

CPSC 522 — Spring 2012

Assignment 1

Due: 12:30 p.m., Wednesday 18 January 2012.

Question 1

Consider the language with constant symbols a , b and c , predicate symbols p , q and r , and the knowledge base KB that consists of the clauses:

$$p(X) \leftarrow q(X).$$

$$p(Y) \leftarrow r(Y).$$

$$q(a).$$

$$r(b).$$

Suppose there are three individuals α , β and γ .

- How many interpretations are there where $D = \{\alpha, \beta, \gamma\}$?
- Give a model of KB . You must specify ϕ and π .
- Give an interpretation with the same domain that isn't a model of KB . You must specify ϕ and π .
- Give three atoms that are logical consequences of the knowledge base.
- Give three atoms that are not logical consequences of the knowledge base.
- How many models of KB are there where $D = \{\alpha, \beta, \gamma\}$?

Question 2

For this question you should use AILog (see http://artint.info/code/ailog/ailog_man.html).

Consider the domain of house plumbing represented in the diagram of Figure 1.

In this example, constants $p1$, $p2$ and $p3$ denote cold water pipes. $p1$ is the pipe coming in from the main water supply. Constants $p4$ and $p5$ represent hot water pipes (coming out from the hot water system). Constants $t1$, $t2$, $t3$, $t4$ and

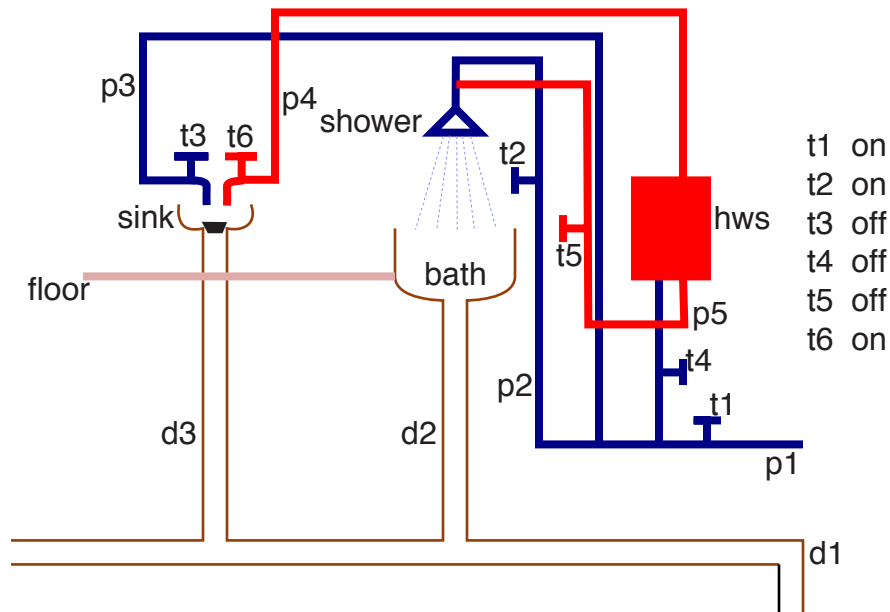


Figure 1: The Plumbing Domain

$t5$ denote taps and $d1$, $d2$ and $d3$ denote drainage pipes. The constants *shower* denotes a shower, *bath* denotes a bath, *sink* denotes a sink, *hws* denotes a hot water system, and *floor* denotes the floor. Figure 1 is intended to give the denotation for the constant symbols.

Suppose we have as predicate symbols:

- *pressurised*, where *pressurised*(P) is true if pipe P has mains pressure in it. $p1$ is always pressurised. Other pipes are pressurised if they are connected to a pressurised pipe through an open tap.
- *on*, where *on*(T) is true if tap T is on.
- *off*, where *off*(T) is true if tap T is off.
- *wet*, where *wet*(B) is true if B is wet.
- *flow*, where *flow*(P) is true if water is flowing through P .
- *plugged*, where *plugged*(S) is true if S is either a sink or a bath and has the plug in.
- *unplugged*, where *unplugged*(S) is true if S is either a sink or a bath and doesn't have the plug in.

Assume that the taps and plugs have been in the same positions for one hour; you don't need to consider the dynamics of turning on taps and inserting and removing plugs.

The file `plumbingbuggy.ailog` contains an AILog axiomatization for the case where tap $t1$ is on, $t2$ is on, $t3$ is off, $t4$ is off, $t5$ is off, and $t6$ is on. There is a plug in the sink, but not in the bath. (This has been formatted to be unreadable, but you don't need to be able to read the program, or know how it is implemented, to be able to debug it.)

Unfortunately there is a bug in the program. We know this because we can prove that the floor is wet, even though it isn't wet in the intended interpretation. Using AILog, your goal is to find a clause that is false in the intended interpretation, using just the intended interpretation of the symbols.

You need to hand in a trace of the AILog session, and show clearly which is the buggy rule.

Question 3

In this question, you are to write a definite clause knowledge base for the design of custom video presentations.

Assume that the video is annotated using the relation

$$\text{segment}(\text{SegId}, \text{Duration}, \text{Covers}),$$

where *SegId* is an identifier for the segment. (In a real application this will be enough information to extract the video segment). *Duration* is the time of the segment (in seconds). *Covers* is a list of topics covered by the video segment. An example of a video annotation is the database

$$\text{segment}(\text{seg0}, 10, [\text{welcome}]).$$

$$\text{segment}(\text{seg1}, 30, [\text{skiing}, \text{views}]).$$

$$\text{segment}(\text{seg2}, 50, [\text{welcome}, \text{computational_intelligence}, \text{robots}]).$$

$$\text{segment}(\text{seg3}, 40, [\text{graphics}, \text{dragons}]).$$

$$\text{segment}(\text{seg4}, 50, [\text{skiing}, \text{robots}]).$$

A presentation is a sequence of segments. You will represent a presentation by a list of segment identifiers.

- (a) Axiomatize a predicate

$$\text{presentation}(\text{MustCover}, \text{Maxtime}, \text{Segments})$$

that is true if *Segments* is a presentation whose total running time is less than or equal to *Maxtime* seconds, such that all of the topics in the list *MustCover* are covered by a segment in the presentation. The aim of this predicate is to design presentations that cover a certain number of topics within a time limit.

For example, the query

```
ask presentation([welcome, skiing, robots], 90, Segs).
```

should return at least the following two answers (perhaps with the segments in some other order):

```
presentation([welcome, skiing, robots], 90, [seg0, seg4]).
```

```
presentation([welcome, skiing, robots], 90, [seg2, seg1]).
```

Give the intended interpretation of all symbols used and demonstrate that you have tested your axiomatization (including finding all answers to your query) in ailog or Prolog. Explain briefly why each answer is correct.

Optional: (for those who have done logic programming before). Write a program that gives the best presentation. Note that this first requires a specification of when one presentation is better than another.

- (b) Assuming you have a good user interface and a way to actually view the presentations, list *three* things that the preceding program does not do that you may want in such a presentation system. (There is no correct answer for this part. You must be creative to get full marks).

Question 4

How long did the assignment take? What did you learn? Was it reasonable? What suggestions do you have to improve the assignment?