CPSC 312 **Functional and Logic Programming** **January-April 2024**

Assignment Four: Logic Programming Introduction

Solution

Question One

(a) The query is `?- assignment(A, march, D).`
Here are the results from the query:

```
% swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- [cs312_2024].
true.

?- assignment(As,march,Day).
As = as4,
Day = 7 ;
As = as5,
Day = 14.
```

(b) Query is `?- ta(cs312,2024,P), email(P,E).`
There are no answers becuase the file doesn't specify the emails of the TAs.

(c) We could write a rule such as the following where `dif(X,Y)` is true if X and Y refer to different people.

```
?-   office_hour(P1,Day,_,_), office_hour(P2,Day,_,_),dif(P1,P2).
P1 = ilias,
Day = wednesday,
P2 = lily ;
P1 = lily,
Day = wednesday,
P2 = ilias ;
P1 = lily,
Day = friday,
P2 = ilias ;
P1 = ilias,
Day = friday,
P2 = lily ;
false.
```

(Without the dif, it could return the same person who has office hours on one day. I used `dif` which is a built-in predicate that true when the symbols are different – which, in general, does not imply the individuals are different). You could write your own facts about `dif` for the given constants.

(d) Here is the trace:

```
?- office_hour(P1,Day,_,_), office_hour(P1,Day2,_,_),nextday(Day,Day2), pname(P1,N1).
P1 = ilias,
Day = wednesday,
Day2 = thursday,
N1 = "Ilias" ;
P1 = ilias,
Day = thursday,
Day2 = friday,
N1 = "Ilias" ;
false.
```

(e) You can't define this given the language defined so far. There may be a TA called *sam* who is not listed and doesn't have any office hours. All of the given facts would be true, and yet the answer would be different. You could define a predicate such as $ta\_with\_no\_office\_hours(sam)$ to indicate that *sam* is a TA with no office hours.

(f) You need to know:

- When classes start
- What days of the week classes are on
- Which dates are holidays
- When classes end

Question Two

See `http://www.cs.ubc.ca/~poole/cs312/2024/as4/plumbing_ws.pl` for a solution.

Figure 1 shows the interpretation for the symbols in (d).

Question Three

(a) The following table gives the atoms added and the clause responsible for one possible sequence of clauses selected.

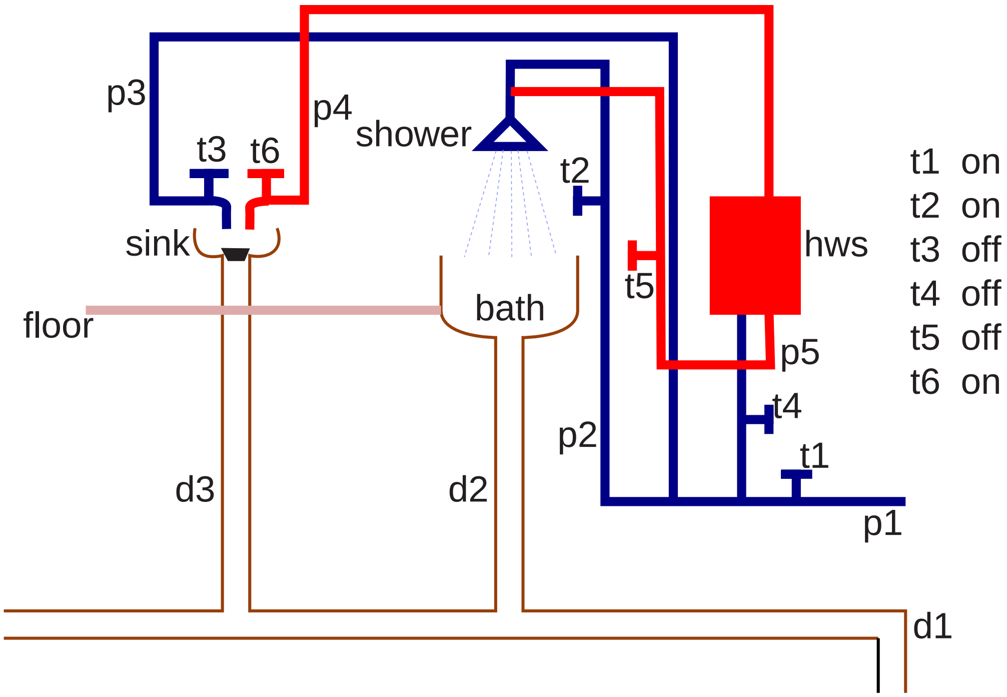


Figure 1: The Plumbing Domain

Atom Added	Clause
$\mathit{outgrabe}$	$\mathit{outgrabe}.$
$\mathit{manxome}$	$\mathit{manxome}.$
toves	$\mathit{toves} \leftarrow \mathit{manxome}.$
vorpal	$\mathit{vorpal} \leftarrow \mathit{manxome}.$
gyre	$\mathit{gyre} \leftarrow \mathit{manxome}.$
gimble	$\mathit{gimble} \leftarrow \mathit{outgrabe}.$
wabe	$\mathit{wabe}.$
slithy	$\mathit{slithy} \leftarrow \mathit{gyre}, \mathit{gimble}, \mathit{wabe}.$

(b) For the query $? - \mathit{slithy}$. here is one failing derivation:

Answer clause	Clause to resolve
$\mathit{yes} \leftarrow \mathit{slithy}$	$\mathit{slithy} \leftarrow \mathit{toves}, \mathit{brillig}.$
$\mathit{yes} \leftarrow \mathit{toves}, \mathit{brillig}.$	$\mathit{toves} \leftarrow \mathit{outgrabe}, \mathit{vorpal}.$
$\mathit{yes} \leftarrow \mathit{outgrabe}, \mathit{vorpal}, \mathit{brillig}.$	$\mathit{outgrabe}.$
$\mathit{yes} \leftarrow \mathit{vorpal}, \mathit{brillig}.$	$\mathit{vorpal} \leftarrow \mathit{manxome}.$
$\mathit{yes} \leftarrow \mathit{manxome}, \mathit{brillig}.$	$\mathit{manxome}.$
$\mathit{yes} \leftarrow \mathit{brillig}.$	$\mathit{brillig} \leftarrow \mathit{jubjub}.$
$\mathit{yes} \leftarrow \mathit{jubjub}.$	FAIL

(c) For the query $? - \mathit{slithy}$. Here is one successful derivation:

Answer clause	Clause to resolve
$yes \leftarrow slithy$	$slithy : -gyre, gimble, wabe.$
$yes \leftarrow gyre, gimble, wabe.$	$gyre : -manxome.$
$yes \leftarrow manxome, gimble, wabe.$	$manxome.$
$yes \leftarrow gimble, wabe.$	$gimble : -outgrabe.$
$yes \leftarrow outgrabe, wabe.$	$outgrabe.$
$yes \leftarrow wabe.$	$wabe.$
$yes \leftarrow .$	SUCCESS!

here is part of the trace

```
Call: (8) brillig ? creep
Call: (9) jubjub ? creep  (*)
Fail: (9) jubjub ? creep  (**)
Fail: (8) brillig ? creep
Redo: (8) toves ? creep   (***)
Call: (9) manxome ? creep
Exit: (9) manxome ? creep  (****)
```

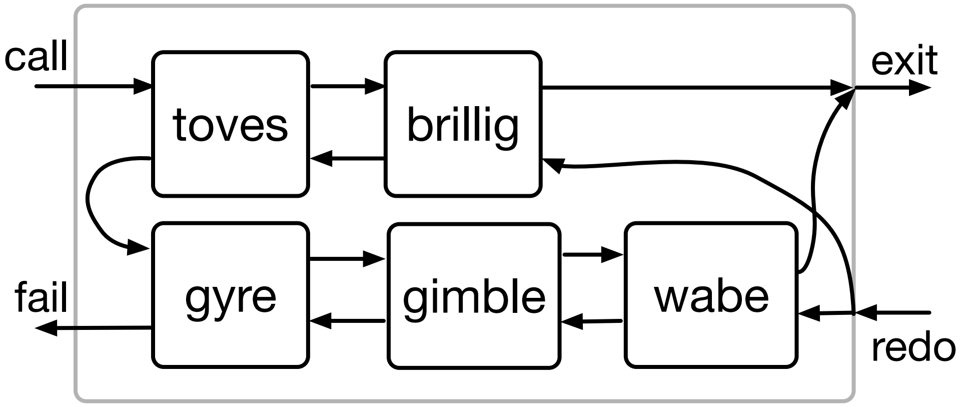
(*) it is calling jujube after brillig because of the clause $brillig : -jubjub.$

(**) jujube has no clauses so it fails. And then brillig fails as all of its clauses fail.

(***) toves needs to find another proof, so it calls the second one. SWI Prolog is reporting redo when it executes the down arrow (from fail to call).

(****) maxome succeeds as there is a fact for it.

(d) Figure



(e) There are 4 answers. There are 2 proofs for gyre, and 2 proofs for gimble. There is a proof for slithy for each combination of these.

Question Four

It should not have taken more than a few hours. Most of this should have been in understanding the material, not in doing busy work. I hope it was reasonable, and you learned something.

This question took about 3 minutes.