人工智能：知识表示和推理 I

饶洋辉
计算机学院,
中山大学
raoyangh@mail.sysu.edu.cn
http://cse.sysu.edu.cn/node/2471

课件来源：中山大学刘咏梅教授、王甲海教授；多伦多大学Hector Levesque教授和Sheila McIlraith教授；海军工程大学贲可荣教授等

知识表示和推理

1 谓词逻辑
2 归结推理
3 知识图谱

知识和知识表示

- 数据一般指单独的事实，是信息的载体。信息由符号组成，如文字和数字，但是对符号赋予了一定的意义，因此有一定的用途或价值。知识是由经验总结升华出来的，因此知识是经验的结晶。知识在信息的基础上增加了上下文信息，提供了更多的意义，因此也就更加有用和有价值。知识是随着时间的变化而动态变化的，新的知识可以根据规则和已有的知识推导出来。
- 知识表示就是研究用机器表示上述这些知识的可行性、有效性的一般方法，可以看作是将知识符号化并输入到计算机的过程和方法。

AI对知识表示方法的要求

(1) 表示能力，要求能够正确、有效地将问题求解所需要的各类知识都表示出来。

(2) 可理解性，所表示的知识应易懂、易读。

(3) 便于知识的获取，使得智能系统能够渐进地增加知识，逐步进化。

(4) 便于搜索，表示知识的符号结构和推理机制应支持对知识库的高效搜索，使得智能系统能够迅速地感知事物之间的关系和变化；同时很快地从知识库中找到有关的知识。

(5) 便于推理，要能够从已有的知识中推出需要的答案和结论。

知识表示语言

- 语法：语言的语法描述了组成语句的可能的搭配关系。
- 语义：语义定义了语句所指的世界中的事实。
- 从语法和语义，可以给出使用该语言的Agent的必要的推理机制。
- 基于该推理机制，Agent可以从已知的语句推导出结论，或判断某条信息是不是已蕴涵在现有的知识当中。

知识表示语言

1) 语法规则和语义解释，

2) 用于演绎和推导的规则。

- 程序设计语言比较善于描述算法和具体的数据结构。
- 知识表示语言应该支持知识不完全的情况。
- 不能表达这种不完全性的语言是表达能力不够的语言。

知识表示语言

- 知识表示语言应结合自然语言和程序设计语言的优点：

  1）表达能力很强，简练；

  2）不含糊，上下文无关；

  3）高效，可以推出新的结论。

- 例如谓词逻辑

谓词逻辑

- **Objects** (个体词): represent a specific object by a, b, …
- **Predicate** (谓词): represent the attribute of objects by A (…), B (…), …Z (…)
  - **Relationships** (关系，n元), 如: bigger than, inside, part of, …
  - **Types** (性质/类型，一元), 如: red, round, …
- **Quantifier** (量词)
  - **universal quantifier:** $\forall$
  - **existential quantifier:** $\exists$

$\forall x$ Frog (x) $\Rightarrow$ Green (x):

$\neg\ \forall x$ Likes (x, cat):

$\neg\ \exists x$ Likes (x, cat):

谓词逻辑

- **Objects** (个体词): represent a specific object by a, b, …
- **Predicate** (谓词): represent the attribute of objects by A (…), B (…), …Z (…)
  - **Relationships** (关系，n元), 如: bigger than, inside, part of, …
  - **Types** (性质/类型，一元), 如: red, round, …
- **Quantifier** (量词)
  - **universal quantifier:** $\forall$
  - **existential quantifier:** $\exists$

$\forall x$ Frog (x) $\Rightarrow$ Green (x): All frogs are green

$\neg\ \forall x$ Likes (x, cat): Not everyone likes cat

$\neg\ \exists x$ Likes (x, cat): No one likes cat

谓词逻辑

- ✓" Robot A is to the right of robot B"
- ✓$\forall u\ \forall v$ is_further_right$(u, v) \Leftrightarrow$

  $\exists x_u\ \exists y_u\ \exists x_v\ \exists y_v$ Position$(u, x_u, y_u) \wedge$ Position(v, x_v, y_v)

  $\wedge$ Larger(x_u, x_v)

- Typically, $\Rightarrow$ is the main connective with $\forall$;

  $\wedge$ is the main connective with $\exists$
  - $\forall x$ At$(x$, SYSU$) \Rightarrow$ Smart (x)
  - $\exists x$ At$(x$, SYSU$) \wedge$ Smart (x)
- **Morgan's law**
  - $\forall x\ \mathrm{L} \equiv \neg\ \exists x\ \neg\mathrm{L}$
  - $\neg(\forall x\ \mathrm{L}) \equiv \exists x\ \neg\mathrm{L}$

谓词逻辑

- ✓" Robot A is to the right of robot B"
- ✓$\forall u\ \forall v$ is_further_right(u, v) $\Leftrightarrow$

  $\exists x_u\ \exists y_u\ \exists x_v\ \exists y_v$ Position(u, x_u, y_u) $\wedge$ Position(v, x_v, y_v)

  $\wedge$ Larger(x_u, x_v)

- Typically, $\Rightarrow$ is the main connective with $\forall$;

  $\wedge$ is the main connective with $\exists$
  - $\forall x$ At(x, SYSU) $\Rightarrow$ Smart (x)
  - $\exists x$ At(x, SYSU) $\wedge$ Smart (x)
- **Morgan's law**
  - $\forall x$ L $\equiv \neg\ \exists x\ \neg$L
  - $\neg(\forall x$ L$) \equiv \exists x\ \neg$L

"Not everyone likes cat"

$\neg(\forall x,$ Likes(x, cat))

$\exists x, \neg$Likes(x, cat)

谓词逻辑的应用

例1 “某些患者喜欢所有医生。没有患者喜欢庸医。所以没有医生是庸医。”

解：P(x)表示“x是患者”，

D(x)表示“x是医生”，

Q(x)表示“x是庸医”，

L(x, y)表示“x喜欢y″。

F_1 $(\exists x)(P(x)\wedge(\forall y)(D(y)\rightarrow L(x,y)))$

$F_2:(\forall x)(P(x)\rightarrow(\forall y)(Q(y)\rightarrow\neg L(x,y)))$

$G:(\forall x)(D(x)\rightarrow\neg Q(x))$

目的是证明G是F1和F2的逻辑结论。

谓词逻辑的应用

例2 每个去临潼游览的人或者参观秦始皇兵马俑，或者参观华清池，或者洗温泉澡。凡去临潼游览的人，如果爬骊山就不能参观秦始皇兵马俑，有的游览者既不参观华清池，也不洗温泉澡。

因而有的游览者不爬骊山。

解：定义G（x）表示“x去临潼游览”；

A（x）表示“x参观秦始皇兵马俑”；

B（x）表示“x参观华清池”；

C（x）表示“x洗温泉澡”；

D（x）表示“x爬骊山”。

谓词逻辑的应用

前提：$\forall x(G(x)\rightarrow A(x)\vee B(x)\vee C(x))$ (1)

$\forall x(G(x)\wedge D(x)\rightarrow\neg A(x))$ (2)

$\exists x(G(x)\wedge\neg B(x)\wedge\neg C(x))$ (3)

结论：$\exists x(G(x)\wedge\neg D(x))$

证明：（4）$G(a)\wedge\neg B(a)\wedge\neg C(a)$ 由(3)

（5）$G(a)\rightarrow A(a)\vee B(a)\vee C(a)$ 由(1)

（6）$G(a)\wedge D(a)\rightarrow\neg A(a)$ 由(2)

(7) $A(a)\rightarrow\neg G(a)\vee\neg D(a)$ 由(6)

（8）$G(a)$ 由（4）

（9）$A(a)\vee B(a)\vee C(a)$ 由（5）（8）

（10）$\neg B(a)$，$\neg C(a)$ 由（4）

（11）$A(a)$ 由（9）（10）

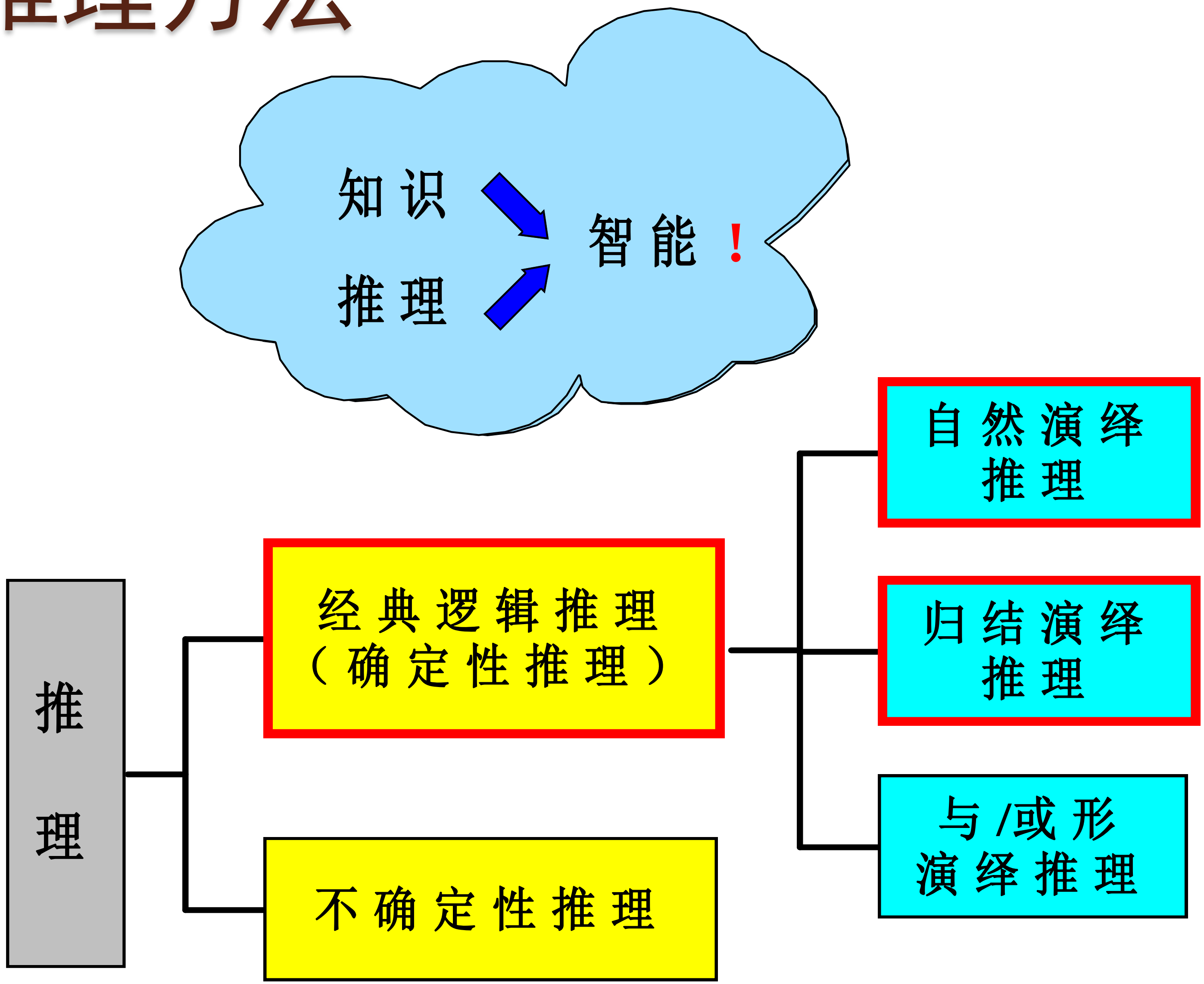
（12）$\neg D(a)$ 由（7）（8）（11）

（13）$\exists x(G(x)\wedge\neg D(x))$ 由（8）（12）

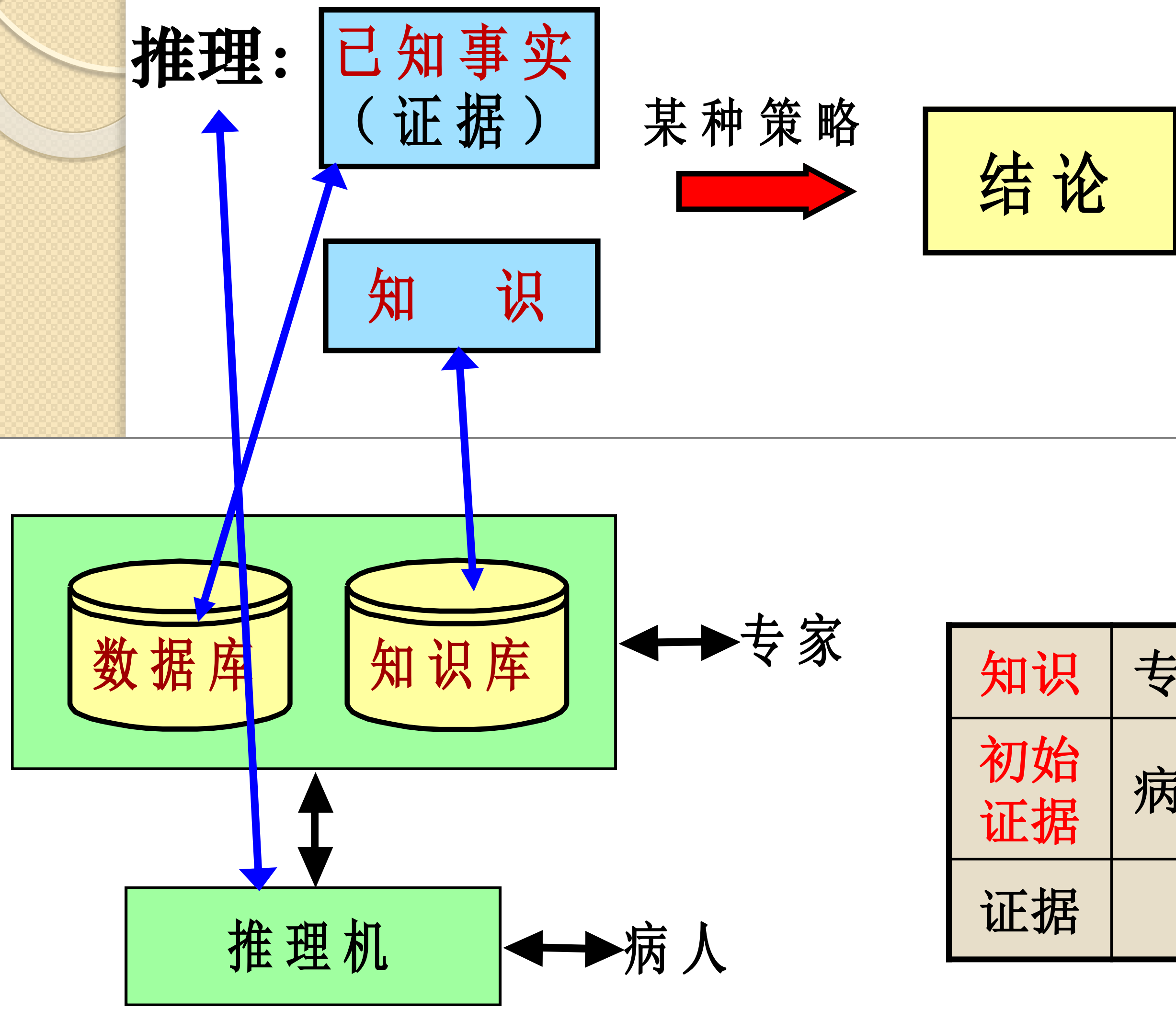
推理方法

- 前面讨论了把知识用某种模式表示出来存储到计算机中去。但是，为使计算机具有智能，还必须使它具有思维能力。推理是求解问题的一种重要方法。因此，推理方法成为人工智能的一个重要研究课题。
- 下面首先讨论关于推理的基本概念，然后介绍鲁宾逊归结原理及其在机器定理证明和问题求解中的应用。鲁宾逊归结原理使定理证明能够在计算机上实现。

推理方法



推理的定义



医疗专家系统

知识	专家的经验、医学常识
初始证据	病人的症状、化验结果
证据	中间结论

推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（1）演绎推理 (deductive reasoning)： 一般 → 个别

- 三段论式（三段论法）

① 足球运动员的身体都是强壮的； （大前提）

② 高波是一名足球运动员； （小前提）

③ 所以，高波的身体是强壮的。 （结 论）

推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（2）归纳推理 **(inductive reasoning):** 个别 → 一般

- 完全归纳推理（必然性推理）
- 不完全归纳推理（非必然性推理）

检查全部产品合格 $\xrightarrow{\text{完全归纳推理}}$ 该厂产品合格

检查全部样品合格 $\xrightarrow{\text{不完全归纳推理}}$ 该厂产品合格

推理方式及其分类

1. 演绎推理、归纳推理、默认推理

（3）默认推理（default reasoning，缺省推理）

- 知识不完全的情况下假设某些条件已经具备所进行的推理。

A **成立**

B **成立？** → **结 论**

（默认*B*成立）

制造鸟笼

鸟会飞？ → **鸟笼要有盖子**

（默认成立）

推理方式及其分类

2. 确定性推理、不确定性推理

（1）**确定性推理**：推理时所用的知识与证据都是确定的，推出的结论也是确定的，其真值或者为真或者为假。

（2）**不确定性推理**：推理时所用的知识与证据不都是确定的，推出的结论也是不确定的。

- 不确定性推理
  - 似然推理 （概率论）
  - 近似推理或模糊推理 （模糊逻辑）

推理方式及其分类

3．启发式推理、非启发式推理

- **启发性知识**：与问题有关且能加快推理过程、提高搜索效率的知识。

- 目标：在脑膜炎、肺炎、流感中选择一个
- 产生式规则

  r_1：脑膜炎

  r_2：肺 炎

  r_3：流 感
- 启发式知识：“脑膜炎危险”、“目前正在盛行流感”

自然演绎推理

- **自然演绎推理：从一组已知为真的事实出发，运用经典逻辑的推理规则推出结论的过程。**
- **推理规则：*P*规则、*T*规则、假言推理、拒取式推理**

- 假言推理： $P,\ \ P\rightarrow Q \quad \Rightarrow Q$
- “如果*x*是金属，则*x*能导电”，“铜是金属” 推出 “铜能导电”

- 拒取式推理： $P\rightarrow Q,\ \neg Q \quad \Rightarrow \quad \neg P$
- “如果下雨，则地下就湿”，“地上不湿” 推出 “没有下雨”

自然演绎推理

错误1——否定前件：$P \to Q, \quad \neg P \quad \not\Rightarrow \quad \neg Q$

（1） 如果下雨，则地上是湿的（ $P \to Q$ ）；

（2）没有下雨（ $\neg P$ ）；

（3）所以，地上不湿（ $\neg Q$ ）。

错误2——肯定后件：$P \to Q, \quad Q \quad \not\Rightarrow \quad P$

（1）如果行星系统是以太阳为中心的，则金星会显示出位相变化（ $P \to Q$ ）；

（2）金星显示出位相变化（ Q ）；

（3） 所以，行星系统是以太阳为中心（ P ）。

自然演绎推理

- **例1 已知事实:**

  （1） 凡是容易的课程小王(Wang)都喜欢;

  （2）C 班的课程都是容易的;

  （3）ds 是 C 班的一门课程。

- **求证：小王喜欢 ds 这门课程。**

自然演绎推理

- **证明:**
- **定义谓词:**
  - $EASY(x)$: x 是容易的
  - $LIKE(x, y)$: x 喜欢 y
  - $C(x)$: x 是 C 班的一门课程

- 已知事实和结论用谓词公式表示:

$(\forall x)(EASY(x) \rightarrow LIKE(Wang, x))$

$(\forall x)(C(x) \rightarrow EASY(x))$

$C(ds)$

$LIKE(Wang, ds)$

自然演绎推理

- **应用推理规则进行推理：**

$(\forall x)(EASY(x) \rightarrow LIKE(Wang, x))$

$EASY(z) \rightarrow LIKE(Wang, z)$ 全称固化

$(\forall x)(C(x) \rightarrow EASY(x))$

$C(y) \rightarrow EASY(y)$ 全称固化

所以 $C(ds), C(y) \rightarrow EASY(y)$

$\Rightarrow EASY(ds)$ *P*规则及假言推理

所以 $EASY(ds), EASY(z) \rightarrow LIKE(Wang, z)$

$\Rightarrow LIKE(Wang, ds)$ *T*规则及假言推理

自然演绎推理

优点：

- 表达定理证明过程自然，易理解。
- 拥有丰富的推理规则，推理过程灵活。
- 便于嵌入领域启发式知识。

缺点：易产生组合爆炸，得到的中间结论一般呈指数形式递增。

归结演绎推理

- 反证法： $P \Rightarrow Q$ ，当且仅当 $P \wedge \neg Q \Leftrightarrow F$ ，
  即 Q为 P 的逻辑推论，当且仅当 $P \wedge \neg Q$ 是不可满足的。

- 定理：$\boldsymbol{Q}$ 为 P_1, P_2 ，…， P_n的逻辑推论，当且仅当
  $(P_1 \wedge P_2 \wedge \cdots \wedge P_n) \wedge \neg Q$是不可满足的。

归结演绎推理

- 思路：定理 $P \Rightarrow Q$ ⟶ $P \wedge \neg Q$ 不可满足

  ⟶ 子句集不可满足 ⟶ **海伯伦定理**

  ↓

  鲁宾逊归结原理

字母表

Logical symbols (fixed meaning and use):

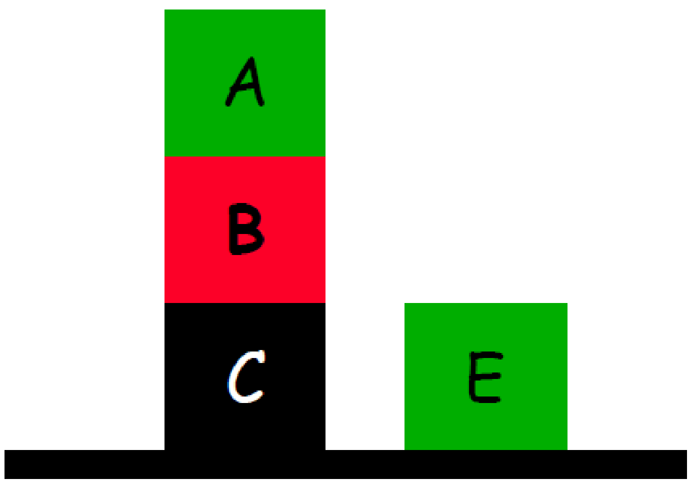
- Punctuation: (,),,,.
- Connectives and quantifiers: $=, \neg, \wedge, \vee, \forall, \exists$
- Variables: $x, x_1, x_2, \ldots, x', x'', \ldots, y, \ldots, z, \ldots$

Non-logical symbols (domain-dependent meaning and use):

- Predicate symbols
  - arity: number of arguments
  - arity 0 predicates: propositional symbols
- Function symbols
  - arity 0 functions: constant symbols

积木世界例子

Environment



Language (Syntax)

- Constants: a,b,c,e
- Functions:
  - No function
- Predicates:
  - on: binary
  - above: binary
  - clear: unary
  - ontable: unary

项和公式

- Every variable is a term
- If $t_1, \ldots, t_n$ are terms and f is a function symbol of arity n, then $f(t_1, \ldots, t_n)$ is a term
- If $t_1, \ldots, t_n$ are terms and P is a predicate symbol of arity n, then $P(t_1, \ldots, t_n)$ is an atomic formula
- If t_1 and t_2 are terms, then $(t_1 = t_2)$ is an atomic formula
- If α and β are formulas, and v is a variable, then $\neg\alpha, (\alpha \wedge \beta), (\alpha \vee \beta), \exists v.\alpha, \forall v.\alpha$ are formulas

记法

- Occasionally add or omit (,)
- Use [,] and {, }
- Abbreviation: $(\alpha \rightarrow \beta)$ for $(\neg\alpha \vee \beta)$
- Abbreviation: $(\alpha \leftrightarrow \beta)$ for $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
- Predicates: mixed case capitalized, *e.g.*, Person, OlderThan
- Functions (and constants): mixed case uncapitalized, *e.g.*, john, father,

变量范围

- Free and bound occurrences of variables
- *e.g.*, $P(x) \wedge \exists x[P(x) \vee Q(x)]$
- A sentence: formula with no free variables
- Substitution: $\alpha[v/t]$ means α with all free occurrences of the v replaced by term t
- In general, $\alpha[v_1/t_1, \ldots, v_n/t_n]$

解释

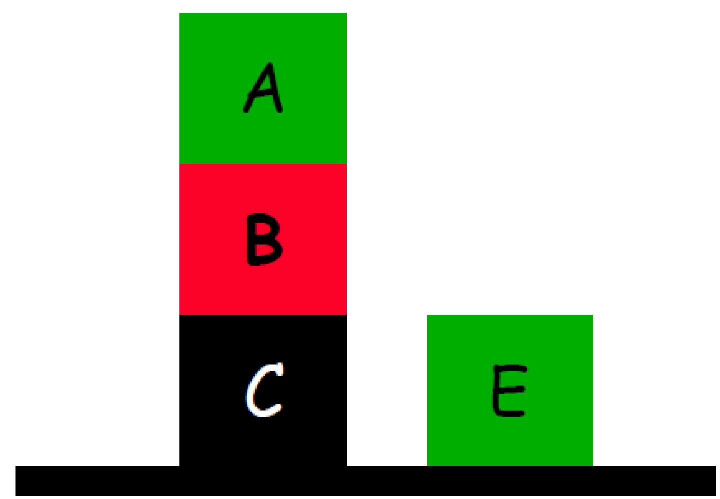
An interpretation is a pair $\Im = \langle D, I \rangle$

- D is the domain, can be any non-empty set
- I is a mapping from the set of predicate and function symbols
- If P is a predicate symbol of arity n, $I(P)$ is an n-ary relation over D, *i.e.*, $I(P) \subseteq D^n$
  - If p is a 0-ary predicate symbol, *i.e.*, a propositional symbol, $I(p) \in \{true, false\}$
- If f is a function symbol of arity n, $I(f)$ is an n-ary function over D, *i.e.*, $I(f) : D^n \to D$
  - If c is a 0-ary function symbol, *i.e.*, a constant symbol, $I(c) \in D$

积木世界例子

- $D = \{\underline{A}, \underline{B}, \underline{C}, \underline{E}\}$
- $\Phi(a) = \underline{A}$, $\Phi(b) = \underline{B}$, $\Phi(c) = \underline{C}$, $\Phi(e) = \underline{E}$.
- $\Psi(on) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C})\}$
- $\Psi(above) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C}),(\underline{A},\underline{C})\}$
- $\Psi(clear)=\{\underline{A},\underline{E}\}$
- $\Psi(ontable)=\{\underline{C},\underline{E}\}$

Environment



项的指称

- Terms denote elements of the domain
- A variable assignment μ is a mapping from the set of variables to the domain D
- $\|v\|_{\Im,\mu} = \mu(v)$
- $\|f(t_1, \ldots, t_n)\|_{\Im,\mu} = I(f)(\|t_1\|_{\Im,\mu}, \ldots, \|t_n\|_{\Im,\mu})$

满足：原子公式

$\Im, \mu \models \alpha$ is read "$\Im, \mu$ satisfies α "

- $\Im, \mu \models P(t_1, \ldots, t_n)$ iff $\langle \|t_1\|_{\Im,\mu}, \ldots, \|t_n\|_{\Im,\mu} \rangle \in I(P)$
- $\Im, \mu \models (t_1 = t_2)$ iff $\|t_1\|_{\Im,\mu} = \|t_2\|_{\Im,\mu}$

满足：联结词

- $\Im, \mu \models \neg\alpha$ iff $\Im, \mu \not\models \alpha$
- $\Im, \mu \models (\alpha \wedge \beta)$ iff $\Im, \mu \models \alpha$ and $\Im, \mu \models \beta$
- $\Im, \mu \models (\alpha \vee \beta)$ iff $\Im, \mu \models \alpha$ or $\Im, \mu \models \beta$

满足：量词

$\mu\{d;v\}$ denotes a variable assignment just like μ, except that it maps v to d

- $\Im, \mu \models \exists v.\alpha$ iff for some $d \in D$, $\Im, \mu\{d;v\} \models \alpha$
- $\Im, \mu \models \forall v.\alpha$ iff for all $d \in D$, $\Im, \mu\{d;v\} \models \alpha$

Let α be a sentence. Then whether $\Im, \mu \models \alpha$ is independent of μ. Thus we simply write $\Im \models \alpha$

积木世界例子

- $D = \{\underline{A}, \underline{B}, \underline{C}, \underline{E}\}$
- $\Phi(a) = \underline{A}$, $\Phi(b) = \underline{B}$, $\Phi(c) = \underline{C}$, $\Phi(e) = \underline{E}$.
- $\Psi(on) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C})\}$
- $\Psi(above) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C}),(\underline{A},\underline{C})\}$
- $\Psi(clear)=\{\underline{A},\underline{E}\}$
- $\Psi(ontable)=\{\underline{C},\underline{E}\}$

$\forall X,Y.\ on(X,Y)\rightarrow above(X,Y)$

✓ $X=\underline{A}$, $Y=\underline{B}$

✓ $X=\underline{C}$, $Y=\underline{A}$

✓ …

$\forall X,Y.\ above(X,Y)\rightarrow on(X,Y)$

✓ $X=\underline{A}$, $Y=\underline{B}$

× $X=\underline{A}$, $Y=\underline{C}$

积木世界例子

- $D = \{\underline{A}, \underline{B}, \underline{C}, \underline{E}\}$
- $\Phi(a) = \underline{A}$, $\Phi(b) = \underline{B}$, $\Phi(c) = \underline{C}$, $\Phi(e) = \underline{E}$.
- $\Psi(on) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C})\}$
- $\Psi(above) = \{(\underline{A},\underline{B}),(\underline{B},\underline{C}),(\underline{A},\underline{C})\}$
- $\Psi(clear)=\{\underline{A},\underline{E}\}$
- $\Psi(ontable)=\{\underline{C},\underline{E}\}$

$\forall X \exists Y. (clear(X) \vee on(Y,X))$

- ✓ $X=\underline{A}$
- ✓ $X=\underline{C}$, $Y=\underline{B}$
- ✓ …

$\exists Y \forall X.(clear(X) \vee on(Y,X))$

- × $Y=\underline{A}$? No! $(X=\underline{C})$
- × $Y=\underline{C}$? No! $(X=\underline{B})$
- × $Y=\underline{E}$? No! $(X=\underline{B})$
- × $Y=\underline{B}$? No! $(X=\underline{B})$

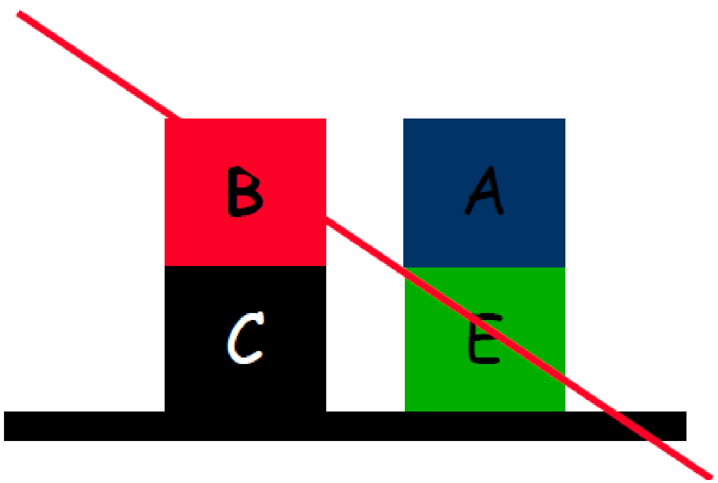
可满足性

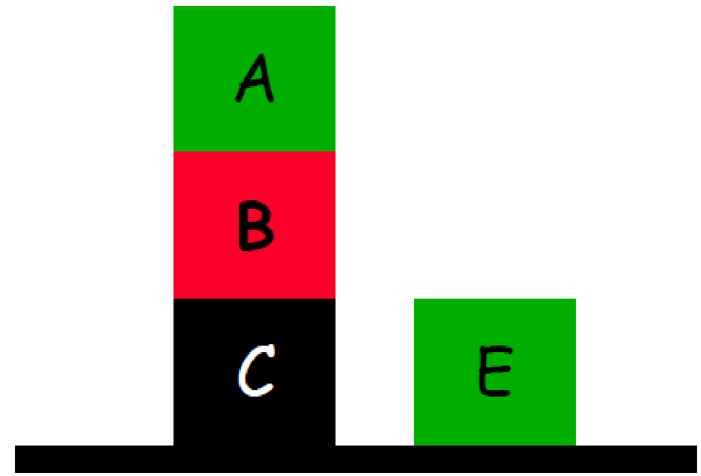
- Let S be a set of sentences
- $\Im \models S$, read $\Im$ satisfies S, if for every $\alpha \in \Im$, $\Im \models \alpha$
- If $\Im \models S$, we say $\Im$ is a model of S
- We say that S is satisfiable if there is $\Im$ s.t. $\Im \models S$, and
- e.g., is $\{\forall x(P(x) \to Q(x)), P(a), \neg Q(a)\}$ satisfiable?

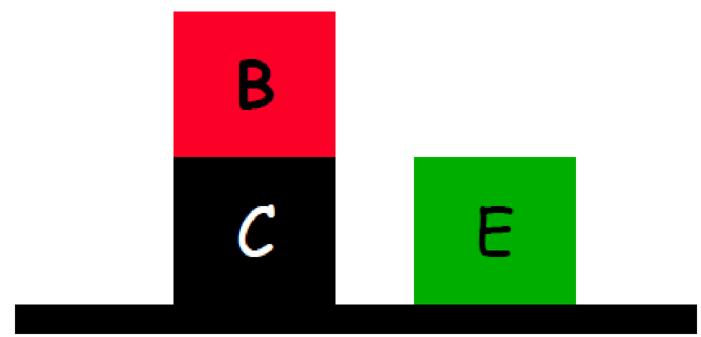
积木世界例子

KB

1. on(b,c)
2. clear(e)





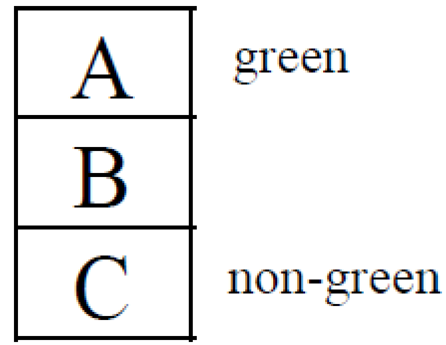


逻辑蕴涵

- $S \models \alpha$ iff for every $\Im$, if $\Im \models S$ then $\Im \models \alpha$
- $S \models \alpha$ is read: S entails α or α is a logical consequence of S
- A special case: $\emptyset \models \alpha$, simply written $\models \alpha$, read "α is valid"
- Note that $\{\alpha_1, \ldots, \alpha_n\} \models \alpha$ iff $\alpha_1 \wedge \ldots \wedge \alpha_n \to \alpha$ is valid iff $\alpha_1 \wedge \ldots \wedge \alpha_n \wedge \neg\alpha$ is unsatisfiable

- Let KB be the set of sentences, and α be the question
- We want to know if $KB \models \alpha$?

示例



- $S = \{On(a,b), On(b,c), Green(a), \neg Green(c)\}$
- $\alpha = \exists x \exists y [Green(x) \wedge \neg Green(y) \wedge On(x,y)]$
- We prove that $S \models \alpha$

示例

- $\forall x A \vee \forall x B \models \forall x(A \vee B)$
- Does $\forall x(A \vee B) \models \forall x A \vee \forall x B$
- $\exists x(A \wedge B) \models \exists x A \wedge \exists x B$
- Does $\exists x A \wedge \exists x B \models \exists x(A \wedge B)$?
- $\exists y \forall x A \models \forall x \exists y A$
- Does $\forall x \exists y A \models \exists y \forall x A$?

The only way to prove that $KB \not\models \alpha$ is to give an interpretation satisfying KB but not α.

逻辑蕴涵和基于知识的系统

- Start with KB representing explicit beliefs, usually what the agent has been told or has learned
- Implicit beliefs: $\{\alpha \mid KB \models \alpha\}$
- Actions depend on implicit beliefs, rather than explicit beliefs