人工智能：知识表示和推理 III

授课对象：计算机科学与技术专业 二年级

课程名称：人工智能（专业必修）

节选内容：第三章 知识表示和推理 II

课程学分：3学分

谓词逻辑的归结推理

谓词逻辑的归结推理

Example 求两个语句$P(\mathrm{a}, x, f(g(y)))$和$P(z, f(z), f(u))$的最一般合一项

1. 初始化： $\sigma = \{\}$， $S = \{P(\mathrm{a}, x, f(g(y))), P(z, f(z), f(u))\}$
2. 找出S的差异集$D = \{\mathrm{a}, z\}$

   $\sigma = \{z/\mathrm{a}\}$， $S = \{P(\mathrm{a}, x, f(g(y))), P(\mathrm{a}, f(\mathrm{a}), f(u))\}$
3. 找出S的差异集$D = \{x, f(\mathrm{a})\}$

   $\sigma = \{z/\mathrm{a}, x/f(\mathrm{a})\}$， $S = \{P(\mathrm{a}, f(\mathrm{a}), f(g(y))), P(\mathrm{a}, f(\mathrm{a}), f(u))\}$
4. 找出S的差异集$D = \{g(y), u\}$

   $\sigma = \{z/\mathrm{a}, x/f(\mathrm{a}), u/g(y)\}$， $S = \{P(\mathrm{a}, f(\mathrm{a}), f(g(y))), P(\mathrm{a}, f(\mathrm{a}), f(g(y)))\}$

返回 语句$P(\mathrm{a}, x, f(g(y)))$和$P(z, f(z), f(u))$的最一般合一项$\sigma = \{z/\mathrm{a}, x/f(\mathrm{a}), u/g(y)\}$

谓词逻辑的归结推理

Example

$$\forall x(\forall y P(x,y)) \to \neg\forall y(Q(x,y) \to R(x,y))$$

谓词公式化为子句集的步骤

1. 消去蕴涵和等价符号

$$\forall x(\neg\forall y P(x,y)) \vee \neg\forall y(\neg Q(x,y) \vee R(x,y))$$

$$P \to Q \Leftrightarrow \neg P \vee Q$$

$$P \leftrightarrow Q \Leftrightarrow (P \wedge Q) \vee (\neg P \wedge \neg Q)$$

2. 内移否定符号¬，将其移到紧靠谓词的位置上

$$\forall x(\exists y \neg P(x,y)) \vee \exists y(Q(x,y) \wedge \neg R(x,y))$$

双重否定律 $\neg(\neg P) \Leftrightarrow P$

德摩根律 $\neg(P \wedge Q) \Leftrightarrow \neg P \vee \neg Q$

$\neg(P \vee Q) \Leftrightarrow \neg P \wedge \neg Q$

量词转换律 $\neg\exists x P \Leftrightarrow \forall x \neg P$

$\neg\forall x P \Leftrightarrow \exists x \neg P$

谓词逻辑的归结推理

谓词公式化为子句集的步骤

3. **变量标准化**，对变量作必要的换名，使每一量词只约束一个唯一的变量名

$$\forall x(\exists y \neg P(x,y) \vee \exists z(Q(x,z) \wedge \neg R(x,z)))$$

4. **消去存在量词（Skolemize）**。对于待消去的存在量词，若不在任何全称量词辖域之内，则用Skolem常量替代公式中存在量词约束的变量；若受全称量词约束，则要用Skolem函数替代存在量词约束的变量，然后就可消去存在量词。

Example

$$\forall x(\forall y\, P(x,y)) \rightarrow \neg \forall y(Q(x,y) \rightarrow R(x,y))$$

$$y = f(x), z = g(x)$$

$$\forall x(\neg P(x,f(x)) \vee (Q(x,g(x)) \wedge \neg R(x,g(x))))$$

Skolemize

对于一般情况

$$\forall x_1 \forall x_2 \ldots \forall x_n \exists y\, P(x_1,x_2,\ldots,x_n,y)$$

存在量词y的Skolem函数为

$$y = f(x_1,x_2,\ldots,x_n)$$

谓词逻辑的归结推理

Example

$$\forall x(\forall y\, P(x,y)) \rightarrow \neg\forall y(Q(x,y) \rightarrow R(x,y))$$

谓词公式化为子句集的步骤

5. **化为前束型**，即前束型=(前缀)[母式]。其中，前缀为全称量词串，母式为不含量词的谓词公式 $\forall x(\neg P(x,f(x)) \vee (Q(x,g(x)) \wedge \neg R(x,g(x))))$

$$P \wedge (Q \vee R) \Leftrightarrow (P \wedge Q) \vee (P \wedge R)$$
$$P \vee (Q \wedge R) \Leftrightarrow (P \vee Q) \wedge (P \vee R)$$

6. **把母式化成合取范式。**反复使用结合律和分配律，将母式表达成合取范式的Skolem标准形

$$\forall x((\neg P(x,f(x)) \vee Q(x,g(x))) \wedge (\neg P(x,f(x)) \vee \neg R(x,g(x))))$$

7. **略去全称量词。**由于母式的变量均受全称量词的约束，因此可省略掉全称量词

$$(\neg P(x,f(x)) \vee Q(x,g(x))) \wedge (\neg P(x,f(x)) \vee \neg R(x,g(x)))$$

谓词逻辑的归结推理

Example

$$\forall x(\forall y\, P(x,y)) \rightarrow \neg\forall y(Q(x,y) \rightarrow R(x,y))$$

谓词公式化为子句集的步骤

8. 把母式用子句集表示。把母式中每一个合取元称为一个子句，省去合取联结词，这样就可把母式写成集合的形式表示，每一个元素就是一个子句。

$$\{\left(\neg P(x,f(x)), Q(x,g(x))\right), \left(\neg P(x,f(x)), \neg R(x,g(x))\right)\}$$

9. 子句变量标准化。对某些变量重新命名，使任意两个子句不会有相同的变量出现。这是因为在使用子句集进行证明推理的过程中，有时需要例化某一个全称量词约束的变量，该步骤可以使公式尽量保持其一般化形式，增加了应用过程的灵活性。

$$\{\left(\neg P(x,f(x)), Q(x,g(x))\right), \left(\neg P(y,f(y)), \neg R(y,g(y))\right)\}$$

谓词逻辑的归结推理

Example

将下列谓词公式化为不含存在量词的前束型。

$$\exists x \forall y(\forall z(P(z) \wedge \neg Q(x,z)) \rightarrow R(x,y,f(a)))$$

谓词公式化为子句集的步骤

1. 消去存在量词

$$\forall y(\forall z(P(z) \wedge \neg Q(b,z)) \rightarrow R(b,y,f(a)))$$

2. 消去蕴涵符号

$$\forall y(\neg \forall z(P(z) \wedge \neg Q(b,z)) \vee R(b,y,f(a)))$$

$$\forall y(\exists z(\neg P(z) \vee Q(b,z)) \vee R(b,y,f(a)))$$

3. 设z的Skolem函数是$g(y)$，则

$$\forall y(\neg P(g(y)) \vee Q(b,g(y)) \vee R(b,y,f(a)))$$

谓词逻辑的归结推理

Example

将下列谓词公式化为子句集

$$\forall x\{[\neg P(x)\vee\neg Q(x)]\to\exists y[S(x,y)\wedge Q(x)]\}\wedge\forall x[P(x)\vee B(x)]$$

谓词公式化为子句集的步骤

$$\forall x\{\neg[\neg P(x)\vee\neg Q(x)]\vee\exists y[S(x,y)\wedge Q(x)]\}\wedge\forall x[P(x)\vee B(x)]$$

$$\forall x\{[P(x)\wedge Q(x)]\vee\exists y[S(x,y)\wedge Q(x)]\}\wedge\forall x[P(x)\vee B(x)]$$

$$\forall x\{[P(x)\wedge Q(x)]\vee\exists y[S(x,y)\wedge Q(x)]\}\wedge\forall w[P(w)\vee B(w)]$$

$$\forall x\{[P(x)\wedge Q(x)]\vee[S(x,f(x))\wedge Q(x)]\}\wedge\forall w[P(w)\vee B(w)]$$ （设y的Skolem函数是$f(x)$）

$$\forall x\forall w\{\{[P(x)\wedge Q(x)]\vee[S(x,f(x))\wedge Q(x)]\}\wedge[P(w)\vee B(w)]\}$$

$$\forall x\forall w\{\{[Q(x)\wedge P(x)]\vee[Q(x)\wedge S(x,f(x))]\}\wedge[P(w)\vee B(w)]\}$$

$$\forall x\forall w\{Q(x)\wedge[P(x)\vee S(x,f(x))]\wedge[P(w)\vee B(w)]\}$$

$$Q(x)\wedge[P(x)\vee S(x,f(x))]\wedge[P(w)\vee B(w)]$$ $$\{Q(x),(P(y),S(y,f(y))),(P(w),B(w))\}$$

谓词逻辑的归结推理

Example

已知：

（1） 会朗读的人是识字的

（2） 海豚都不识字

（3） 有些海豚是很机灵的

求证：有些很机灵的东西不会朗读

Prove

用谓词逻辑描述问题：

（1） $\forall x\ (R(x) \rightarrow W(x))$

（2） $\forall x\ (D(x) \rightarrow \neg W(x))$

（3） $\exists x\ (D(x) \rightarrow S(x))$

（结论） $\exists x\ (S(x) \wedge \neg R(x))$

谓词逻辑的归结推理

Prove

前提化简，待证结论取反并化成子句形，求得子句集：

(1) $(\neg R(x), W(x))$

(2) $(\neg D(y), \neg W(y))$

(3) $D(\mathrm{a})$

(4) $S(\mathrm{a})$

(5) $(\neg S(z), R(z))$

进行归结：

(6) [4, 5] $\{z/\mathrm{a}\}$ $(R(\mathrm{a}))$

(7) [1, 6] $\{x/\mathrm{a}\}$ $(W(\mathrm{a}))$

(8) [2, 7] $\{y/\mathrm{a}\}$ $(\neg D(\mathrm{a}))$

(9) [3, 8] NIL

得证

可判定 vs 不可判定

- 可判定问题：如果存在一个算法或过程，该算法用于求解该类问题时，可在有限步内停止，并给出正确的解答。
- 如果不存在这样的算法或过程则称这类问题是不可判定的。例如，There can be no procedure to decide if a set of clauses is satisfiable.

可判定 vs 不可判定

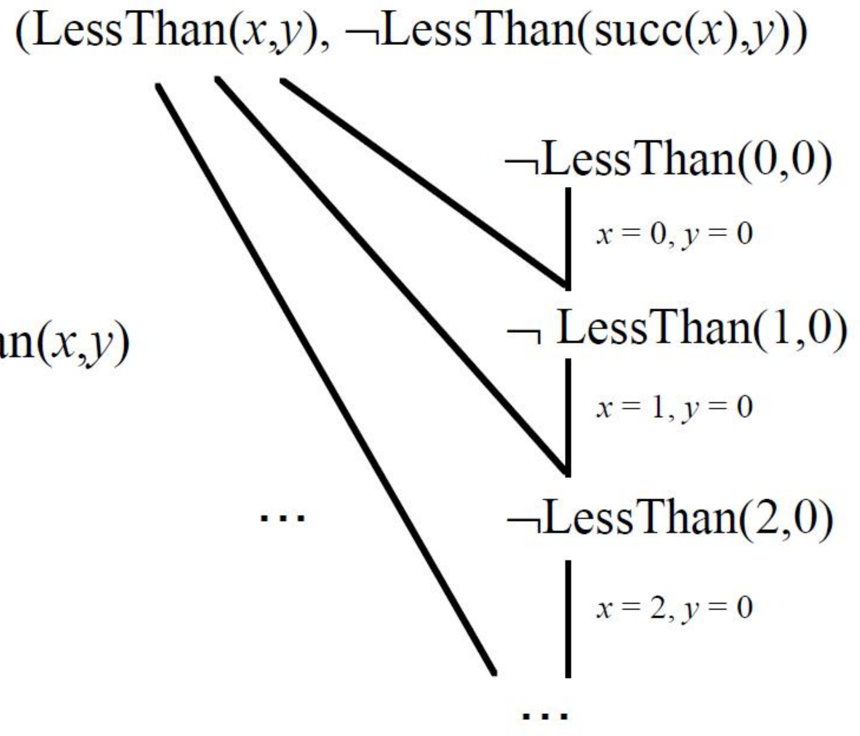
KB:

LessThan(succ(x),y) -> LessThan(x,y)

Query:

LessThan(0,0)

Should fail since KB $\not\models$ Q



Infinite branch of resolvents

对于谓词逻辑，若子句集不可满足，则必存在一个从该子句集到空子句的推导；若从子句集存在一个到空子句的推导，则该子句集是不可满足的。如果没有归结出空子句，则既不能说 S 不可满足，也不能说 S 是可满足的。

问题求解

问题求解

步骤 应用归结原理求解问题：

（1）已知前提 F 用谓词公式表示，并化为子句集 S；

（2）把待求解的问题 P 用谓词公式表示，并否定P， 再与 *answer* 构成析取式（$\neg P \vee answer$）；

（3）把（$\neg P \vee$ answer）化为子句集，并入到子句集S中，得到子句集S'；

（4）对 S' 应用归结原理进行归结；

（5）若得到归结式*answer*，则答案就在*answer*中。

问题求解

Example

设A,B,C三人中有人从不说真话，也有人从不说假话。某人向这三人分别提出同一个问题：谁是说谎者？A答：“B和C都是说谎者”；B答：“A和C都是说谎者”；C答：“A和B中至少有一个是说谎者”。求谁是老实人，谁是说谎者？

设用T(x)表示x说真话。

$T(C) \vee T(A) \vee T(B)$

$\neg T(C) \vee \neg T(A) \vee \neg T(B)$

$T(A) \rightarrow \neg T(B) \wedge \neg T(C)$

$\neg T(A) \rightarrow T(B) \vee T(C)$

$T(B) \rightarrow \neg T(A) \wedge \neg T(C)$

$\neg T(B) \rightarrow T(A) \vee T(C)$

$T(C) \rightarrow \neg T(A) \vee \neg T(B)$

$\neg T(C) \rightarrow T(A) \wedge T(B)$

把所有公式化成子句集，得到S:

(1) $\neg T(A) \vee \neg T(B)$

(2) $\neg T(A) \vee \neg T(C)$

(3) $T(C) \vee T(A) \vee T(B)$

(4) $\neg T(B) \vee \neg T(C)$

(5) $\neg T(C) \vee \neg T(A) \vee \neg T(B)$

(6) $T(A) \vee T(C)$

(7) $T(B) \vee T(C)$

问题求解

Example

设A,B,C三人中有人从不说真话，也有人从不说假话。某人向这三人分别提出同一个问题：谁是说谎者？A答："B和C都是说谎者"；B答："A和C都是说谎者"；C答："A和B中至少有一个是说谎者"。求谁是老实人，谁是说谎者？

把所有公式化成子句集，得到S：

(1) $\neg T(A) \vee \neg T(B)$

(2) $\neg T(A) \vee \neg T(C)$

(3) $T(C) \vee T(A) \vee T(B)$

(4) $\neg T(B) \vee \neg T(C)$

(5) $\neg T(C) \vee \neg T(A) \vee \neg T(B)$

(6) $T(A) \vee T(C)$

(7) $T(B) \vee T(C)$

下面先求谁是老实人。把$\neg T(x) \vee Answer(x)$并入S得到S'。即多一个子句：

(8) $\neg T(x) \vee Answer(x)$

应用归结原理对S1进行归结：

(9) $\neg T(A) \vee T(C)$　　(1)和(7)归结

(10) $T(C)$　　(6)和(9)归结

(11) $Answer(C)$　　(8)和(10)归结

所以C是老实人，即C从不说假话。

问题求解

Example

设A,B,C三人中有人从不说真话，也有人从不说假话。某人向这三人分别提出同一个问题：谁是说谎者？A答：“B和C都是说谎者”；B答：“A和C都是说谎者”；C答：“A和B中至少有一个是说谎者”。求谁是老实人，谁是说谎者？

把所有公式化成子句集，得到S:

(1) $\neg T(A) \vee \neg T(B)$

(2) $\neg T(A) \vee \neg T(C)$

(3) $T(C) \vee T(A) \vee T(B)$

(4) $\neg T(B) \vee \neg T(C)$

(5) $\neg T(C) \vee \neg T(A) \vee \neg T(B)$

(6) $T(A) \vee T(C)$

(7) $T(B) \vee T(C)$

下面证明A不是老实人，即证明$\neg T(A)$。

对$\neg T(A)$进行否定，并入S中，得到子句集S2，即S2比S多如下子句：

(8) $\neg(\neg T(A))$, 即$T(A)$

应用归结原理对S2进行归结：

(9) $\neg T(A) \vee T(C)$　　(1)和(7)归结

(10) $\neg T(A)$　　(2)和(9)归结

(11) NIL　　(8)和(10)归结

所以A不是老实人。同样可以证明B也不是老实人。

吴氏方法

吴文俊

吴文俊（1919年5月12日—2017年5月7日），1919年5月12日出生于上海，祖籍浙江嘉兴，数学家，中国科学院院士，中国科学院数学与系统科学研究院研究员，系统科学研究所名誉所长。

吴文俊先生的研究工作涉及数学的诸多领域，其主要成就表现在拓扑学和数学机械化两个领域。他为拓扑学做了奠基性的工作；他的示性类和示嵌类研究被国际数学界称为“吴公式”，“吴示性类”，“吴示嵌类”，至今仍被国际同行广泛引用。

吴氏方法

几何定理机器证明

第一步是几何问题代数化，建立坐标系，并将命题涉及的几何图形的点选取适当的坐标；然后把命题的条件和结论表示为坐标的多项式方程组；最后判断条件方程组的解是否满足结论方程。

通常的几何命题涉及的多项式方程组都是非线形的，一般无法将约束变元求出。吴氏方法是利用伪除法判定条件方程组的解是否是结论方程组的解。而且利用吴氏方法不仅可以判断定理的正确与否，还可以自动找出定理赖以成立的非退化条件，这是传统的做法无法做到的。

王氏算法

王浩

王浩（1921年5月20日—1995年5月13日）数理逻辑学家。祖籍山东省德州市齐河县，生于山东省济南市。

20世纪50年代初被选为美国科学院院士，后又被选为不列颠科学院外国院士。1983年，被国际人工智能联合会授予第一届“数学定理机械证明里程碑奖”，以表彰他在数学定理机械证明研究领域中所作的开创性贡献。著有《数理逻辑概论》、《从数学到哲学》、《哥德尔》、《超越分析哲学》等专著。

王氏算法

一阶逻辑定理证明

1959年，王浩用他首创的“王氏算法”，在一台速度不高的IBM-704电脑上再次向《数学原理》发起挑战。不到9分钟，王浩的机器把这本数学史上视为里程碑的著作中全部（350条以上）的一阶逻辑定理，统统证明了一遍。

该书作者，数学大师罗素得知此事后，在信里写到：“我真希望，在怀特海和我浪费了10年的时间用手算来证明这些定理之前，就知道有这种可能。”王浩教授因此被国际上公认为机器定理证明的开拓者之一。

小结

内容总结

- 谓词公式化为子句集
- 应用归结原理求解问题