Wentworth Institute of Technology
COMP3071 – Intro To Programming Languages
Fall 2016, Kreimendahl

Lab Assignment 2

This assignment is intended to work with a larger codebase in Prolog and apply knowledge of a declarative language to a problem that may occur in the real world.

1. Repository

First, fork the repository at `https://gitlab.com/comp3071fk/la4` and use git to clone the repository to your local machine. Modify the `la4.pl` file in the repository and when you are ready to submit, use git to add, commit, and push the file. Make sure that your submitted code is all in `la4.pl` in your forked version of `LA4`, not a different file or project. Add me to the project on gitlab as a member with developer access so I have access to your code.

2. Specifications

The purpose of this assignment is to implement several predicates related to arithmetic and lists in Prolog. Read each description carefully in order to implement the correct rules.

2.1. Language

This assignment should be completed in Prolog. Please add sufficient comments to your code so that it's clear to a smart but uninformed reader what your intention is with a code block.

2.2. Predicates

Define the following predicates by writing a rule or rules for them. The examples are just that – examples. You will have to write some facts to test your predicates correctly and completely.

brotherInLaw/2

Using predicates `married/2`, `sibling/2`, and `male/1`, write a rule `brotherInLaw/2` to tell if someone is a brother-in-law. A person is your brother in law either if they are a brother of someone you are married to, or if they are male and married to one of your siblings. See the following knowledge base for argument ordering:

```
sibling(jeff, tina). married(jeff, mark). male(mark).
```

The query `brotherInLaw(mark, tina).` should return `true`.

uniques/2

Write a `uniques/2` rule that finds all of the unique elements in a list. The first argument should be an input list and the second argument is the return value. The query `uniques([4, 2, ana, 5, 2, cat, ana], L).` should result in `L = [4, 5, cat]`.

holes/2

The `holes/2` rule should have an input as the first argument that is a list of integers, and an output that is all of the holes in the list. The built-in predicate `sort/2` will probably be very handy. As some examples, `holes([1, 3, 5], X).` should `yield X = [2, 4].` `holes([5, 1], Y).` should be `Y = [2, 3, 4].`

2.3. Testing

In order for you to test your rules, you will need to write some of your own facts and queries. I will be testing your submission on queries similar to the examples provided in this text, as well as some boundary conditions.