

HW6

8.24(a-k)

Question:

8.24 用一个相容的词汇表（需要你自己定义）在一阶逻辑中表示下列语句：

- a. 某些学生在 2001 年春季学期上法语课。
- b. 上法语课的每个学生都通过了考试。
- c. 只有一个学生在 2001 年春季学期上希腊语课。
- d. 希腊语课的最好成绩总是比法语课的最好成绩高。
- e. 每个买保险的人都是聪明的。
- f. 没有人会买昂贵的保险。
- g. 有一个代理，他只卖保险给那些没有投保的人。
- h. 镇上有一个理发师，他给所有不自己刮胡子的人刮胡子。
- i. 在英国出生的人，如果其双亲都是英国公民或永久居住者，那么此人生来就是一个英国公民。
- j. 在英国以外的地方出生的人，如果其双亲生来就是英国公民，那么此人血统上是一个英国公民。
- k. 政治家可以一直愚弄某些人，也可以在某个时候愚弄所有人，但是他们无法一直愚弄所有的人。

Answer:

词汇表

Takes(x, c, s): 学生x在学期s上课程c

Pass(x, c, s): 学生x在学期s通过了课程c

Score(x, c, s): 学生x在学期s的课程c成绩

$x > y$: x比y大

F, G: 代表法语课和希腊语课

Subject(c, f)

Buy(x, y, z): x从z购买了y

Sell(x, y, z): x把y卖给z

Shave: x给y刮胡子

Born(x, c): x在c国家出生

Parent(x, y): x是y的父母

Citizen(x, c, r): x由于r原因是y国家的人

Resident(x, c): x住在c国家

Fool(x, y, t): x在时间t愚弄了y

Student(x), Person(x), Man(x), Barber(x), Expensive(x), Agent(x), Insured(x), Smart(x), Politician(x)

a.

$\exists x \text{ Student}(x) \wedge \text{Take}(x, F, \text{Spring}2001)$

b.

$$\forall x, s \text{ Student}(x) \wedge \text{Take}(x, F, s) \Rightarrow \text{Pass}(x, F, s)$$

c.

$$\exists x \text{ Student}(x) \wedge \text{Take}(x, G, \text{Spring2001}) \wedge \forall y y \neq x \Rightarrow \neg \text{Take}(y, G, \text{Spring2001})$$

d.

$$\forall s \exists y \text{ Score}(x, G, s) > \text{Score}(y, F, s)$$

e.

$$\forall x \text{ Person}(x) \wedge (\exists y, z \text{ Policy}(y) \wedge \text{Buy}(x, y, z)) \Rightarrow \text{Smart}(x)$$

f.

$$\forall x, y, z \text{ Person}(x) \wedge \text{Policy}(y) \wedge \text{Expensive}(y) \Rightarrow \neg \text{Buy}(x, y, z)$$

g.

$$\exists x \text{ Agent}(x) \wedge \forall y, z \text{ Policy}(y) \wedge \text{Sell}(x, y, z) \Rightarrow (\text{Person}(z) \wedge \neg \text{Insured}(z))$$

h.

$$\exists x \text{ Barber}(x) \wedge \forall y \text{ Man}(y) \wedge \neg \text{Shave}(y, y) \Rightarrow \text{Shave}(x, y)$$

i.

$$\forall x \text{ Person}(x) \wedge \text{Born}(x, UK) \wedge (\forall y \text{ Parent}(y, x) \Rightarrow ((\exists r \text{ Citizen}(y, UK, r)) \wedge \text{Resident}(y, UK))) \Rightarrow \text{Citizen}(x, UK, \text{Birth})$$

j.

$$\forall x \text{ Person}(x) \wedge \neg \text{Born}(x, UK) \wedge (\exists y \text{ Parent}(y, x) \wedge \text{Citizen}(y, UK, \text{Birth})) \Rightarrow \text{Citizen}(x, UK, \text{Descent})$$

k.

$$\begin{aligned} \forall x \text{ Politician}(x) \Rightarrow \\ (\exists y \forall t \text{ Person}(y) \wedge \text{Fool}(x, y, t)) \wedge \\ (\exists t \forall y \text{ Person}(y) \Rightarrow \text{Fool}(x, y, t)) \wedge \\ \neg (\forall t \forall y \text{ Person}(y) \Rightarrow \text{Fool}(x, y, t)) \end{aligned}$$

8.17

Question:

8.17 解释下面给出的 Wumpus 世界中相邻方格的定义存在什么问题:

$$\forall x, y \text{ Adjacent}([x, y], [x + 1, y]) \wedge \text{Adjacent}([x, y], [x, y + 1])$$

Answer:

没有定义

$$\text{Adjacent}([x, y], [x - 1, y]) \text{ 和 } \text{Adjacent}([x, y], [x, y - 1])$$

9.3

Question:

9.3 假定知识库中只包括一条语句: $\exists x \text{ AsHighAs}(x, \text{Everest})$ 。下列那个语句是应用存在量词实例化以后的合法结果?

a. $\text{AsHighAs}(\text{Everest}, \text{Everest})$

b. $\text{AsHighAs}(\text{Kilimanjaro}, \text{Everest})$

c. $\text{AsHighAs}(\text{Kilimanjaro}, \text{Everest}) \wedge \text{AsHighAs}(\text{BenNevis}, \text{Everest})$ (在两次应用之后)

Answer:

b,c是合法的

9.4

Question:

9.4 对于下列每对原子语句，如果存在，请给出最一般合一置换：

- a. $P(A, B, B), P(x, y, z)$
- b. $Q(y, G(A, B)), Q(G(x, x), y)$
- c. $Older(Father(y), y), Older(Father(x), John)$
- d. $Knows(Father(y), y), Knows(x, x)$

Answer:

a.

$\{x/A, y/B, z/B\}$

b.

不存在

c.

$\{x/John, y/John\}$

d.

不存在

9.6

Question:

9.6 写出下列语句的逻辑表示，使得它们适用一般化假言推理规则：

- a. 马、奶牛和猪都是哺乳动物。
- b. 一匹马的后代是马。
- c. Bluebeard 是一匹马。
- d. Bluebeard 是 Charlie 的家长。
- e. 后代和家长是逆关系。
- f. 每个哺乳动物都有一个家长。

Answer:

a.

$Horse(x) \Rightarrow Mammal(x)$

$Cow(x) \Rightarrow Mammal(x)$

$Pig(x) \Rightarrow Mammal(x)$

b.

$Offspring(x, y) \wedge Horse(y) \Rightarrow Horse(x)$

c.

$Horse(Bluebeard)$

d.

$Parent(Bluebeard, Charlie)$

e.

$Offspring(x, y) \Rightarrow Parent(y, x)$

$Parent(x, y) \Rightarrow Offspring(y, x)$

f.

$Mammal(x) \Rightarrow Parent(G(x), x)$

9.13(a,b,c)

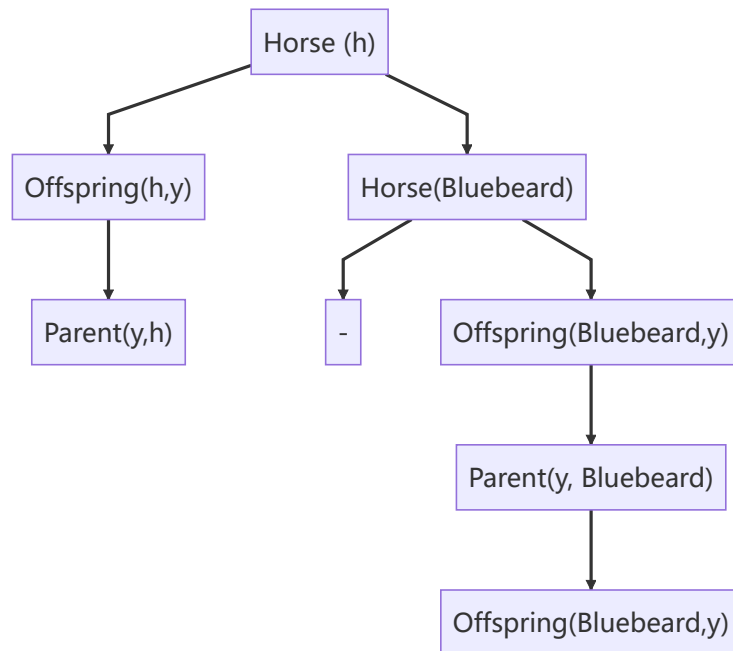
Question:

9.13 本题中需要用到你在习题 9.6 中写出的语句，运用反向链接算法来回答问题。

- 画出用穷举反向链接算法为查询 $\exists h \text{ horse}(h)$ 生成的证明树，其中子句按照给定的顺序进行匹配。
- 对于本领域，你注意到了什么？
- 实际上从你的语句中得出了多少个 h 的解？
- 你是否可以想出一种方法找出所有的解？（提示：参见 Smith 等人（1986）。）

Answer:

a.



b.

这是一个无限循环

c.

两个解

Bluebeard和Carlie都是horse