# 人工智能基础 HW6

PB18111697 王章瀚

2021 年 5 月 2 日

## 8.24

**Represent the following sentences in first-order logic, using a consistent vocabulary(which you must define):**

### a, b, c, d:

**a. Some students took French in spring 2001.**
**b. Every student who takes French passes it.**
**c. Only one student took Greek in spring 2001.**
**d. The best score in Greek is always higher than the best score in French.**

My definitions:

- $Student(x)$: $x$ is a student
- $Take(x, c, s)$: $x$ takes course $c$ in $s$
  - domain of $c$: $\{F, G\}$. $F$ means French and $G$ means Greek.
  - $s$ is a semester like "spring 2001"
- $pass(x, c, s)$: $x$ passes course $y$ in $s$
- $score(x, c, s)$: the score of student $x$ in course $c$ in semister $s$
- $a > b$: $a$ is greater than $b$

Now, abcd can be represented as:

- **a.** $\exists x(Student(x) \wedge Take(x, F, Spring\ 2001))$
- **b.** $\forall x(Student(x) \wedge \forall s\ (Take(x, G, s) \Rightarrow Pass(x, G, s)))$
- **c.** $\exists x(Student(x) \wedge Take(x, G, Spring\ 2001)) \wedge \forall y(y \neq x \wedge \neg Take(y, G, Spring\ 2001))$
- **d.** $\forall s(\exists x(score(x, G, s) > score(x, F, s)))$

## efg.

**e. Every person who buys a policy is smart.**
**f. No person buys an expensive policy.**
**g. There is an agent who sells policies only to people who are not insured.**

My definitions:

- $Person(x)$, $Agent(x)$, $Smart(x)$, $Insured(x)$: respectively, $x$ is a person, a Agent, smart, insured
- $Policy(p)$: $p$ is a policy
- $Expensive(p)$: $p$ is expensive
- $buys(x, p, s)$: $x$ buys a policy $p$ from agent $s$

Now, efg can be represented as:

- **e.** $\forall x((Person(x) \land \exists p, s(Policy(p) \land buys(x, p, s))) \Rightarrow Smart(x))$
- **f.** $\neg\exists x, p, s(Person(x) \land Policy(p) \land Agent(s) \land buys(x, p, s) \land Expensive(p))$
- **g.** $\exists s(Agent(s) \land \forall x, p(Person(x) \land Policy(p)) \Rightarrow (Person(x) \land \neg Insured(x)))$

## h.

**There is a barber who shaves all men in town who do not shave themselves.**

My definitions:

- $Barber(x)$: $x$ is a barber
- $MenInTown(x)$: $x$ is a man in town
- $Shave(x, y)$: $x$ shaves for $y$

Now, h can be represented as:

$$\exists x(Barber(x) \land \forall y(MenInTown(y) \land \neg(Shave(y, y))) \Rightarrow Shave(x, y))$$

## ij.

**i. A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.**
**j. A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.**

My definitions:

- $Person(x)$: $x$ is a person
- $BornInUK(x)$: $x$ was born in UK

- $Parent(x, y)$: $y$ is $x$'s parent
- $CitizenOfUK(x, r)$: $x$ is a UK citizen for reason $r$
- $ResidentOfUK(x)$: $x$ is a UK residnet

Now, ij can be represented as:

- **i.** $\forall x(Person(x) \land BornInUK(x) \land (\forall y(Parent(x, y) \land (\exists r(CitizenOfUK(y, r) \lor ResidentOfUK(y))))) \Rightarrow CitizenOfUK(x, birth))$
- **j.** $\forall x(Person(x) \land \neg BornInUK(x) \land (\exists y(Parent(x, y) \land CitizenOfUK(y, birth))) \Rightarrow CitizenOfUK(x, descent))$

### k.

**Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can't fool all of the people all of the time.**

My definitions:

- $Politician(x)$: $x$ is a politician
- $Person(x)$: $x$ is a person
- $fool(x, y, t)$: $x$ fools $y$ at time $t$

Now, k can be represented as:

$\neg\exists x(Politician(x) \land (\exists y \forall t Person(y) \Rightarrow fool(x, y, t)) \land (\forall y \exists t\, Person(y) \Rightarrow fool(x, y, t)) \land \neg(\forall y \forall t\, Person(y) \Rightarrow fool(x, y, t)))$

## 8.17

**Explain what is wrong with the following proposed definition of adjacent squares in the wumpus world:**

$$\forall x, y \quad Adjacent([x, y], [x+1, y]) \land Adjacent([x, y], [x, y+1])$$

- 这样的定义将无法考虑到边界情况
- 对于 $[2, 1], [1, 1]$ 将不会被考虑进来. 因此需要考虑

  $$\forall x, y \quad Adjacent([x, y], [x+1, y]) \land Adjacent([x, y], [x, y+1]) \land Adjacent([x, y], [x, y-1]) \land Adjacent([x, y], [x-1, y])$$

- 这个式子只是定义了"充分条件", 因此不能被用来推导不相邻的情况

## 9.3

**Suppose a knowledge base contains just one sentence, $\exists x AsHighAs(x, Everest)$. Which of the following are legitimate results of applying Existential Instantiation?**

**a.** $AsHighAs(Everest, Everest)$
**b.** $AsHighAs(Kilimanjaro, Everest)$
**c.** $AsHighAs(Kilimanjaro, Everest) \land AsHighAs(BenNevis, Everest)$ **(after two applications)**

Existentil Instantiation 要求替换的符号不在 KB 中出现. 因此 a 不合理, 而 b, c 均合理. 其中 b 由一次替换得到, c 由两次替换得到.

## 9.4

**For each pair of atomic sentences, give the most general unifier if it exists:**

### a.

$P(A, B, B), P(x, y, z)$

最一般合一代换为:

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

### b.

$Q(y, G(A, B)), Q(G(x, y), y$

最一般合一代换要求 $\{y/G(A, B), y/B\}$, 这是不可能的. 因此 **不存在最一般合一代换**.

### c.

$Older(Father(y), y), Older(Father(x), John)$

最一般合一代换为:

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

### d.

$Knows(Father(y), y), Knows(x, x)$

最一般合一代换要求 $x/Father(y)$, 这又要求 $y$ 能代换以 $Father(y)$. 所以 **不存在最一般合一代换**.

## 9.6

**Write down logical representations for the following sentences, suitable for use with Generalized Modus Ponens:**

### a.

**Horses, cows, and pigs are mammals.**

- $Horse(x) \Rightarrow Mammal(x)$
- $Cow(x) \Rightarrow Mammal(x)$
- $Pig(x) \Rightarrow Mammal(x)$

### b.

**An offspring of a horse is a horse.**
Define: $Offspring(x, y)$: $x$ is $y$'s offspring
Then: $Offspring(x, y) \wedge Horse(y) \Rightarrow Horse(x)$

### c.

**Bluebeard is a horse.**

- $Horse(Bluebeard)$

### d.

**Bluebeard is Charlie' s parent.**
Define: $Parent(x, y)$: $x$ is $y$'s parent
Then: $Parent(Bluebeard, Charlie)$

### e.

**Offspring and parent are inverse relations.**

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

### f.

**Every mammal has a parent.**

- $Mammal(x) \Rightarrow \exists y Parent(y, x)$

## 9.13

**In this exercise, use the sentences you wrote in Exercise 9.6 to answer a question by using a backward-chaining algorithm.**

### a.

**Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query $\exists h Horse(h)$ , where clauses are matched in the order given.**

如下图所示, 最先用的应该是 $Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$, 从而有左子树的一系列推导. 但右子树中, 由于会先用 $Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$, 因此会出现无尽的递归:

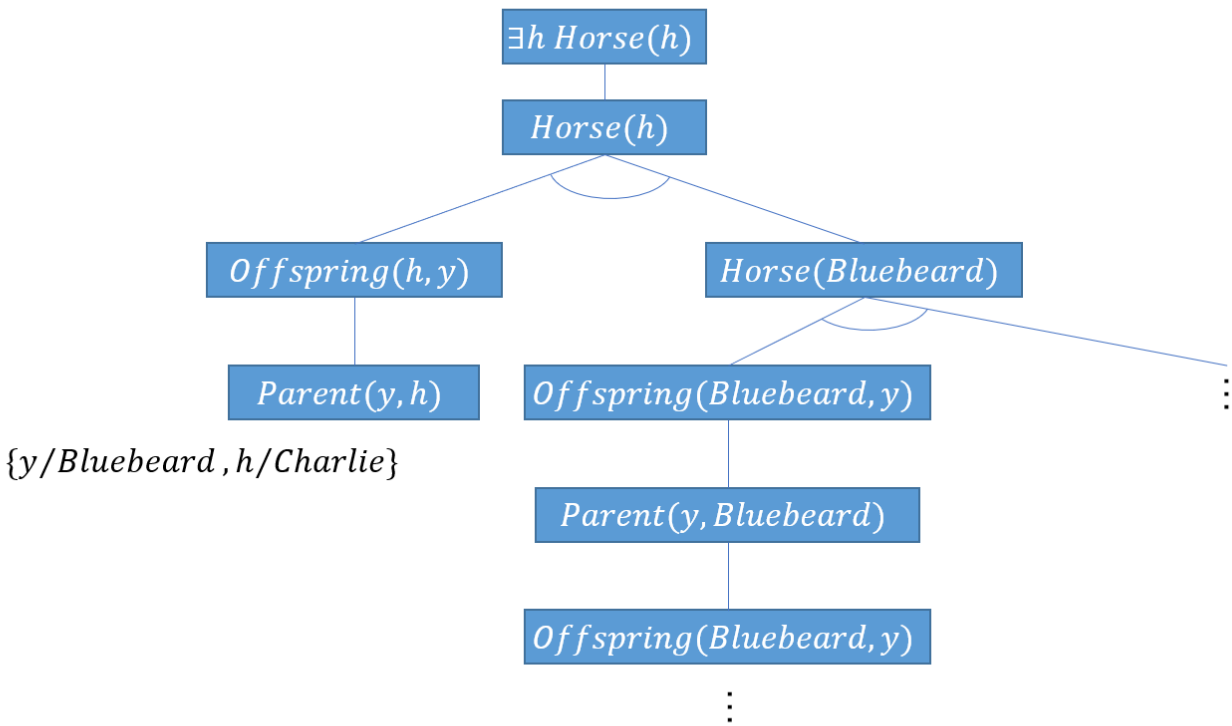


图 1: 9.13.a

### b.

**What do you notice about this domain?**

如前所述, 右子树中, 由于会先用 $Offspring(x,y) \wedge Horse(y) \Rightarrow Horse(x)$, 因此会出现无尽的递归.

### c.

**How many solutions for $h$ actually follow from your sentences?**

只有一个, 即 $Charlie$ 作为 $h$.