

Solutions to Extra Credit Assignment: Unification & Prolog

Reminder: Lowercase letters and words beginning with a lowercase letter are values/constants. Uppercase letters and words beginning with an uppercase letter are variable names.

Problem 1 - Unification

- (a) `pointer(a)` and `pointer(pointer(b))` can be unified. The unifier is: $\{a \rightarrow \text{pointer}(b)\}$. (For this problem, `a` could've been pre-defined to be `pointer(b)`. If you said no to this question, I gave you full points.)
- (b) `boo` and `Baz` can be unified. The unifier is: $\{\text{Baz} \rightarrow \text{boo}\}$.
- (c) `[H, T]` and `[a, b, c]` can **NOT** be unified.
- (d) `foo(a)` and `foo(X, Y)` can **NOT** be unified.

Problem 2 - Facts & Queries in Prolog

Translate the following into a set of Prolog facts and rules. From (a) through (d), it should be possible for Prolog to infer (e). Define what your function means. For instance, if you use `Department(X, Y)` to be mean "X is in department Y", please say so.

- (a) `j` is in the `cs` department
- (b) If someone is in a department, they report to the head of that department.
- (c) `h` is the head of the `cs` department.
- (d) Everyone's salary is less than the salary of the person they report to.
- (e) `j`'s salary is less than `h`'s salary.

Here is one possible solution:

```
department(X, Z) means "X is in department Z"
report(X, Y) means "X reports to Y"
head(X, Z) means "X is the head of department Z"
lsalary(X, Y) means "X has less salary than Y"
```

```
j is in the cs department:
    department(j, cs).
```

```
If someone is in a department they report to the head of that department.
    report(X, Y) :- department(X, V) , head(Y, V).
```

```
h is the head of the cs department.
    head(h, cs).
```

```
Everyones salary is less than the salary of the person they report to.
    lsalary(X, Y) :- department(X, _) , report(X, Y).
```

```
j's salary is less than h's salary.
Query j's salary is less than h's salary by querying lsalary(j, h).
    lsalary(joe, sam).
```

Problem 3 - Cut

Let's suppose we have the following facts:

```
likes(a, dogs).
likes(a, cats).
likes(b, cats).
dislikes(c, dogs).
dislikes(d, dogs).
dislikes(d, cats).
```

What would the following queries print? (Suppose we keep pressing ";" until false is the output.)

(a) ?- likes(a, Animaltype), dislikes(Person, Animaltype).

```
Person = c
Animaltype = dogs ;

Person = d
Animaltype = dogs ;

Person = d
Animaltype = cats ;

false.
```

(b) ?- likes(a, Animaltype), !, dislikes(Person, Animaltype).

```
Person = c
Animaltype = dogs ;

Person = d
Animaltype = dogs ;

false.
```

(c) ?- likes(a, Animaltype), dislikes(Person, Animaltype), !.

```
Person = c
Animaltype = dogs ;

false.
```