AI-hw6

PB21081601 张芷苒

- 8.24 (a-k)， 8.17
- 9.3, 9.4, 9.6, 9.13(a,b,c)

> 8.17 Explain what is wrong with the following proposed definition of adjacent squares in the wumpus world

The proposed definition of adjacent squares in the wumpus world:

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

is flawed due to several reasons:

1. **Logical Structure**: Using the conjunction operator ($\wedge$) incorrectly suggests that $[x+1, y]$ and $[x, y+1]$ must both be adjacent to $[x, y]$ at all times, which is not necessary for defining adjacency.
2. **Incomplete Definition**: The definition only includes right and upward adjacencies, omitting left $([x-1, y])$ and downward $([x, y-1])$ adjacencies.
3. **No Boundary Considerations**: It fails to account for grid boundaries where adjacent squares might not exist.

Better comprehensive definition would include all possible adjacent directions and consider grid boundaries.

> 8.24 Represent the following sentences in first-order logic, using a consistent vocabulary(which you must define):
> 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.
> 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.
> h. There is a barber who shaves all men in town who do not shave themselves.
> 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.
> 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.

To represent the given sentences in first-order logic, need to establish a consistent vocabulary:

- **Student(s)**: A person who is studying.
- **Course(c, t)**: A person `c` takes a course `t`.
- **Passes(c, t)**: A person `c` passes a course `t`.
- **Score(c, t, score)**: A person `c` gets a score `score` in a course `t`.
- **Policy(p)**: Represents an insurance policy `p`.
- **Buys(c, p)**: A person `c` buys a policy `p`.
- **Expensive(p)**: A policy `p` is expensive.
- **Sells(a, c, p)**: An agent `a` sells policy `p` to customer `c`.
- **Insured(c)**: A person `c` is insured.
- **Shaves(b, c)**: A barber `b` shaves a person `c`.
- **Man(c)**: A person `c` is a man.
- **ParentOf(p, c)**: `p` is a parent of `c`.
- **BornIn(c, place)**: A person `c` is born in `place`.
- **CitizenOf(c, country)**: A person `c` is a citizen of `country`.
- **ResidentOf(c, country)**: A person `c` is a resident of `country`.
- **Fool(p, t)**: A politician `p` fools people at time `t`.

translate each sentence into first-order logic:

a. **Some students took French in spring 2001.**

$$\exists s\, \text{Course}(s, \text{"French, Spring 2001"})$$

b. **Every student who takes French passes it.**

$$\forall s\, (\text{Course}(s, \text{"French"}) \rightarrow \text{Passes}(s, \text{"French"}))$$

c. **Only one student took Greek in spring 2001.**

$$\exists! s\, \text{Course}(s, \text{"Greek, Spring 2001"})$$

d. **The best score in Greek is always higher than the best score in French.**

$$\forall s, t\, (\text{Score}(s, \text{"Greek"}, x) \wedge \text{Score}(t, \text{"French"}, y) \rightarrow x > y)$$

e. **Every person who buys a policy is smart.**

$$\forall p\, (\text{Buys}(p, \_) \rightarrow \text{Smart}(p))$$

f. **No person buys an expensive policy.**

$$\forall p\, \neg(\text{Buys}(p, q) \wedge \text{Expensive}(q))$$

g. **There is an agent who sells policies only to people who are not insured.**

$$\exists a\, \forall c, p\, (\text{Sells}(a, c, p) \rightarrow \neg\text{Insured}(c))$$

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

$$\exists b \, \forall c \, (\text{Man}(c) \wedge \neg \text{Shaves}(c, c) \rightarrow \text{Shaves}(b, c))$$

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.**

$$\forall c \, (\text{BornIn}(c, \text{"UK"}) \wedge \forall p \, (\text{ParentOf}(p, c) \rightarrow (\text{CitizenOf}(p, \text{"UK"}) \vee \text{ResidentOf}(p, \text{"UK"}))) \rightarrow \text{Citiz}$$

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

$$\forall c \, (\neg \text{BornIn}(c, \text{"UK"}) \wedge \exists p \, (\text{ParentOf}(p, c) \wedge \text{CitizenOf}(p, \text{"UK"}) \wedge \text{BornIn}(p, \text{"UK"})) \rightarrow \text{CitizenOf}($$

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.**

$$\exists p \, \forall t \, \exists c \, \text{Fool}(p, t) \wedge \exists t \, \forall c \, \text{Fool}(p, t) \wedge \neg \forall t \, \forall c \, \text{Fool}(p, t)$$

> 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) AsHighAs(BenNevis, Everest) (after two applications).

The legitimate results of applying Existential Instantiation to the sentence $\exists x \text{AsHighAs}(x, \text{Everest})$ are:

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

All these options correctly instantiate (x) with specific constants, which is allowed in Existential Instantiation.

> 9.4 For each pair of atomic sentences, give the most general unifier if it exists:
> a. $P(A, B, B), P(x, y, z)$.
> b. $Q(y, G(A, B)), Q(G(x, x), y)$.
> c. Older $(Father(y), y), \text{Older}(Father(x), John)$.
> d. $\text{Knows}($ Father $(y), y), \text{Knows}(x, x)$.

For each pair of atomic sentences, here are the most general unifiers (MGUs) if they exist:

a. $P(A, B, B), P(x, y, z)$:

- MGU: $\{x \mapsto A, y \mapsto B, z \mapsto B\}$

b. $Q(y, G(A, B)), Q(G(x, x), y)$:

- MGU: $\{x \mapsto A, y \mapsto G(A, A)\}$

c. $\text{Older}(\text{Father}(y), y), \text{Older}(\text{Father}(x), \text{John})$:

- MGU: $\{x \mapsto \text{John}, y \mapsto \text{John}\}$

d. $\text{Knows}(\text{Father}(y), y), \text{Knows}(x, x)$:

- No unifier exists because no substitution can make $\text{Father}(y)$ equal to y, which would be required to match the pattern of $\text{Knows}(x, x)$.

> 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.
> b. What do you notice about this domain?
> c. How many solutions for h actually follow from your sentences?

a. **Proof Tree for the Query** $\exists h \text{Horse}(h)$:

- The proof tree starts with the query $\exists h \text{Horse}(h)$ and looks for a fact or a universally quantified statement in the knowledge base that can instantiate (h). The proof tree stops at a clause that instantiates (h) with a specific example or uses a universal fact about horses.

b. **Observation about the Domain**:

- The domain likely includes straightforward definitions or assertions involving horses, focusing on their existence, attributes, or activities. This simplicity suggests the domain is tailored for demonstrating logical deduction, particularly existential queries.

c. **Number of Solutions for (h) That Follow From the Sentences**:

- The number of solutions for (h) would depend on the specific instances or applicable rules about horses mentioned in the knowledge base. If general statements or specific examples like "Shadowfax is a horse" exist, each instance provides a distinct solution. If the knowledge base contains a general statement asserting the existence of horses, it implies at least one solution, but without more specifics, only one solution can be conclusively confirmed from the query alone.