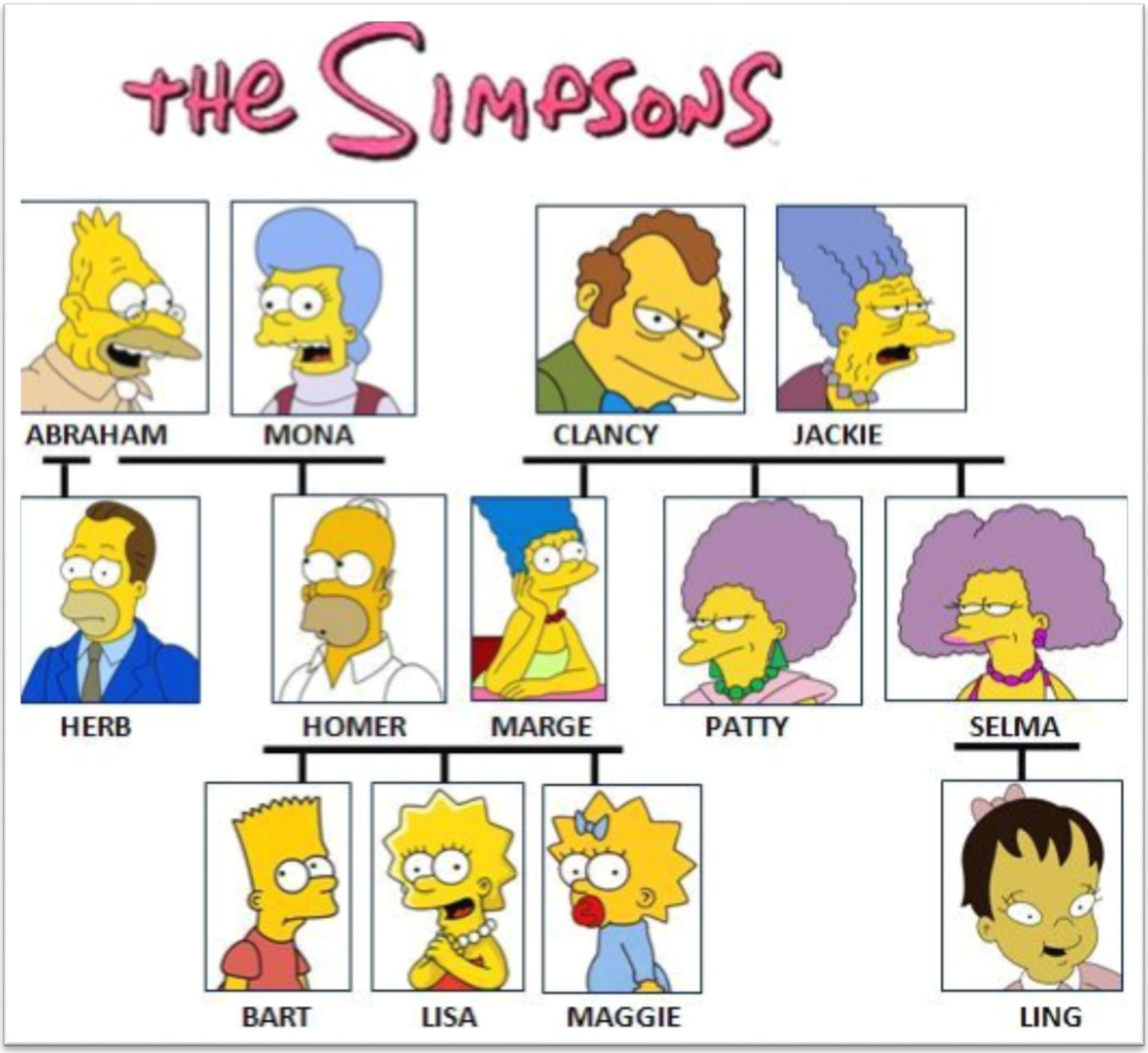
# CS 360 Lab 5 – Prolog

Lab Created by Lucas Cordova

## Part One:

### Problem 1: Product a Prolog Knowledge Base

Use the predicates male/1, female/1, and the parent_of/2 to represent the following Simpson's family tree as a Prolog knowledge base.



## Problem 2: Formulate Relationships

Formulate rules to capture the following relationships:

1. father_of(Father,Child) and mother_of(Mother,Child)
2. grandfather_of(Grandfather,Child) and grandmother_of(Grandmother,Child)
3. sister_of(Sister,Person)
4. aunt_of(Aunt,Person) and uncle_of(Uncle,Person)

## Problem 3: Queries

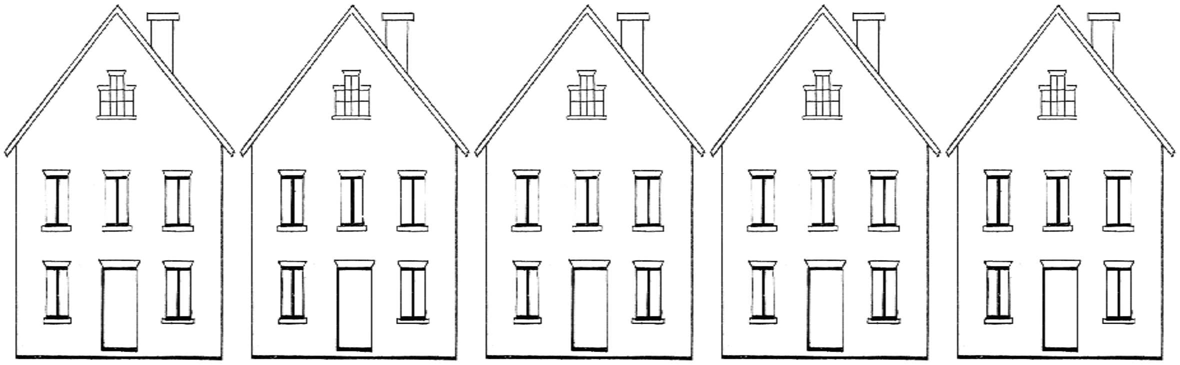
Produce the following queries that respond with the answer.

1. Who are Bart’s grandmothers?
2. List a Person’s grandchildren.
3. List “Person X’s” aunts.
4. List “Person X’s” grandparents.
5. List “Person X’s” siblings.

# Part Two:

## Prolog Owl Puzzle

Puzzles make great problems to solve with Prolog. The following puzzle is adapted from the classic Zebra problem published in Life International, December 1962.



There are 5 alums from Oregon that moved to the same street. Each alum has a different preference for what the color their house was painted, what they eat for dessert, what they drink, and what type of pet they have. Your objective is to find out which Oregon alum owns the owl.

## Knowledgebase

The following facts comprise the knowledge base.

1. There are five houses on the street.
2. The OSU alum lives in the red house.
3. The OIT alum owns the dog.
4. Coffee is drunk in the green house.
5. The UofO alum drinks tea.
6. The green house is immediately to the right of the ivory house.
7. The cookies eater owns snails.
8. Twinkies are eaten in the yellow house.
9. Milk is drunk in the middle house.
10. The PSU alum lives in the first house.
11. The alum who eats pie lives in the house next to the alum with the fox.
12. Twinkies are eaten in the house next to the house where the horse is kept.
13. The alum who eats ice cream drinks orange juice.
14. The WOU alum eats cheesecake.
15. The PSU alum lives next to the blue house.

## house(C, A, P, D, S)

We first define the house(C, A, P, D, S) predicate which signifies the following information.

C: the color of the house

A: the alma mater of the alum

P: the alum’s pet

D: the alum's drink

S: the alum's signature dessert

With the aid of the anonymous variable '_', we can represent the fact

> "The Oregon State alum lives in the red house." as a Prolog statement:
>
> > house(red, osu_alum, _, _, _).

The following predicates are provided to aid you in the definition of the knowledgebase.

## next_to(X, Y, List)

> The next_to(X, Y, List) rule is true if the elements X and Y are adjacent in the List.

```
next_to(X, Y, List) :- is_right(X, Y, List).
next_to(X, Y, List) :- is_right(Y, X, List).
```

## is_right(L, R, [L | [R | _]])

is_right(L, R, [L | [R | _]]) is true if the element L comes first in the list and the element R is at the second position.

```
is_right(L, R, [L | [R | _]]).
is_right(L, R, [_ | Rest]) :- is_right(L, R, Rest).
```

## Program Start & Hints

```
owns_owl(Street, Who) :-


% There are five houses on the street.
Street = [_House1, _House2, _House3, _House4, _House5],

% The OSU alum lives in the red house.
member(house(red, osu_alum, _, _, _), Street),


...

...


% the green house is immediately to the right of the ivory house
is_right(house(green,_,_,_,_),
house(ivory,_,_,_,_), Street),


...

...


% Milk is drunk in the middle house.
[_, _, house(_, _, _, milk, _), _, _] = Street,
```

...

```
% the final statement
member(house(_, Who, owl, _, _), Street).
```

## Execution

```
? owns_owl(Street, Who).
```

```
% A list with the street population will output followed by alum that owns the owl
```

```
Who = xxxxxxx_alum
```