

Cut and Not Example Test Questions

1.

What is the purpose of the cut in Prolog? Explain by describing three different ways in which cut is used.

2.

What is the purpose of using the not predicate? Explain by describing the general case and a specific example.

3.

It is asserted that cut and not are logically equivalent in the following way.

$$\begin{array}{lcl} A & :- & B, !, C. \\ A & :- & D. \end{array} \quad \equiv \quad \begin{array}{lcl} A & :- & B, C. \\ A & :- & \text{not} (B) , D. \end{array}$$

Is the equivalence true or is it false? You must explain your answer. Just stating true or false has no value.

4.

Consider the following Prolog code.

```
pair(bert, ernie).
pair(bigbird, cookiemonster).
pair(bert, elmo).
pair(cookiemonster, tina).
pair(tina, bert).
single(oscar).
```

- A. What is the first result of the query `pair(A,B), pair(B,C), not(single(A)).`
- B. What is the first result of the query `pair(A,B), pair(B,C), !, not(single(A)).`
- C. What is the first result of the query `not(single(A)), pair(A,B), pair(B,C).`

5.

Consider the following Prolog program.

```
w(11).
w(22).
test1(X) :- w(X), fail; w(X).
test1(X) :- w(X).

test2(X) :- w(X), !, fail ; w(X).
test2(X) :- w(X).

test3(X) :- !, w(X), fail ; w(X).
test3(X) :- w(X).

test4(X) :- !, ( w(X), fail ; w(X) ).
test4(X) :- w(X).
```

- A Give all possible answers to the query `test1(X)`, in the order that Prolog would produce them.
- B Give all possible answers to the query `test2(X)`, in the order that Prolog would produce them.
- C Give all possible answers to the query `test3(X)`, in the order that Prolog would produce them.

D Give all possible answers to the query `test4(X)`, in the order that Prolog would produce them.

6.

Consider the following Prolog program.

```
a(X) :- b(X).  
a(X) :- f(X).  
b(X) :- g(X) , v(X).  
b(X) :- X=44 , v(X).  
g(11).  
G(3).  
V(X).  
F(5).
```

A Give all possible answers to the query `a(X)`, in the order that Prolog would produce them.

B Now consider the same Prolog program but with the first rule for `b` **replaced** by the following.

```
B(X) :- g(X) , ! , v(X).
```

Give all possible answers to the query `a(X)`, in the order that Prolog would produce them.

7.

The `not` predicate can be represented in Prolog as follows.

```
not(P) :- call(P) , ! , fail.  
not(_).
```

Show how `cut` can be represented with `not`.