

CSE 341 — Additional CLP(\mathcal{R}) Discussion Questions

These questions use the following CLP(\mathcal{R}) rules:

```
member(X, [X|Xs]) .
member(X, [_|Ys]) :- member(X, Ys) .

member_cut(X, [X|Xs]) :- !.
member_cut(X, [_|Ys]) :- member(X, Ys) .

length([], 0) .
length(_|Xs, N) :- N>0, length(Xs, N-1) .
```

1. What are all the answers that CLP(\mathcal{R}) returns for the following goals? (If there are an infinite number give the first several; if there are none, say so.)

```
member(squid, [clam, squid, tuna])

member(squid, X)

member_cut(squid, [clam, squid, tuna])

member_cut(squid, X)

member(X, [a, b, c, d]), member(X, [c, d, e, f])

length([A, B, C], N)

length([A, B, C|Cs], N)
```

2. Write a CLP(\mathcal{R}) rule to find the average of a list of numbers. Fail if the list is empty.
3. Write a CLP(\mathcal{R}) rule `range(Lo, Hi, List)` that succeeds if `List` consists of all the numbers between `Lo` and `Hi` inclusive. (Assume that `Lo` and `Hi` are integers.) If `Lo` is greater than `Hi`, `List` should be empty.
4. Given your `range` rule from Question 3, what are all the results for the following goals?

```
range(2, 5, A)
range(5, 2, A)
range(A, B, [2, 3, 4, 5, 6])
range(A, B, [2, 4, 6])
```

5. Write another version of `range` that also takes a `Step` parameter: `range(Lo, Hi, Step, List)`. `Lo`, `Hi`, and `Step` don't have to be integers for this version. For example, `range(2.2, 3.0, 0.2, L)` should succeed with `L = [2.2, 2.4, 2.8, 3.0]`.
6. Is CLP(\mathcal{R}) statically typed? Is it type safe?
7. Compare the way parameters are passed in Haskell and CLP(\mathcal{R}).