Your NAME: Solutions

DCS 440, Artificial Intelligence, Midterm 2, 1999
80 minutes, open book, open notes.
This exam has 4 problems on 7 single-sided pages.
Answer each question in the space provided.
Use backs for further work but indicate when you do.

1. Recall that the *situation calculus* provides a way of describing the effects of actions in logic. Predicates that change over time get an additional, final argument for a *situation*; a function $do(A,S)$ gives the situation resulting from doing action A in situation S.

Now, consider the action of a simplified CD player *p*—the CD player has two buttons, *a* and *b*; *b* turns the CD player on; *a* is the button that starts the CD playing when the player is on. The following situation calculus axioms are intended to describe the action of *p*.

```
playing(p,C,do(press(a),S)) ←
   isInserted(C,p,S) ∧
   isOn(p,S).                                  (Axiom 1)
isOn(p,do(press(b),S)).                        (Axiom 2)
```

1a. Describe the content of Axiom 1 and the content of Axiom 2 in English.

Axiom 1: After *a* is pressed in a situation where a CD is inserted into the player and the player is on, the player is playing that CD.
Axiom 2: After *b* is pressed in any situation, the player is on.

1b. Assuming that this CD player operates like others — suppose that initially *Pokémon: the first CD* is inserted in the player and the player is off. What sequence of actions will cause the Pokémon CD to be playing?

First press *b*, then press *a*.

1c. Based on your answer to 1b, you can derive a *query* (in the notation of the situation calculus) which you might then try to prove, to check that this sequence of actions has this effect. What is this query?

```
?playing(p,pokemon,do(press(a),do(press(b),init)))
```

1d. Formalize the initial conditions described in 1b that go into this planning problem—i.e., use the notation of situation calculus to write these conditions as atomic facts about some initial situation *init*.

`isInserted(p,pokemon,init)`
`isOff(p,init)` (optional)

1e. In addition to the representation of initial conditions and axioms 1 and 2, a "frame axiom", indicating what in the world is not changed by an action, is needed to prove the query in 1c. Write this frame axiom here.

`isInserted(p,C,do(press(b),S)))` ← `isInserted(p,C,S).` (and variants)

1f. Sketch the proof of the query in 1c from axioms 1 and 2 and your answers to 1d and 1e.

`?playing(p,pokemon,do(press(a),do(press(b),init))`
`?isOn(p,do(press(b),init), isInserted(p,pokemon,do(press(b),init)` (Axiom 1)
`?isInserted(p,pokemon,do(press(b),init)` (Axiom 2)
`?isInserted(p,pokemon,init)` (Frame axiom)
done. (Initial conditions)

2 This question is about the following scenario, which we can designate Scenario *W*:

> Suddenly in December, it's a warm sunny day—perfect for skipping school and strolling around New York City. You hit the web from `remus.rutgers.edu` to plan your trip, but you fail to access your trusty web sites: `www.weather.com` and `www.sidewalk.com`. Oddly, however, you can access `www.libraries.rutgers.edu` fine. What's going on? Is the web trying to send you a message?

Scenario *W* poses an explanation problem. Let's suppose we know a number of rules that predict failure and success of web access under a range of possible circumstances. These rules are written in terms of observations: `y(Addr)` to indicate that accessing `Addr` succeeds, and `n(Addr)` to indicate that accessing `Addr` fails. The circumstances that you can hypothesize are: `ok(Addr)` meaning everything is working to allow acess to a site; `crash`, indicating that netscape has crashed; `net`, indicating that the network connection outside Rutgers has gone off line; and `down(Addr)`, indicating that the web server at site `Addr` is not responding. Here are the rules:

```
n(Addr) ← outside(Addr) ∧ net.
n(Addr) ← down(Addr).
n(Addr) ← crash.
y(Addr) ← ok(Addr).
false ← outside(Addr) ∧ ok(Addr) ∧ net.
false ← down(Addr) ∧ ok(Addr).
false ← ok(Addr) ∧ crash.
```

Naturally, we also assume that we have a rule that allows us to correctly infer if an internet address `A` is outside Rutgers, so that `outside(A)` will be provable at the right times.

2a. In class we defined a relation `explain(F,H)` to find an explanation `H` for observations `F`. That is, if `explain(F,H)` is true, then you can assume the atomic facts in list `H` simultaneously and consistently (without proving `false`) and when you make these assumptions you then obtain proofs of each of the atomic facts in list `F`. Formalize a *query* that uses `explain` to pose the problem described in scenario *W*.

```
?explain([y(www.libraries.rutgers.edu),
n(www.weather.com), n(www.sidewalk.com)], H)
```

2b. Which of the following are consistent explanations for scenario *W*, given our knowledge base about web access? In other words, which of the following answer the query you identified in 2a given that knowledge base. (BRIEFLY explain your answers.)

Suppose (perhaps unrealistically) that any instance of failure is equally unlikely. Which explanation seems best, and why?

- `[ok(www.libraries.rutgers.edu), down(www.weather.com), down(www.sidewalk.com)]`

  Good explanation (consistent, allows proofs of all three observations). Two failures make it less likely.

- `[crash]`

  Not an explanation, because it gives no way to prove observation `y(www.libraries.rutgers.edu)`.

- `[ok(www.libraries.rutgers.edu), crash]`

  Inconsistent, because you can prove `false` from the two assumptions.

- `[ok(www.libraries.rutgers.edu), net]`

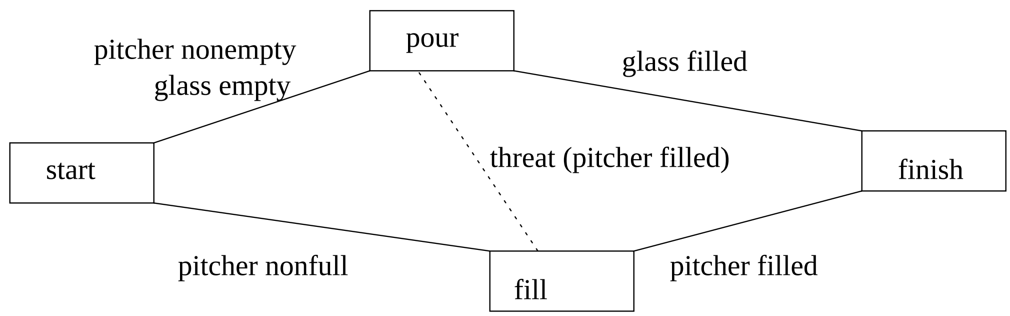
  Good explanation (consistent, allows proofs of all three observations). One failure makes it the most likely and hence the best explanation.

3 You have a table with a 4-glass pitcher 3/4 full of water and an empty glass of water. You can fill the pitcher at the sink and you can pour water from the pitcher into the glass. You want a full pitcher and a full glass of water.

3a What do you do?

First pour the water from the pitcher into the glass (pour) then fill the pitcher at the sink (fill).

3b Draw the plan you would need for this situation. Use nodes for the start and finish virtual actions and any actions in the plan; you can label the actions informally, in English. Use solid lines to show causal dependence among actions (again, label them, in informal English, with the condition that the lines represent). Use dotted lines to show other constraints in the plan.



3c Explain the ordering among the actions in your plan using the terminology of planning. Briefly describe how that ordering would be arrived at automatically by a partial-order planning algorithm.

Pour occurs before fill so that it does not threaten the causal link that fill establishes.

The POP algorithm adds pour and fill to the plan between start and finish actions. Then pour threatens the causal link between fill and finish that the pitcher stays fill. The threat has to be resolved by moving pour earlier (since pour cannot follow finish).

4 You are awarding prizes to children at a party. To facilitate tranquility, you want to give each child a prize that they regard as better than the prizes given to the others. You have five kinds of prizes: books (b), pens (p), software cds (s), organizers (o), and calculators (c). There are three children: Sandy, Maxie and Robin; with these preferences:

- Sandy (the bookish one): b, p equal and better than s, o, c.
- Maxie (the fussy one): o, p, c equal and better than s, b.
- Robin (the geeky one): s, c equal and better than b, o, p.

You have a constraint satisfaction problem.

4a Indicate the variables and possible values of the constraint satisfaction problem.

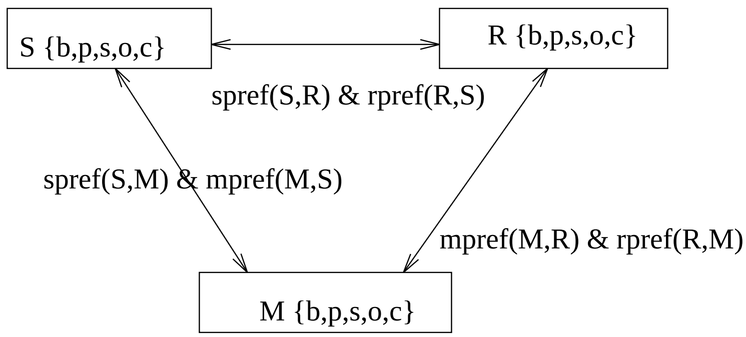
var	vals
S	b, p, s, o, c
M	b, p, s, o, c
R	p, p, s, o, c

4b Express the constraints involved among the values of variables.

Use `ipref(X,Y)` to indicate that individual i prefers item X to item Y.

```
spref(S,M)  spref(S,R)
mpref(M,S)  mpref(M,R)
rpref(R,S)  rpref(R,M)
```

4c Draw the constraint network for this problem.



4d Solve the network using arc consistency. Write out your answer as follows. Make a table saying in order: what constraint you treat next; and what values are eliminated by looking at that constraint. Then give the final value(s) for each variable.

A sample run:

constraint	eliminated
spref(S,M)	from S: $\{s,o,c\}$; from M: $\{b,p\}$
mpref(M,S)	from S: $\{p\}$; from M: $\{s\}$
spref(S,R)	from R: $\{b,p\}$
rpref(R,S)	from R: $\{o\}$
mpref(M,R)	from R: $\{c\}$
rpref(R,M)	from M: $\{c\}$

Final result: S is b, M is o, R is s

For reference: Sandy: b,p equal and better than s,o,c; Maxie: o,p,c equal and better than s,b; Robin: s,c equal and better than b,o,p.