Artificial Intelligence

3. 谓词逻辑与机器证明

罗晓鹏

xpluo@nju.edu.cn

工管 · 南京大学 · 2022 秋

Outline

1. 量词和谓词
2. 一阶语言 $\mathscr{L}$ 和一阶系统 $K_{\mathscr{L}}$
3. $K_{\mathscr{L}}$ 的性质
4. 一阶逻辑的归结推理
5. 知识工程与专家系统
6. 形式系统的扩充
7. 不完备性定理简介
8. 讨论 (选)

量词和谓词

量词和谓词

- 直言三段论对命题的解析：拆句成词
- (存在/任意)主词<u>(是/不是)宾词</u>
- 命题语言无法表达“存在”和“任意”的含义
- 拆解的符号化：
  - ▷ 命题：断言个体具有某种属性 (张三是个 badboy)
  - ▷ 主词：命题所断言的个体，用小写字母表示
  - ▷ 谓词：个体所具有的属性，用大写字母表示
  - ▷ $B(x)$, 其中 x 代表<u>张三</u>，B 代表<u>是 badboy</u>
  - ▷ 李四不是: $\neg B(y)$
- 全称量词：$(\forall x)(A(x) \to B(x))$,   All
- 存在量词：$(\exists x)(P(x) \wedge Q(x))$,   Exist

一阶语言 $\mathscr{L}$ 和一阶系统 $K_{\mathscr{L}}$

一阶语言 $\mathscr{L}$ 的符号

定义3.1 (一阶语言 $\mathscr{L}$ 的符号)

- <u>常元</u>: $a_1, a_2, \cdots$
- 变元: $x_1, x_2, \cdots$
- <u>谓词</u>: $A_1, A_2, \cdots$
- <u>函词</u>: $f_1, f_2, \cdots$
- 连接符: $\neg, \rightarrow$
- 量词: $\forall$
- 括号: $(,), \cdots$

划线部分不是必须的.

定义3.2 (一阶语言 $\mathscr{L}$ 的项)

$\mathscr{L}$ 中的一个项被定义为:

(i) 变元和常元是项.

(ii) 若 f 是函词而 $t_1, \cdots, t_n$ 是项,则 $f(t_1, \cdots, t_n)$ 也是项.

(iii) 有且仅有以上两种项.

定义3.3 (一阶语言 $\mathscr{L}$ 的公式)

$\mathscr{L}$ 中的一个公式被定义为:

(i) 若 A 是谓词而 $t_1, \cdots, t_n$ 是项,则 $A(t_1, \cdots, t_n)$ 是公式.

(ii) 若 $\mathscr{A}, \mathscr{B}$ 是公式,则 $\neg\mathscr{A}, \mathscr{A} \to \mathscr{B}$ 和 $(\forall x)\mathscr{A}$ 也是公式.

(iii) 有且仅有以上两种公式.

逻辑符号的最小列表

- 逻辑符号: $\neg, \vee, \wedge, \rightarrow, \leftrightarrow, \forall, \exists$
- 最小列表: $\neg, \rightarrow, \forall$
  - ▷ $(\exists x)\mathscr{A}$ 是 $\neg((\forall x)(\neg\mathscr{A}))$ 的缩写
  - ▷ $\mathscr{A} \vee \mathscr{B}$ 是 $\neg\mathscr{A} \rightarrow \mathscr{B}$ 的缩写
  - ▷ $\mathscr{A} \wedge \mathscr{B}$ 是 $\neg(\mathscr{A} \rightarrow (\neg\mathscr{B}))$ 的缩写

定义3.4 (一阶系统)

给定一阶语言 $\mathscr{L}$，系统 $K_{\mathscr{L}}$ 由以下公理和规则构成：

1. 公理：对 $\mathscr{L}$ 中任意的公式 $\mathscr{A},\mathscr{B},\mathscr{C}$,

   $K1: \mathscr{A} \to (\mathscr{B} \to \mathscr{A})$.

   $K2: (\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C}))$.

   $K3: (\neg\mathscr{B} \to \neg\mathscr{A}) \to (\mathscr{A} \to \mathscr{B})$.

   $K4: (\forall x)\mathscr{A} \to \mathscr{A}$.

   $K5: (\forall x)(\mathscr{A} \to \mathscr{B}(x)) \to (\mathscr{A} \to (\forall x)\mathscr{B}(x))$.

   $K6: (\forall x)\mathscr{A}(x) \to \mathscr{A}(t)$, t 是 $\mathscr{L}$ 中的一个项.

2. 规则：对任意的公式 $\mathscr{A},\mathscr{B}$,

   (1) 分离规则：由 $\mathscr{A}$ 和 $\mathscr{A} \to \mathscr{B}$ 可以推出 $\mathscr{B}$.

   (2) 概括规则：由 $\mathscr{A}$ 可以推出 $(\forall x)\mathscr{A}(x)$, x是任意变元.

$K_{\mathscr{L}}$ 的性质

命题3.5

(a) K 的每个定理都是逻辑有效的.

(b) 不存在 K 中的公式 $\mathscr{A}$ 使得 $\mathscr{A}$ 和 $\neg\mathscr{A}$ 都是 L 中的定理.

(c) 如果 $\mathscr{A}$ 是 $\mathscr{L}$ 中的一个公式且是重言式，那么 $\vdash_K \mathscr{A}$.

(c') 如果 $\mathscr{A}$ 是 $\mathscr{L}$ 中一个逻辑有效的公式，那么 $\vdash_K \mathscr{A}$.

(d) 不存在一种可行的方法判定 K 中的给定公式是否为定理.

(e) 若给定公式是 K 中的定理，则存在一种可行的判定方法.

注

1. 证明略(不做要求).
2. (d)和(e)分别称为不可判定性以及半可判定性.

一阶逻辑的归结推理

一阶逻辑的归结概述

- 一阶逻辑是命题逻辑的拓展
- 1930 年的 Herbrand 定理：
  - ▷ 保证了一阶逻辑可转换到命题逻辑
  - ▷ 给出了一阶逻辑的半可判定算法
- 1936 年，Turing 和 Church 证明了一阶逻辑的不可判定性
- 1965 年，Robinson 针对一阶逻辑提出了归结原理，有效地提高了机器定理证明的实际效率

例3.6

Fido 是条狗。狗都是动物。动物都会死。

求证：Fido 会死

- 将前提转换为子句形式

  $\text{dog}(Fido)$

  $\forall X(\text{dog}(X) \to \text{animal}(X)) \;\to\; \neg\text{dog}(X) \vee \text{animal}(X)$

  $\forall X(\text{animal}(X) \to \text{die}(X)) \;\to\; \neg\text{animal}(X) \vee \text{die}(X)$

- 将目标的否定式子句形式加入前提

  $\neg\text{die}(Fido)$

- 归结论证：

$$
\begin{tikzcd}
\neg\mathrm{dog}(X)\vee\mathrm{animal}(X) \arrow[dr] & & \neg\mathrm{animal}(X)\vee\mathrm{die}(X) \arrow[dl] \\
& \neg\mathrm{dog}(X)\vee\mathrm{die}(X) \arrow[d] & \mathrm{dog}(Fido) \arrow[dl] \\
& \mathrm{die}(Fido) \arrow[d] & \neg\mathrm{die}(Fido) \arrow[dl] \\
& \varnothing &
\end{tikzcd}
$$

- 归结树

例3.7

不贫穷而且聪明的人是快乐的。能读书的人是聪明的。

John 能读书且不贫穷。快乐的人过着激动人心的生活。

求证：有人过着激动人心的生活。

- 一阶谓词表述
  - ▷ $\forall X(\neg \mathrm{poor}(X) \wedge \mathrm{smart}(X) \rightarrow \mathrm{happy}(X))$
  - ▷ $\forall Y(\mathrm{read}(Y) \rightarrow \mathrm{smart}(Y))$
  - ▷ $\mathrm{read}(john) \wedge \neg \mathrm{poor}(john)$
  - ▷ $\forall Z(\mathrm{happy}(Z) \rightarrow \mathrm{exciting}(Z))$
  - ▷ $\neg \exists W(\mathrm{exciting}(W))$

基于搜索的归结论证：转换为子句形式

- $\forall X(\neg \text{poor}(X) \wedge \text{smart}(X) \rightarrow \text{happy}(X))$

  $\leftrightarrow \ \text{poor}(X) \vee \neg \text{smart}(X) \vee \text{happy}(X))$
- $\forall Y(\text{read}(Y) \rightarrow \text{smart}(Y))$

  $\leftrightarrow \ \neg \text{read}(Y) \vee \text{smart}(Y))$
- $\text{read}(john)$
- $\neg \text{poor}(john)$
- $\forall Z(\text{happy}(Z) \rightarrow \text{exciting}(Z))$

  $\leftrightarrow \ \neg \text{happy}(Z) \vee \text{exciting}(Z)$
- $\neg \exists W(\text{exciting}(W))$

  $\leftrightarrow \ \neg \text{exciting}(W)$

基于搜索的归结论证：搜索$_1$

poor(X) v ¬ smart(X) v happy(X)

¬ read(Y) v smart(Y)

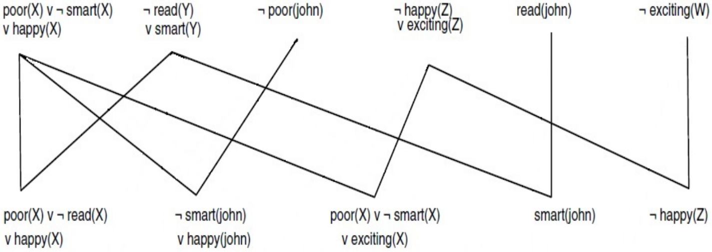
¬ poor(john)

¬ happy(Z) v exciting(Z)

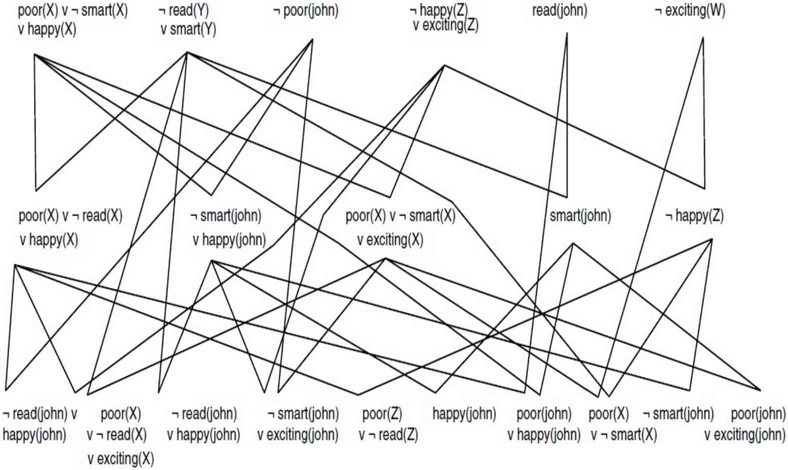
read(john)

¬ exciting(W)

基于搜索的归结论证：搜索$_2$



基于搜索的归结论证：搜索$_3$



知识工程与专家系统

- 专家系统:
  - 知识表示
  - 问题求解
- Prolog 与日本第五代计算机:
  - Prolog：一阶谓词逻辑、归结、搜索
  - 日本第五代计算机
- 硬件与系统的局限性:
  - 硬件的飞速发展与更新
  - 知识表示存在统一框架?
  - 日本第五代计算机的失败

形式系统的扩充

形式系统的扩充

- 演绎系统:
  - ▷ 固定形式语言 $\mathscr{L}$
  - ▷ 在该语言的基础上定义一个形式演绎系统 $K_{\mathscr{L}}$
- 数学系统:
  - ▷ 针对具体背景将 $\mathscr{L}$ 扩充为一个特有语言 $\mathscr{L}'$
  - ▷ 对 $K_{\mathscr{L}'}$ 添加规则和公理，形成一个扩充系统 $K'_{\mathscr{L}'}$
- 针对某一门学科，其理论可分为三个方面:
  - ▷ 形式逻辑
  - ▷ 直观背景
  - ▷ 应用

不完备性定理简介

哥德尔不完备性定理

- 1898年，希尔伯特的《几何基础》
  - 相比与《几何原本》
  - 独立性
  - 相容性 (一致性、无矛盾性)
- 1922 年，希尔伯特纲领
  - 完备性：系统内所有能表达的命题均可判定真假
- 1931 年，哥德尔不完备性定理
  - 蕴含皮亚诺算术公理的形式系统，若相容，则不完备
  - 过度解读 (认知极限、法律体系、测不准原理)
- 集合论公理系统的一致性仍然未知

讨论 (选)

讨论 (选)

- 逻辑系统的基础性
- 元语言与元理论
- 逻辑是智能的高等形态，但为何实现简单？
- 逻辑推理与机器学习的结合

END