CISC 360 Assignment 4
due Tuesday, 2021–11–16 at 11:59pm, via onQ

Jana Dunfield

November 8, 2021

Reminder: All work submitted must be your own, or, if you are working with one other student, your teammate's.

Late policy: Assignments submitted up to 24 hours late (that is, by 11:59 pm the following day) will be accepted **without penalty**. Assignments submitted more than 24 hours late will **not** be accepted, except with an accommodation or a consideration granted according to policy.

If you choose to work in a group of 2

You **must** use version control (such as GitHub, GitLab, Bitbucket, etc.). This is primarily to help you maintain an equitable distribution of work, because commit logs provide information about the members' level of contribution.

Your repository **must** be private—otherwise, anyone who has your GitHub (etc.) username can copy your code, which would violate academic integrity. However, upon request from the course staff, you must give us access to your repository. (But you do not need to give us access unless we ask.)

We only need *one* submission of the assignment ("Assignment 4"). However, each of you *must* submit a brief statement ("Assignment 4 Group Statements"):

1. Estimate the number of hours you spent on the assignment.
2. Briefly describe your contribution, and your teammate's contribution. (Coding, trying to understand the assignment, testing, etc.)

This is meant to ensure that both group members reflect on their relative contributions.

If you do not submit a statement, you will not receive an assignment mark. This is meant to ensure that each group member is at least involved enough to submit a statement. **Each** member must submit a statement. That is, you must make two separate submissions in "Assignment 4 Group Statements".

IMPORTANT: Your file must compile

Your file **must** load (`consult` in SWI-Prolog) successfully, or we will subtract **30%** from your mark.

If you are halfway through a problem and run out of time, **comment out the problematic code** by surrounding it with `/*...*/` and add a comment describing what you were trying to do. We generally give (partial) marks for evidence of progress in solving a problem, but **we need the file to load**.

It is your responsibility to submit the right version of the file.

(Warnings about singleton variables do **not** prevent the file from loading.)

1 Add your student ID

Begin by adding your student ID number in `a4.pl`, after `student_id(`, replacing "`this is a syntax error`".

```
/*
 * Q1: Student ID
 */
student_id( this is a syntax error ).
% other_student_id(  ).
% if in a group, uncomment the above line and put the other student ID here
```

2 Q2: Prime numbers

The file `a4.pl` contains some of the code necessary to compute prime numbers:

- the definition of a relation (predicate) `isPrime`, which "returns" 'prime' if a natural number is prime (for example, `isPrime(7, pr)`), and 'composite' otherwise (for example, `isPrime(8, co)`);
- the definition of a relation (predicate) `hasFactor`, which is true if the first argument is divisible by the second (that is, the second argument is a factor of the first argument).

To finish the job, you need to define a relation `findPrimes`:

```
/*
  findPrimes(Numbers, Primes)
    Primes = all prime numbers in Numbers

  Q2a. Replace the word "change_this" in the rules below.
       Hint:  Try to use  findPrimes(Xs, Ys).
*/
findPrimes([], []).

% In this rule, we include X in the output: [X | Ys].
% So this rule should check that X is prime.
findPrimes([X | Xs], [X | Ys]) :-
  change_this.

% In this rule, we do not include X in the output.
% So this rule should check that X is composite.
findPrimes([X | Xs], Ys) :-
  change_this.
```

In the second part of this question, use `upto` (defined in `a4.pl`) and `findPrimes` to define a relation that finds all the prime numbers in a certain range.

```
% primes_range(M, N, Primes)
%   Primes = all prime numbers between M and N
%   Example:
%     ?- primes_range(60, 80, Primes).
%     Primes = [61, 67, 71, 73, 79] .

% Q2b. Replace the word "change_this" in the rule below.
%      HINT: Use upto and findPrimes.
primes_range(M, N, Primes) :-
   change_this.
```

3 Q3: spiral

Translate the function `spiral` from Assignment 1. The file `a4.pl` includes sample Haskell solutions. However, if you prefer, you may translate your own (correct) solution to `spiral` from a1.

4 Q4: Trees

Follow the instructions in `a4.pl`.