

CS383 Programming Languages

Quiz 13

1. Which is **not** right with logical language?

- a. There are two strategies to search rules: backward reasoning and forward reasoning.
- b. Logic programming can not compute values.**
- c. When a goal matches the conclusion of more than one rule, it will use backtracking to search.
- d. The order of subgoal evaluation has a significant impact on the computation

2. What is **not** one of the terms in Prolog?

a. Variables

b. Constants

c. Structures

d. None of the above

3. Which of the following is **not** a valid variable in Prolog?

a. Bob

b. Alice

thing

c. Jim

d. monkey

4. The following Prolog program has been consulted:
`a(B,B).`

Determine the output of the following query:

`-? a(X,Y).`

a. true

b. false

c. $X=Y$

d. $X=B, Y=B$

8. The following Prolog program has been consulted:
`a(B,B).`

Determine the output of the following query:

`-? a(x,y).`

a. true

b. false

c. `x=y`

d. `x=B,y=B`

5. In Prolog the following statement is called a?

a. Fact

b. Rule

c. Constant

d. None of above

Sunny(Shanghai).

6. What is the most possible relationship that the following Prolog program represent?

a. X is the father of Y

b. X is the brother of Y

c. Y is the sister of X

d. X and Y are NOT related.

```
r(X, Y) :-  
    parent(Z,X).  
    parent(Z,Y).  
    male(X).  
    X \= Y.
```



```
7.  like(amy, bob).  
    like(bob, ceph).  
    hate(X,Z):-like(X,Y),like(Y,Z).
```

What is result of hate(Who, ceph) in Prolog?

Who=amy

9. Which of the following is false for the programming language Prolog?

- a. A Prolog variable can only be assigned to a value once
- b. Prolog is a Strongly Typed Language
- c. The scope of a variable in Prolog is a single clause or rule
- d. There are some predefined predicates in Prolog

10. The clausal form of the disjunctive normal form $\neg A \vee \neg B \vee \neg C \vee D$ is ?

$$A \Rightarrow B = \neg A \vee B$$

$$\begin{aligned} & \neg A \vee \neg B \vee \neg C \vee D \\ & \rightarrow (\neg A \vee \neg B \vee \neg C) \vee D \\ & \rightarrow \neg(A \wedge B \wedge C) \vee D \\ & \rightarrow A \wedge B \wedge C \Rightarrow D \end{aligned}$$

- a. $A \wedge B \wedge C \Rightarrow D$
- b. $A \vee B \vee C \vee D \Rightarrow \text{True}$
- c. $A \wedge B \wedge C \wedge D \Rightarrow \text{True}$
- d. $A \wedge B \wedge C \wedge D \Rightarrow \text{False}$

Horn Clauses and Predicates

- Any Horn clause

$$h \leftarrow p_1, p_2, \dots, p_n$$

can be written as a predicate:

$$p_1 \wedge p_2 \wedge \dots \wedge p_n \supset h$$

or equivalently:

$$\neg(p_1 \wedge p_2 \wedge \dots \wedge p_n) \vee h$$

- But not every predicate can be written as a Horn clause, such as *disjunctions*:
- E.g., $iterate(x) \supset reads(x) \vee writes(x)$