Lab 2 – 08226 Artificial Intelligence

This lab tutorial will introduce you to arithmetic and to using variables in your Prolog programs and in queries.

1.0 Arithmetic

Start SWI-Prolog-Editor from the Windows Menu system and create a new Prolog file called **Lab2.pl** and store it in **G:/08226/Lab 2/**.

You can use Prolog to compare some arithmetic expressions.

Ask the following query:

```
?- 2=2.
```

You should now be presented with the answer **true**. Clearly 2 does equal 2 so we would expect this.

Ask the following query:

```
?- two=two.
```

You should now be presented with the answer **true**. Prolog is not doing any arithmetic calculation when we use the **=** sign. It is comparing the left hand side of the equal sign to the right hand side of the equal sign, and returns true if they are identical. To confirm this, ask the following query:

```
?- rhubarb=rhubarb.
```

Ask the following query:

```
?- 2 is 2.
```

You should now be presented with the answer **true**. The **predicate** (keyword) **is** is used here to actually calculate the right hand side of the **predicate**, and then compares it to the left hand side of the **predicate** to see if they are equivalent.

1.1 Arithmetic Test Exercise

Write down your answers to the following questions without using Prolog, and then check your answers at the end of this document and/or using Prolog.

Given the following Prolog queries, which value will the queries return?

1) `?- 4=2+2.` ***true / false***
2) `?- 4 is 2+2.` ***true / false***
3) `?- (5+2+3)*2 is 20.` ***true / false***
4) `?- 20 is (5+2+3)*2.` ***true / false***
5) `?- X=10, X=X+1.` ***true / false / X=11***
6) `?- X=10, X is X+1.` ***true / false / X=11***

Check your answers here: **Arithmetic Test Exercise Answers**

2.0 Variables with Single Facts

In Lab 1 we simply had facts which we then queried by stating the facts that we knew back to Prolog. What we are going to do now is use variables to retrieve information that we may not know.

Enter the following code into your program and then select **consult**:

```
red(ferrari).

car(ferrari).

car(mazda).
```

We are going to ask Prolog what objects it has that are red in its database. We are going to do this by using a variable which we will call **`What`**. Note that a variable must begin with a capital-letter.

Ask the following query:

```
?- red(What).
```

You should now be presented with the answer:

```
What = ferrari
```

As a result of this query, the variable **`What`** has matched (or **unified** in Prolog terminology) with **ferrari** . We also say that the variable **`What`** has a **binding** with **ferrari**.

3.0 Variables with Multiple Facts

We are going to ask Prolog what cars it has in its database. We are going to do this by using a variable which we will call **`What_cars`**. Note that a variable must begin with a capital-letter.

Ask the following query:

```
?- car(What_cars).
```

You should be presented with an answer but note that the prompt has not yet appeared which means that there are more responses from Prolog to come, so press the Enter key until the prompt appears.

You should now be presented with the answers:

```
What_cars = ferrari

What_cars = mazda
```

3.1 Variables Example

Enter the following code into your program and then select **consult**:

```
food(apple,fruit).        /* apple is a fruit */

food(tomato,fruit).       /* tomato is a fruit */

food(lettuce,salad).      /* lettuce is a salad */

food(beef,meat).          /* beef is a meat */
```

The above facts are describing food items by their name and the food type.

We are going to ask Prolog what foods it has that are classified as salad. Ask the following query:

```
?- food(Food_name,salad).
```

You should now be presented with the answer:

```
Food_name = lettuce
```

[remember since we have more than one fact for food then we need to keep pressing the Return key until the prompt appears]

We are now going to ask Prolog what foods it items it has. Ask the following query:

```
?- food(Food_name,Food_type).
```

You should now be presented with the answers:

```
Food_name = apple,

Food_type = fruit ;

Food_name = tomato,

Food_type = fruit ;

Food_name = lettuce,

Food_type = salad ;

Food_name = beef,

Food_type = meat.
```

3.1.1 Variables Test Exercise

Write down your answers to the following questions without using Prolog, and then check your answers at the end of this document and/or using Prolog.

Given the following Prolog program, which value will the queries return?

```
likes(darren,beer).
likes(steve,running).
likes(steve,programming).
likes(paul,football).
likes(dawn,shopping).
likes(paul,programming).
```

7) `?- likes(darren,What).` ***What = beer / What = darren***

8) `?- likes(paul,What).` ***What = football,programming / What = football***

9) `?- likes(Who,Shopping).` ***Who = dawn / other result***

10) `?- likes(Who,running).` ***Who = steve / other result***

11) `?- likes(Who,programming).` ***Who = steve,paul / other result***

Check your answers here: **Variables Test Exercise Answers**

4.0 Arithmetic Test Exercise Answers

Given the following Prolog queries, which value will the queries return?

1) `?- 4=2+2.` ***false*** *– = sign only returns true if both sides are identical*
2) `?- 4 is 2+2.` ***true***
3) `?- (5+2+3)*2 is 20.` ***false*** *– only the right hand side is evaluated*
4) `?- 20 is (5+2+3)*2.` ***true***
5) `?- X=10, X=X+1.` ***false*** *– Prolog substitutes the value X=10 directly into the X=X+1 statement, therefore it does the comparison of 10=10+1 which is clearly false. The same result would be found with the query ?- X=10, X is X+1*
6) `?- X=10, X is X+1.` ***false*** *– The same result as with X=X+1.*

Check the answers using Prolog.

Return to your exercise here: **Arithmetic Test Exercise**

5.0 Variables Test Exercise Answers

Given the following Prolog program, which value will the queries return?

```
likes(darren,beer).
likes(steve,running).
likes(steve,programming).
likes(paul,football).
likes(dawn,shopping).
likes(paul,programming).
```

1)	`?- likes(darren,What).`	***What = beer***
2)	`?- likes(paul,What).`	***What = football;programming***
3)	`?- likes(Who,Shopping).`	***Other result*** *– lists all of the facts data*
4)	`?- likes(Who,running).`	***Who = steve***
5)	`?- likes(Who,programming).`	***Who = steve;paul***

Check the answers using Prolog.

Return to your exercise here: **Variables Test Exercise**