@edgar resources

ECII/ECSI3206: ARTIFITIAL INTELLIGENCE [prolog Sample practices]

Hello world example:
write('helo world'),nl.

Variable assignment:
X = elephant, write(X), nl.

Checking the type of a prolog term:
atom(elephant).

Check for X,Y pairs:
Functor(X,Y)

Simple Arithmetic operations:
is(X,1+2).

Check if an item exist within a list:
member(dog, [elephant, horse, donkey, dog, monkey]).

Check the length of a list:
length([elephant, [], [1, 2, 3, 4]], Length).

Select an item form a list and omit it:
select(bird, [mouse, bird, jellyfish, zebra], X).

Reversing list order:
reverse([1, 2, 3, 4, 5], X).

-Write a predicate replace/4 to replace all occurrences of a given element (second argument) by another given element (third argument) in a given list (first argument)
replace([1, 2, 3, 4, 3, 5, 6, 3], 3, x, List).

Family exercise:

```
female(mary).
female(sandra).
female(juliet).
female(lisa).
male(peter).
male(paul).
male(dick).
male(bob).
male(harry).
parent(bob, lisa).
parent(bob, paul).
parent(bob, mary).
parent(juliet, lisa).
parent(juliet, paul).
parent(juliet, mary).
parent(peter, harry).
parent(lisa, harry).
parent(mary, dick).
parent(mary, sandra).
```

-create new predicate rules using unification that can be used to check if one is a son of a particular parent.:

```
son(X,Y):-parent(Y,X),male(X).
```

-create new predicate rules using unification that can be used to check if one is a daughter of a particular parent:

```
daughter(X,Y):-parent(Y,X),female(X).
```

-write a query to check son pairs

```
son(X,Y).
```

-Write query to check daughter pairs:

```
daughter(X,Y).
```

-Get all the children of bob:

```
parent(bob,X).
```

-Get all sons of Bob(brothers):

```
son(X,bob).
```

-Get all daughters of bob(sisters):

```
daughter(X,bob).
```

-Get all daughters and sons of bob(children):

```
daughter(X,bob);son(X,bob).
```