Exercise 2.2

October 5, 2024

In this exercise we will go over all of the different versions of the program that we are able to crate using the original unalter version. For will for each of these variations give 1) a brief explaination what happens in the program and 2) why it happens.

For this first version all things work the way that you would expect them, the middle(a, X) gives all the lists which contain the letter "a" as the middle of the list. The middle(X, [a,b,c]) will gives the answer that X is equal to the letter b.

Listing 1: Unaltered middle predicate

```
% middle(X,Xs)
% X is the middle element in the list Xs
middle(X, [X]).
middle(X, [First|Xs]) :-
    append(Middle, [Last], Xs),
    middle(X, Middle).
```

For this version of the program the recursive predicate comes first then the base case. This means that the query middle(a, X) will skip the base case of $X = [a]$ and will instead go directly to $X = [\_, a, \_]$. The resonse for this is because it will try to append first then conclude that there is no substitution that will not lead to a finit failure using the recursive case. Only then will another substitution be found to unify using the base case which will then succeed The other query of interest still gives the expected answer.

Listing 2: Altered version 1

```
middle(X, [First|Xs]) :-
    append(Middle, [Last], Xs),
```

```
3       middle(X, Middle).
4   middle(X, [X]).
```

For this particular version if you execute the query middle(a,X) you will get an overflow error. This is because prolog will look at the order of the program clauses in order to determin which one to try and unify first. As such prolog will expand the recursive case over and over never being able to get to the base case. The other query that we are using (middle(X, [a,b,c]) will also get an overflow error for the same reason.

Listing 3: Altered version 2

```
1   middle(X, [First|Xs]) :-
2       middle(X, Middle),
3       append(Middle, [Last], Xs).
4   middle(X, [X]).
```

For this last version we also get the kind of results that we expect. The first query middle(a,X) we get the all the lists that have the letter "a" as the center element, including the list with just "a". For the second query middle(X, [a,b,c]) we also get the expected results of the letter "b". However, after that answer is produced if we try and get another answer we get an overflow error as we will always unify with the recursive case over and over as we are unable to unify with the base case.

Listing 4: Altered version 3

```
1   middle(X, [X]).
2   middle(X, [First|Xs]) :-
3       middle(X, Middle),
4       append(Middle, [Last], Xs).
```