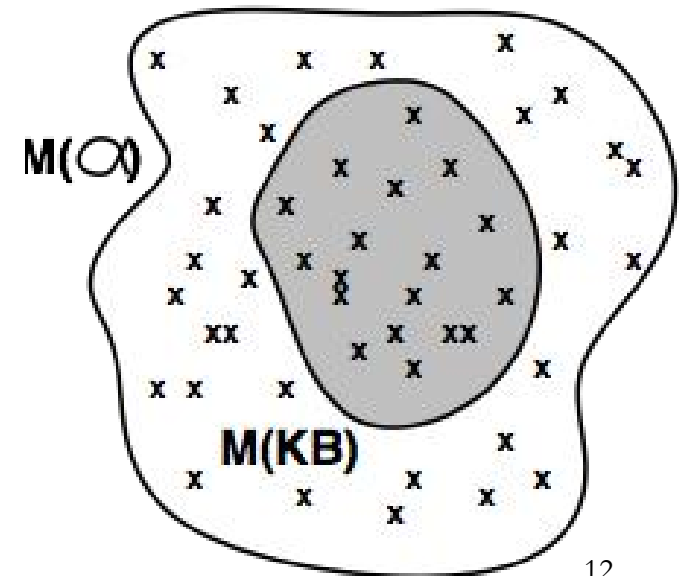
Wumpus World PEAS description

- **Performance measure**
  - Gold +1000, death -1000
  - -1 per step, -10 for using the arrow
- **Environment**
  - Squares adjacent to Wumpus are smelly
  - Squares adjacent to are breezy
  - Glitter iff gold is in the same square
  - Shooting kills Wumpus if you are facing it
  - Shooting uses up the only arrow
  - Grabbing picks up gold if in same square
  - Releasing drops the gold in same square
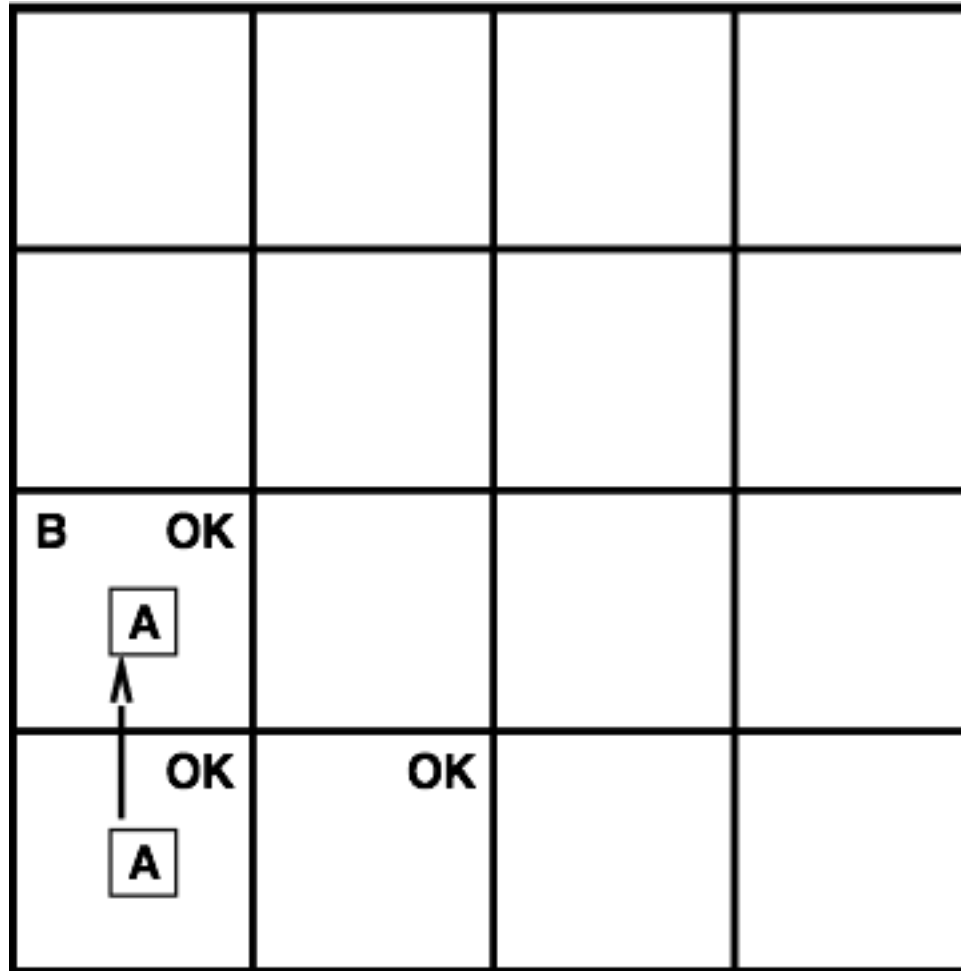- **Sensors**   Breeze, Glitter, Smell
- **Actuators**
  - Forward, Grab, Release, Shoot, Left turn, Right turn
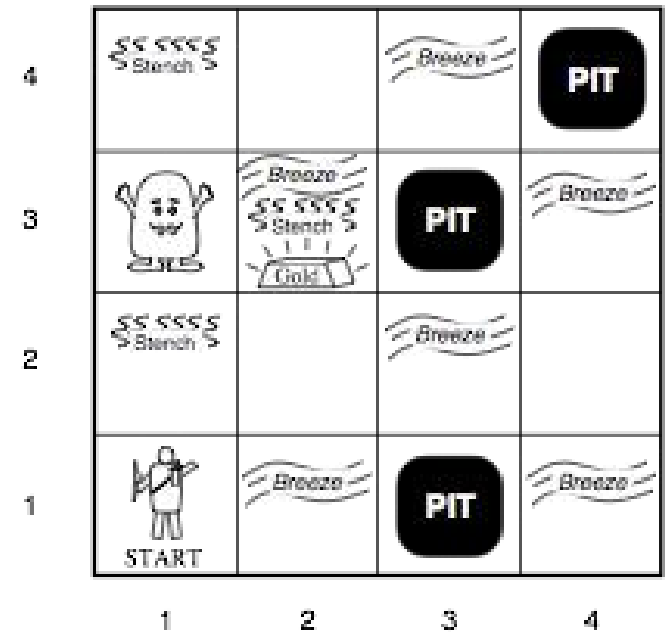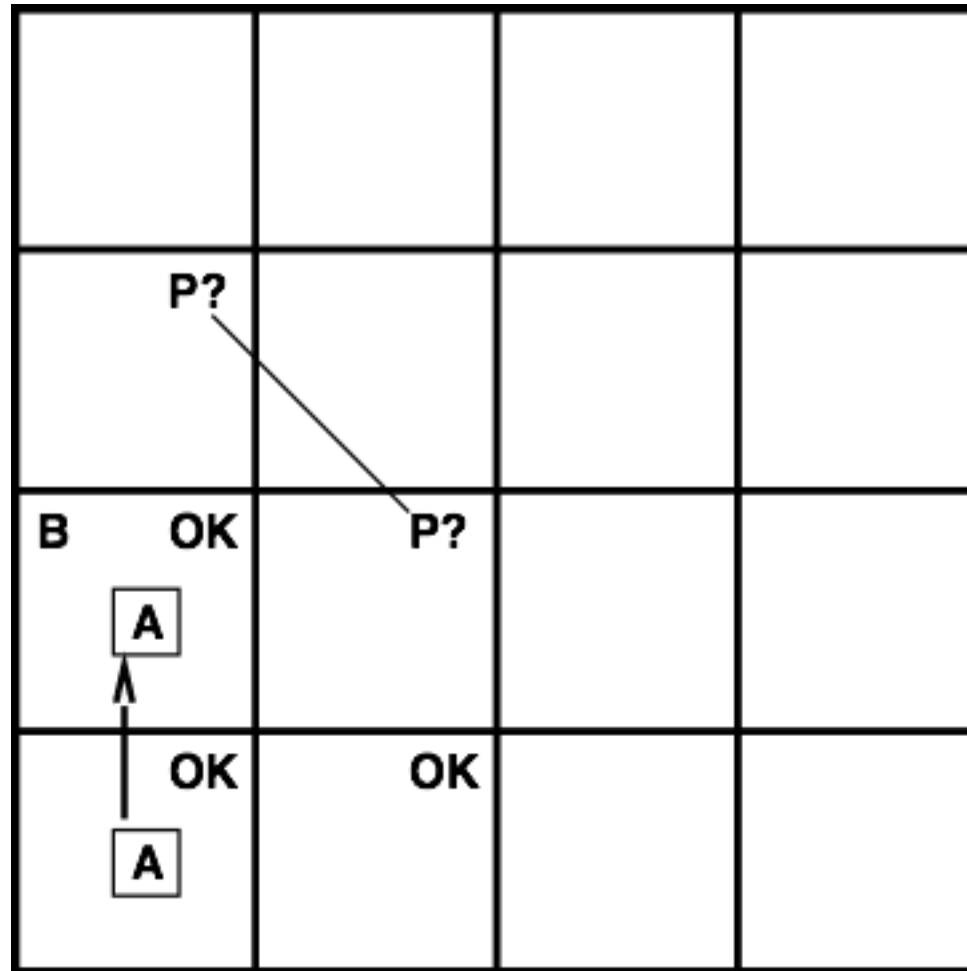
Exploring a wumpus world

Exploring a wumpus world
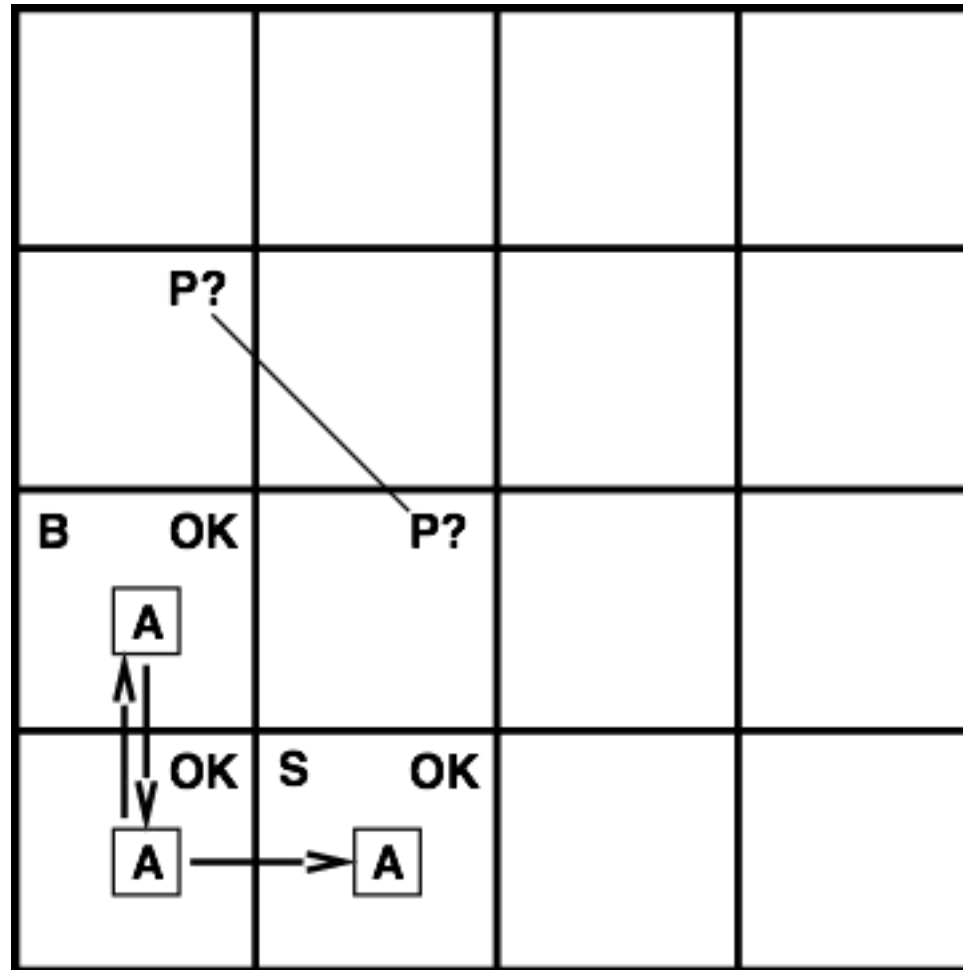
Exploring a wumpus world

Exploring a wumpus world

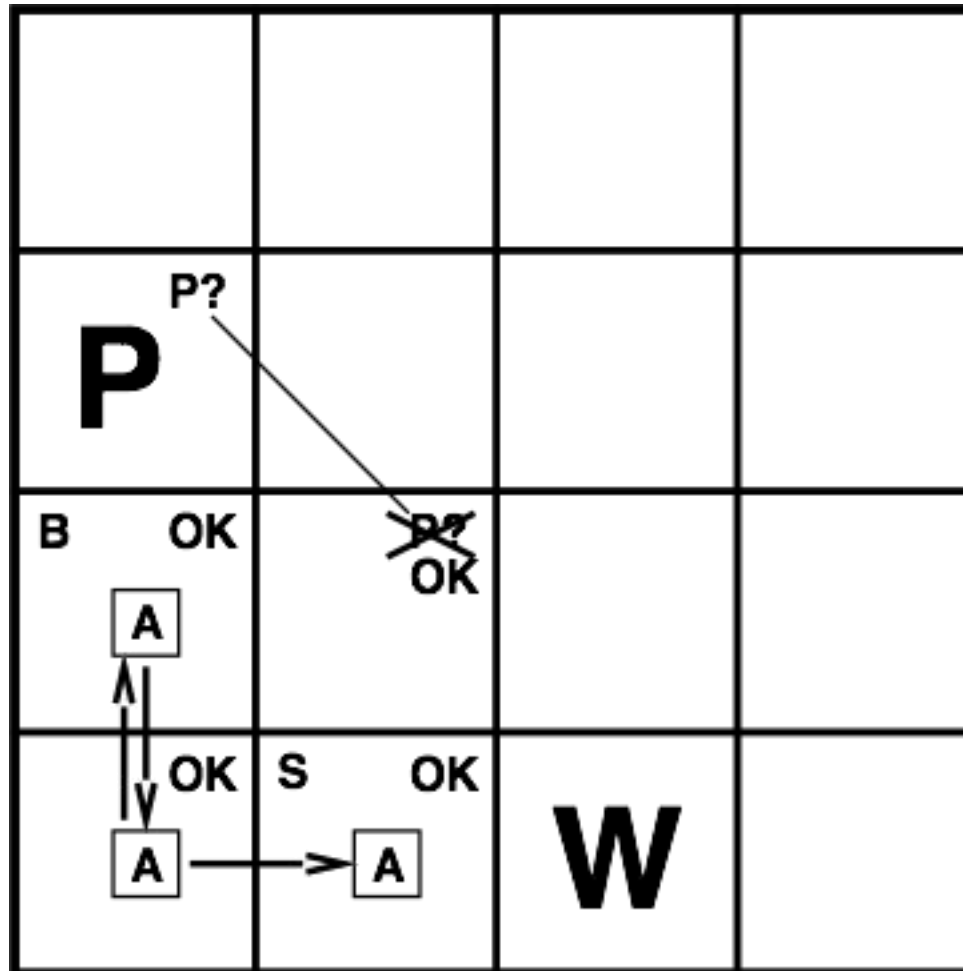# Exploring a wumpus world

Exploring a wumpus world
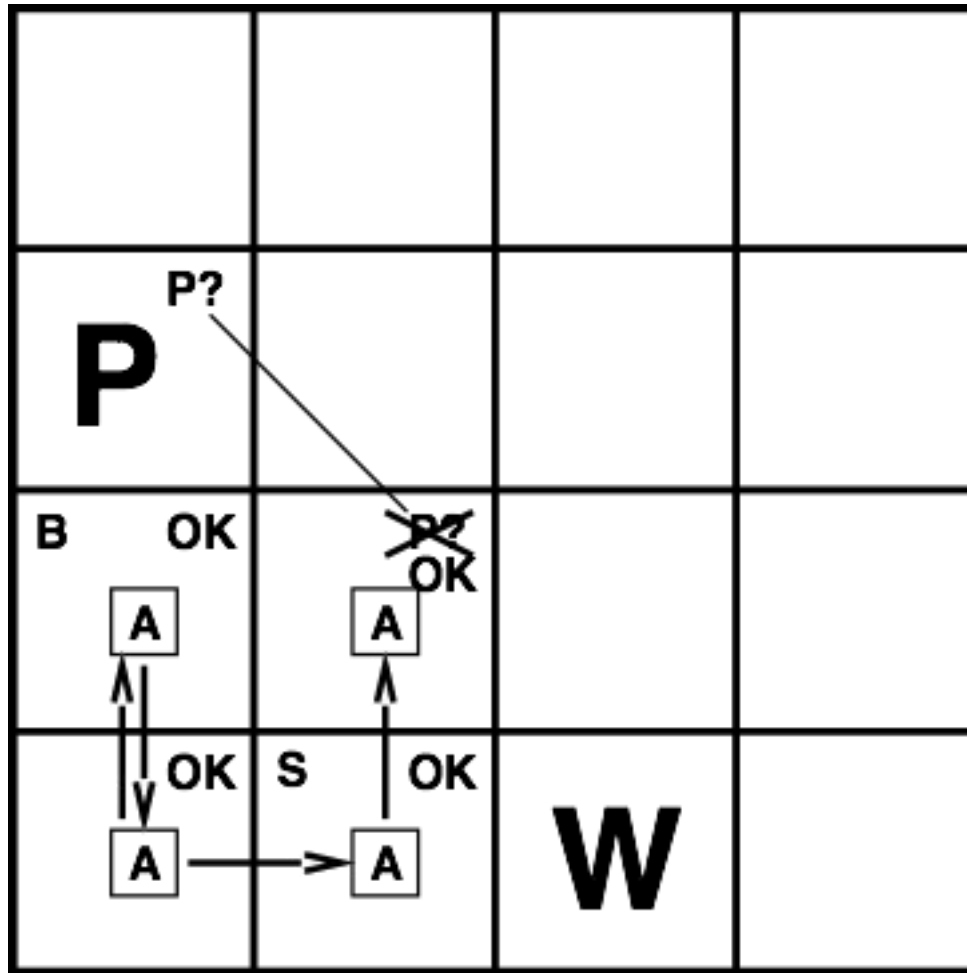
Exploring a wumpus world
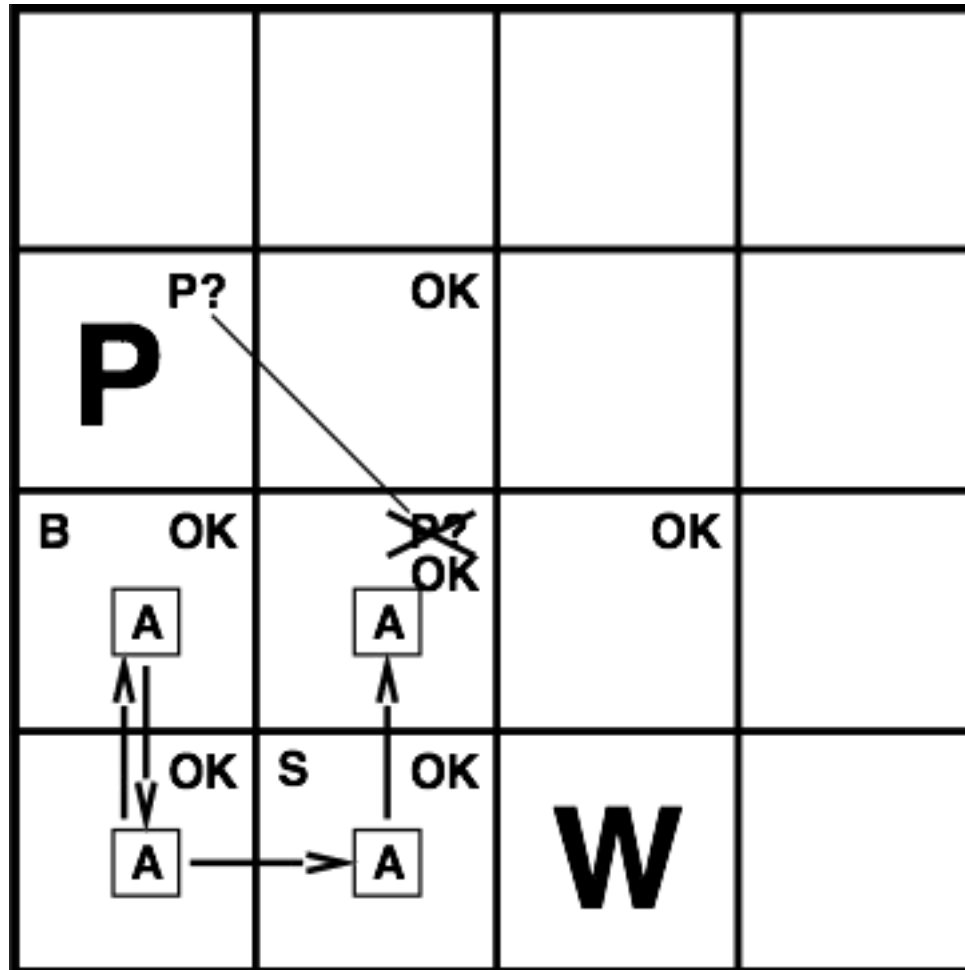
Exploring a wumpus world
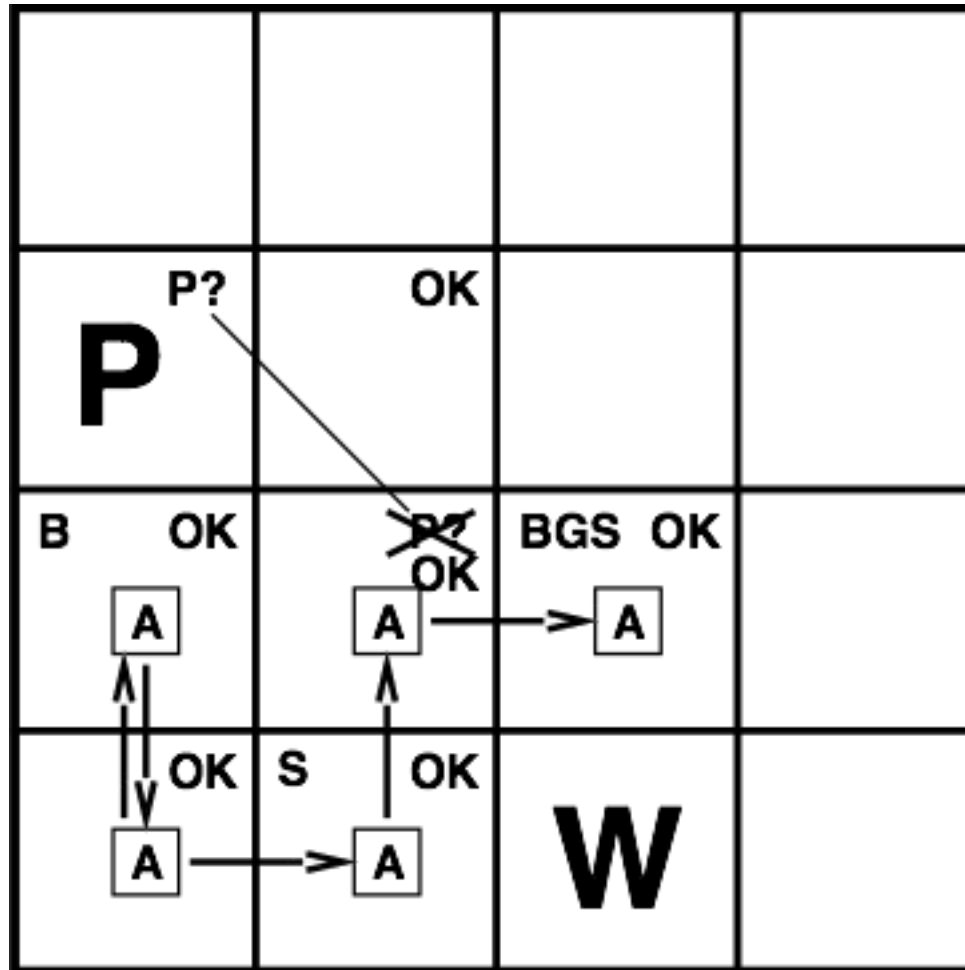
Entailment in the wumpus world

- Situation after detecting nothing in [1,1], moving right, breeze in [2,1]

- Consider possible models for ?s assuming only pits

- 3 Boolean choices ⟹ 8 possible models
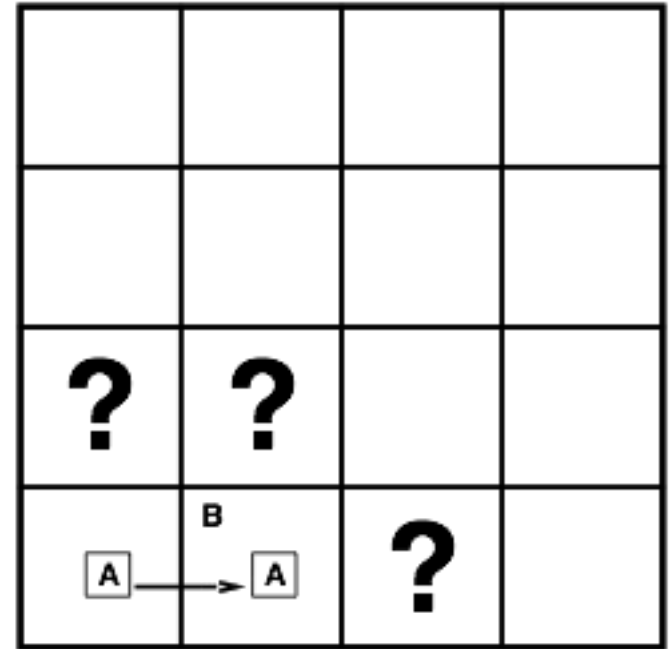
Wumpus models
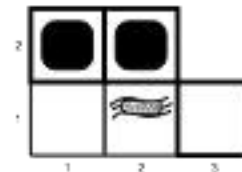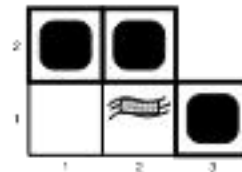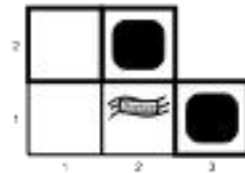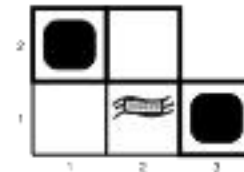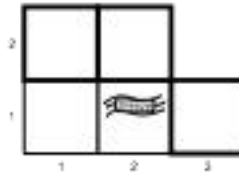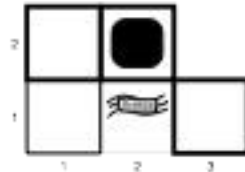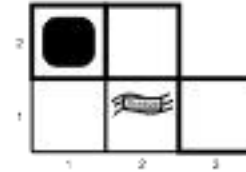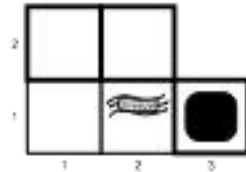
Wumpus models
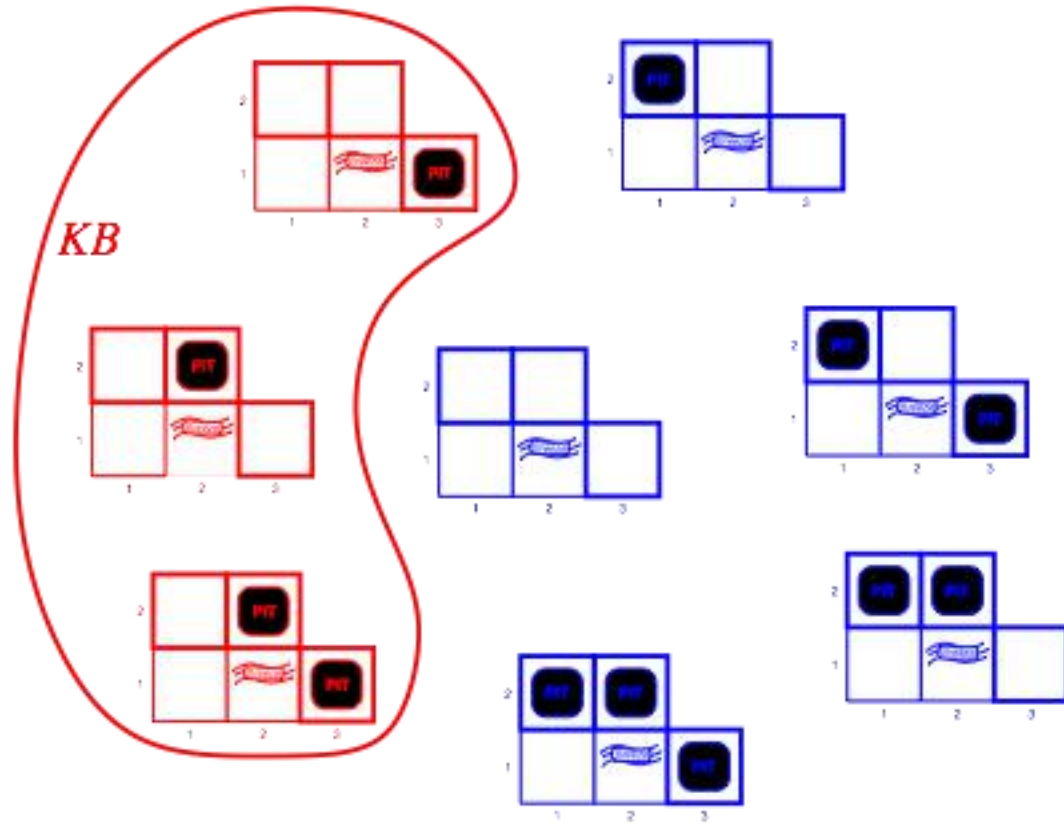
KB = Wumpus-world rules + observations （[2,1]有风）

Wumpus models
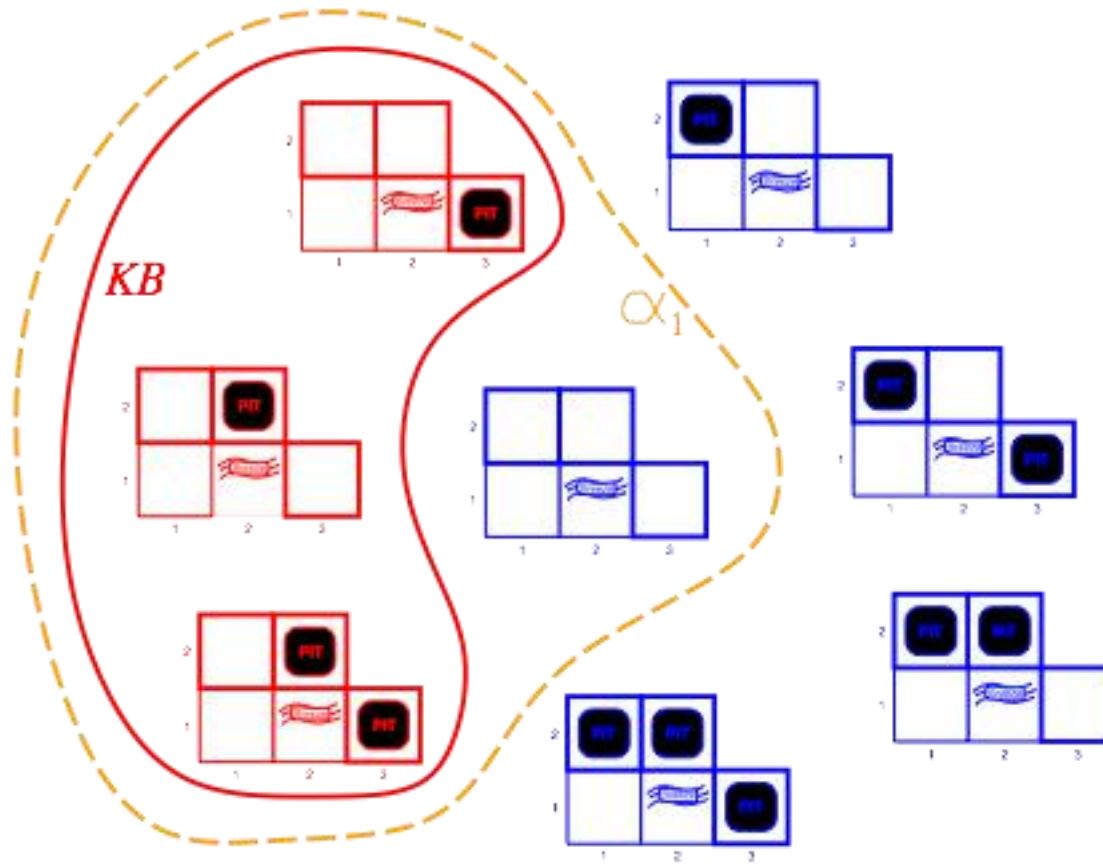
KB = Wumpus-world rules + observations

α_1 = "[1,2] is not pit", $KB \vDash \alpha_1$, proved by model checking

25

Wumpus models
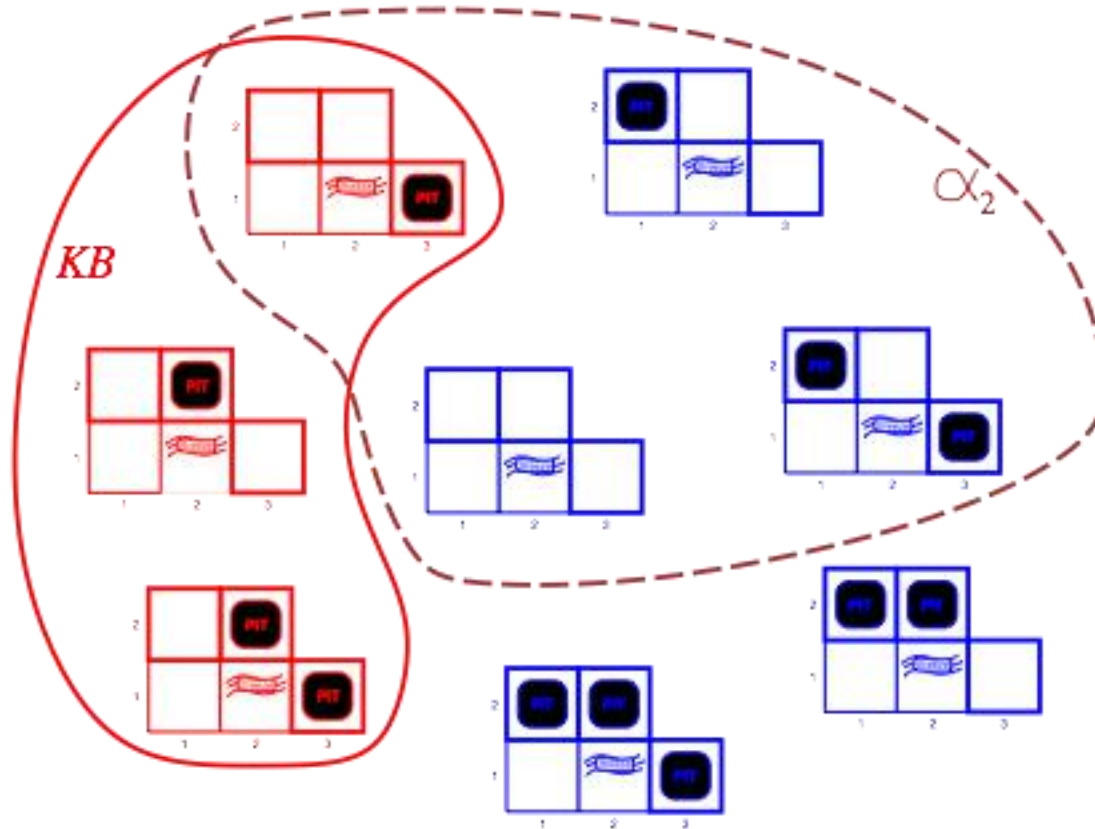
KB = Wumpus-world rules + observations

α_2 = "[2,2] is not pit", $KB \not\models \alpha_2$

PROPOSITIONAL LOGIC: SYNTAX AND SEMANTICS

Propositional logic（命题逻辑）：Syntax

- **Proposition:** a declarative sentence that is either true or false

- Propositional logic usually does not consider time

- If the truth of a proposition varies over time, we call it **fluent** ("today is Monday")

- **Atomic propositions (原子命题)** are minimal propositions

- **Literals (文字)** are atomic propositions or their negations

Propositional logic (命题逻辑): Syntax

■ Defined in Backus-Naur form (BNF)

$$
\begin{aligned}
Sentence \;\;&\rightarrow\;\; AtomicSentence \mid ComplexSentence \\
AtomicSentence \;\;&\rightarrow\;\; True \mid False \mid P \mid Q \mid R \ldots \\
ComplexSentence \;\;&\rightarrow\;\; (\text{Sentence}) \mid [Sentence] \\
&\mid\quad \neg Sentence \\
&\mid\quad Sentence \wedge Sentence \\
&\mid\quad Sentence \vee Sentence \\
&\mid\quad Sentence \Rightarrow Sentence \\
&\mid\quad Sentence \Leftrightarrow Sentence \\
OPERATOR\ PRECEDENCE \;\;&:\;\; \neg,\ \wedge,\vee,\Rightarrow,\Leftrightarrow
\end{aligned}
$$

Propositional logic（命题逻辑）: Syntax

- Propositional logic is the simplest logic—illustrates basic ideas
- The proposition symbols P_1, P_2 etc are sentences
- If S is a sentence, $\neg S$ is a sentence(negation)
- If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence (conjunction)
- If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence (disjunction)
- If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence (implication)
- If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence (biconditional)

Propositional logic: Semantics

- Each model specifies true/false for each proposition symbol

- E.g. $\dfrac{P_{1,2}}{false} \dfrac{P_{2,2}}{true} \dfrac{P_{3,1}}{false}$

- (With these symbols,8 possible models, can be enumerated automatically.)

- Rules for evaluating truth with respect to a model m:

- $\neg S$ 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

- Simple recursive process evaluates an arbitrary sentence, e.g.,

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

Truth tables for connectives （真值表）

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

$\wedge, \vee$ vs. $\cap, \cup$
区别和联系

$$M(P \wedge Q) = M(P) \cap M(Q)$$
$$M(P \vee Q) = M(P) \cup M(Q)$$

Logical equivalence

- Two sentences are logically equivalent iff true in same models:

    $\alpha \equiv \beta$ if and only if $\alpha \vDash \beta$ and $\beta \vDash \alpha$

$$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha) \quad \text{commutativity of } \wedge$$
$$(\alpha \vee \beta) \equiv (\beta \vee \alpha) \quad \text{commutativity of } \vee$$
$$((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma)) \quad \text{associativity of } \wedge$$
$$((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma)) \quad \text{associativity of } \vee$$
$$\neg(\neg\alpha) \equiv \alpha \quad \text{double-negation elimination}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\beta \Rightarrow \neg\alpha) \quad \text{contraposition}$$
$$(\alpha \Rightarrow \beta) \equiv (\neg\alpha \vee \beta) \quad \text{implication elimination}$$
$$(\alpha \Leftrightarrow \beta) \equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) \quad \text{biconditional elimination}$$
$$\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta) \quad \text{De Morgan}$$
$$\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta) \quad \text{De Morgan}$$
$$(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) \quad \text{distributivity of } \wedge \text{ over } \vee$$
$$(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) \quad \text{distributivity of } \vee \text{ over } \wedge$$

作业

命题逻辑中的知识库*KB*

- *KB*: 满足命题逻辑语法的sentence的集合
- 假设：这组sentence中，一共有*n*个原子命题
- 真值指派（truth assignment）：对每个原子名字赋值
- 一共有2^n种真值指派，其中：使得*KB*中的每个sentence都为真的真值指派，就是*KB*的model
- 在此基础上，在命题逻辑中，我们可以明确的定义

$$KB \models \alpha$$

Inference by enumeration

Depth−first enumeration of all models is sound and complete

```
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
    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))
```

$O(2^n)$ for n symbols; problem is co−NP−complete

Wumpus world sentences

- 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].

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

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

$$B_{2,1}$$

- "Pits cause breezes in adjacent squares"

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

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

Truth tables for inference

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

- Enumerate rows(different assignments to symbols),

  if KB is true in row, check that α is too

Entailment与implication的区别

- Entailment: 逻辑上的概念，刻画两组sentence之间的关系

- Implication: proposition之间的一种运算子，使用真值表刻画其语义

Validity and satisfiability

- A sentence is valid if it is true in all models,

    e.g., $True, \quad A \lor \neg A, \quad A \Rightarrow A, \quad (A \land (A \Rightarrow B)) \Rightarrow B$

- Validity is connected to inference via the Deduction Theorem:

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

- A sentence is satisfiable if it is true in some model

    e.g., $A \lor B, \quad C$

- A sentence is unsatisfiable if it is true in no models

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

- Satisfiability is connected to inference via the following:

    $KB \vDash \alpha$ if and only if $(KB \land \neg \alpha)$ is unsatisfiable

- i.e., prove α by *reductio ad absurdum*

在命题逻辑中运用语义蕴含

奇怪小岛上的逻辑问题

问题1：奇怪小岛上的逻辑问题

Edgar Abercrombie was an anthropologist who was particularly interested in the logic and sociology of lying and truth-telling. One day he decided to visit the Island of Knights and Knaves，in which those called knights always tell the truth and knaves always lie. Furthermore, each inhabitant is either a knight or a knave.

推理问题1：

On the day of his arrival, Abercrombie came across three inhabitants, whom we will call A, B and C.

He asked A: "Are you a knight or a knave?" A answered, but so indistinctly that Abercrombie could not understand what he said. He then asked B: "What did he say?" B replied: "He said that he is a knave." At this point, C piped up and said: "Don't believe that; it's a lie!"

Was C a knight or a knave?

问题1：一个很可能的思考过程

假设   1. A或者是骑士（只说真话），或者是无赖（只说假话），二者必居其一。

2. 假如A是骑士，他不可能说自己是无赖，因为那不是事实。

穷举

3. 假如A是无赖，也同样不可能说自己是无赖，因为那是真话。

反证   4. 所以，<u>A说的话不可能是"我是无赖"</u>。

5. 因此，B说的不是事实，他必然是无赖。

6. 因此，C说的是事实。

7. 结论：C是骑士。

A可以是任何人，所以实际上到这里我们就知道这个岛上不会有任何人说自己是无赖。

这个问题太简单，C没说话就可以判定B是knave。

问题1：把思考过程机械化

- **知识表示**

  - 第i个人是骑士： A_i

  - 第i个人是无赖： $\neg A_i$

  - 第i个人说了一句话P： $A_i \Leftrightarrow P$

- **当前的知识库**

  - B说：A说的话是A是无赖： $A_2 \Leftrightarrow (A_1 \Leftrightarrow \neg A_1)$

  - C说B在撒谎： $A_3 \Leftrightarrow \neg A_2$

问题1：把思考过程机械化

A_1	A_2	A_3	$A_2 \Leftrightarrow (A_1 \Leftrightarrow \neg A_1)$	$A_3 \Leftrightarrow \neg A_2$
0	0	0	1	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

根据蕴含的定义，我们有：$KB \vDash A_3$

问题2：奇怪小岛上的逻辑问题

Edgar Abercrombie was an anthropologist who was particularly interested in the logic and sociology of lying and truth-telling. One day he decided to visit the Island of Knights and Knaves，in which those called knights always tell the truth and knaves always lie. Furthermore, each inhabitant is either a knight or a knave.

推理问题2

Next, Abercrombie met just two inhabitants, A and B. A made the following statement: "Both of us are knaves."

What is A and what is B?

问题3：奇怪小岛上的逻辑问题

Edgar Abercrombie was an anthropologist who was particularly interested in the logic and sociology of lying and truth-telling. One day he decided to visit the Island of Knights and Knaves，in which those called knights always tell the truth and knaves always lie. Furthermore, each inhabitant is either a knight or a knave.

推理问题 3

According to another version of the story, Abercrombie didn't ask A whether he was a knight or a knave (because he would have known in advance what answer he would get), but instead asked A **how many of the three were knaves**. Again A answered indistinctly, so Abercrombie asked B what A had said. B then said that A had said that exactly two of them were knaves. Then, as before, C claimed that B was lying.
Is it now possible to determine whether C is a knight or a knave?

思考题：怎么用机械化的方法解决问题2和3

- 步骤一：如何进行知识表示

- 步骤二：根据蕴含的定义得出结论

形式推演 DEDUCTION

形式推演 Deduction

本节中要定义形式推演,定义公式之间的形式可推演性关系. 形式推演涉及公式的语法结构,它的正确性是能够机械地检验的.

我们先要介绍一些使用记号的约定.

设 $\Sigma = \{A_1, A_2, A_3, \cdots\}$. 为了方便,我们把 Σ 写成序列的形式 $A_1, A_2, A_3, \cdots$. 但是,这样写时,因为 Σ 是集,所以这个序列 $A_1, A_2, A_3, \cdots$ 中的元的次序是没有关系的. 于是,集 $\Sigma \cup \{A\}$ 和 $\Sigma \cup \Sigma'$ 分别可以写作 Σ, A 和 Σ, Σ'.

我们用记号 $\vdash$ 表示形式可推演性关系,用

$$\Sigma \vdash A$$

表示 A 是由 Σ 形式可推演(或形式可证明)的(见本节后面的定义 2.6.1). 形式可推演性关系 $\vdash$ 是 Σ(作为前提的公式)和 A(作为结论的公式)之间的关系. "$\vdash$" 可以读作 "推出". 注意,记号 $\vdash$ 不是形式语言中的符号, $\Sigma \vdash A$ 不是形式语言中的公式. $\Sigma \vdash A$ 是关于 Σ 和 A 的(元语言中的)命题.

形式推演的**11**条规则（某一种推演系统）

形式推演将由形式推演的规则定义. 在命题逻辑中有以下的 11 条形式推演规则.

(Ref)	$A \vdash A$	（自反）
(+)	如果 $\Sigma \vdash A$,	
	则 $\Sigma, \Sigma' \vdash A$.	（增加前提）
($\neg -$)	如果 $\Sigma, \neg A \vdash B$,	
	$\Sigma, \neg A \vdash \neg B$,	
	则 $\Sigma \vdash A$.	（$\neg$ 消去）

形式推演的11条规则

$(\rightarrow -)$　　如果 $\Sigma \vdash A \rightarrow B$,

　　　　　　　　$\Sigma \vdash A$,

　　　　　　则 $\Sigma \vdash B$.　　　　　　　　　　（→消去）

$(\rightarrow +)$　　如果 $\Sigma, A \vdash B$,

　　　　　　则 $\Sigma \vdash A \rightarrow B$.　　　　　　（→引入）

$(\wedge -)$　　如果 $\Sigma \vdash A \wedge B$,

　　　　　　则 $\Sigma \vdash A$,

　　　　　　　　$\Sigma \vdash B$.　　　　　　　　　（∧消去）

$(\wedge +)$　　如果 $\Sigma \vdash A$,

　　　　　　　　$\Sigma \vdash B$,

　　　　　　则 $\Sigma \vdash A \wedge B$,　　　　　　（∧引入）

$(\vee -)$　　如果 $\Sigma, A \vdash C$,

　　　　　　　　$\Sigma, B \vdash C$,

　　　　　　则 $\Sigma, A \vee B \vdash C$.　　　　　（∨消去）

《面向计算机科学的数理逻辑》49页

形式推演的11条规则

$$(\vee +) \quad 如果\ \Sigma \vdash A,$$
$$则\ \Sigma \vdash A \vee B,$$
$$\Sigma \vdash B \vee A. \qquad (\vee 引入)$$

$$(\leftrightarrow -) \quad 如果\ \Sigma \vdash A \leftrightarrow B,$$
$$\Sigma \vdash A,$$
$$则\ \Sigma \vdash B.$$
$$如果\ \Sigma \vdash A \leftrightarrow B,$$
$$\Sigma \vdash B,$$
$$则\ \Sigma \vdash A. \qquad (\leftrightarrow 消去)$$

$$(\leftrightarrow +) \quad 如果\ \Sigma, A \vdash B,$$
$$\Sigma, B \vdash A,$$
$$则\ \Sigma \vdash A \leftrightarrow B. \qquad (\leftrightarrow 引入)$$

《面向计算机科学的数理逻辑》49页

定义 2.6.1(形式可推演性) A 是在命题逻辑中由 Σ 形式可推演(或形式可证明)的,记作

$$\Sigma \vdash A,$$

当且仅当 $\Sigma \vdash A$ 能由(有限次使用)命题逻辑的形式推演规则生成.

由上述定义,$\Sigma \vdash A$ 成立,当且仅当有有限序列

6) $$\Sigma_1 \vdash A_1, \cdots, \Sigma_n \vdash A_n$$

使得 6)中的每一项 $\Sigma_k \vdash A_k (1 \leqslant k \leqslant n)$ 由使用某一形式推演规则生成,并且 $\Sigma_n \vdash A_n$ 就是 $\Sigma \vdash A$(即 $\Sigma_n = \Sigma, A_n = A$).

形式推演：例子

下面先给出例子说明怎样使用这些规则.

例 设 $A \in \Sigma$ 并且 $\Sigma' = \Sigma - \{A\}$. 下面由两个步骤构成一个序列：

(1) $A \vdash A$ （由 (Ref)）.

(2) $A, \Sigma' \vdash A$ （由 (+)，(1)）.

（即 $\Sigma \vdash A$.）

第 (1) 步直接由规则 (Ref) 生成. 第 (2) 步由规则 (+) 使用于 (1) 生成. 在每一步, 所使用的规则和所涉及的前面的步骤(如果涉及了)构成使这一步成立的理由. 我们把理由写在右边. 这些步骤构成一个证明, 它是其中最后一步的证明.

因此, 在这个例子中证明了, 当 $A \in \Sigma$ 时, $\Sigma \vdash A$ 成立. 它记作 $(\in)$, 使用集论中关于元素属于集合的记号. (Ref) 是 $(\in)$ 的特殊情形.

逻辑推导 vs. 形式推演

附注

（1）逻辑推论（$\Sigma \models A$）和形式可推演性（$\Sigma \vdash A$）是不同的事情. 前者属于语义, 后者属于语法. 第四章中的可靠性和完备性将研究这两个概念之间的关系.

（2）逻辑推论和形式可推演性都是在元语言中研究的, 研究时所用的推理是直观的非形式的推理.

（3）前面讲过, $\models$ 和 $\vdash$ 都不是形式语言中的符号. 不应把它们与 $\rightarrow$ 混淆, $\rightarrow$ 是 $\mathscr{L}^p$ 的符号. 是用来构成公式的联结符号. 但是 $\models$（或 $\vdash$）和 $\rightarrow$ 之间有这样的联系: $A \models B$ 当且仅当 $\varnothing \models A \rightarrow B$（即 $A \rightarrow B$ 是重言式）, $A \vdash B$ 当且仅当 $\varnothing \vdash A \rightarrow B$.

形式推演：例子

定理 2.6.4

(i) A→B, A ⊢B.

(ii) A ⊢B→A.

(iii) A→B, B→C ⊢A→C.

(iv) A→(B→C), A→B ⊢A→C.

证 我们选证(iii)：

(1) A→B, B→C, A ⊢A→B   （由(∈)）.

(2) A→B, B→C, A ⊢A   （由(∈)）.

(3) A→B, B→C, A ⊢B   （由(→－), (1), (2)）.

(4) A→B, B→C, A ⊢B→C   （由(∈)）.

(5) A→B, B→C, A ⊢C   （由(→－), (4), (3)）.

(6) A→B, B→C ⊢A→C   （由(→＋), (5)）.

其余的证明留给读者.   □

Homework：其余题目

形式推演：例子

定理 **2.6.9**

(i) $A \vdash A \lor B, B \lor A.$

(ii) $A \lor B \dashv\vdash B \lor A.$　（$\lor$ **交换律**）

(iii) $(A \lor B) \lor C \dashv\vdash A \lor (B \lor C).$　（$\lor$ **结合律**）

(iv) $A \lor B \dashv\vdash \neg A \to B.$

(v) $A \to B \dashv\vdash \neg A \lor B.$

(vi) $\neg (A \lor B) \dashv\vdash \neg A \land \neg B.$　（De Morgen **律**）

(vii) $\neg (A \land B) \dashv\vdash \neg A \lor \neg B.$　（De Morgen **律**）

(viii) $\varnothing \vdash A \lor \neg A.$　（**排中律**）

Homework：证明这些性质

形式推演：常用的定理

- 定理2.6.3，2.6.4，2.6.5，2.6.6，2.6.7，2.6.8，2.6.9，2.6.10,2.6.11

- 可以用基本的11条法则证明，它们则可以在其它证明中使用

《面向计算机科学的数理逻辑》

形式推演：Wumpus

- $P_{x,y}$ is true if there is a pit in [x,y]
- $W_{x,y}$ is true if there is a wumpus in [x,y]
- $B_{x,y}$ is true if the agent perceives BREEZE in [x,y]
- $B_{x,y}$ is true if the agent perceives STENCH in [x,y]
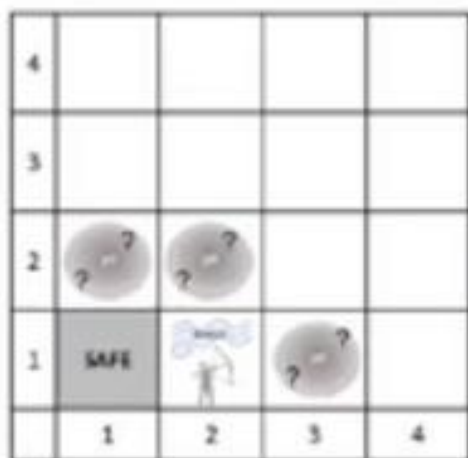
KB

- $R_1 : \neg P_{1,1}.$
- $R_2 : B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1}).$
- $R_3 : B_{2,1} \Leftrightarrow (P_{1,1} \vee P_{2,2} \vee P_{3,1}).$
- $R_4 : \neg B_{1,1}.$
- $R_5 : B_{2,1}.$

证明：$KB \vdash \neg P_{1,2} \wedge \neg P_{2,1}$

形式推演：**Wumpus**

Biconditional elimination from R_2

$R_6: \left(B_{1,1} \Rightarrow \left(P_{1,2} \vee P_{2,1}\right)\right) \wedge \left(\left(P_{1,2} \vee P_{2,1}\right) \Rightarrow B_{1,1}\right)$

And $-$ elimination from R_6

$R_7: \left(P_{1,2} \vee P_{2,1}\right) \Rightarrow B_{1,1}$

Contrapositive from R_7 .

$R_8: \neg B_{1,1} \Rightarrow \neg\left(P_{1,2} \vee P_{2,1}\right)$

Modus Ponens from R_8 .

$R_9: \neg\left(P_{1,2} \vee P_{2,1}\right)$

De Morgan 's rule from R_9

$R_{10}: \neg P_{1,2} \wedge \neg P_{2,1}$

证明：$\mathrm{KB} \vdash \neg P_{1,2} \wedge \neg P_{2,1}$
证明过程并不是一个机械化的方法

Summary

- 逻辑系统
- Syntax：formal structure of sentences
- Semantics: truth of sentences wrt models; Entailment: necessary truth of one sentence given another
- Deduction: formal deduction based on deduction rules

Inference

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

Consequences of _KB_ are a haystack; _α_ is a needle.

Entailment $=$ needle in haystack; inference $=$ finding it

Soundness: i is sound if
    whenever $KB \vdash_i \alpha,$ it is also true that $KB \vDash \alpha$

Completeness: i is complete if
    whenever $KB \vDash \alpha,$ 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_.

- 哥德尔不完全 定理：在一个大的范围内（证明法和问题与正整数存在一一对应关系），不存在既sound又complete的inference过程

62