

Solution Logic Programming

Exercise 1

We will build a genealogy that covers relations in a family. Consider a genealogy database consisting of the following predicates (**as defined during lecture hours**):

```
female(X), male(X), parent(X,Y),  
mother(X,Y), father(X,Y),  
sister(X,Y), brother(X,Y)
```

Define rules allowing you to determine the following relations:

```
grandfather(X,Y), grandmother(X,Y), grandparent(X,Y),  
son(X,Y), daughter(X,Y), child(X,Y),  
grandson(X,Y), granddaughter(X,Y), grandchild(X,Y)
```

Answer:

```
% GENEALOGY DATABASE  
  
female(anne).  
female(diana).  
female(elizabeth).  
female(kate).  
female(charlotte).  
  
male(andrew).  
male(charles).  
male(edward).  
male(harry).  
male(philip).  
male(william).  
male(george).  
male(loius).  
  
parent(andrew, elizabeth).  
parent(andrew, philip).  
parent(anne, elizabeth).  
parent(anne, philip).  
parent(charles, elizabeth).  
parent(charles, philip).  
parent(edward, elizabeth).  
parent(edward, philip).  
parent(harry, charles).  
parent(harry, diana).  
parent(william, charles).  
parent(william, diana).  
parent(george, william).  
parent(george, kate).  
parent(charlotte, william).  
parent(charlotte, kate).  
parent(louis, william).  
parent(louis, kate).  
  
mother(X, M) :- parent(X,M), female(M).
```

```
father(X, M) :- parent(X,M), male(M).

% FURTHER RELATIONS

grandfather(X, G) :- parent(X, P), parent(P, G), male(G).
grandmother(X, G) :- parent(X, P), parent(P, G), female(G).
grandparent(X, G) :- grandfather(X, G).
grandparent(X, G) :- grandmother(X, G).

son(X, S) :- parent(S, X), male(S).
daughter(X, D) :- parent(D, X), female(D).
child(X, C) :- parent(C, X).

grandson(X, S) :- grandparent(S, X), male(S).
granddaughter(X, D) :- grandparent(D, X), female(D).
grandchild(X, C) :- grandparent(C, X).
```

Exercise 2

Define the following predicates to determine if a list:

- a. has an even number of elements
- b. is a palindrome (i.e. it reads the same from left to right as it does from right to left).

Answer:

```
evenNumber([]).
evenNumber(_|R) :- \+ evenNumber(R).

lastElement([X],X).
lastElement(_|R,X) :- lastElement(R,X).

withoutLast([],[]).
withoutLast([X|R], [X|Y]) :- withoutLast(R,Y).

isPalindrom([]).
isPalindrom(_).
isPalindrom([X|R]) :- lastElement(R,X), withoutLast(R,RR), isPalindrom(RR).
```