6.825

Topics in Artificial Intelligence

2005/09/29

Dan Ports
drkp@mit.edu

Project 1a

Problem 1:

We begin by translating the original six axioms (1-6):

10

```
(Connects(W1, W2) ^
 Connects(W2, W3) ^
 Connects(W3, W4) ^
 Connects(W4, W1)) ^

(~Connects(W1, W3) ^ ~Connects(W1, W4) ^ ~Connects(W2, W4) ^
 ~Connects(W2, W1) ^ ~Connects(W3, W1) ^ ~Connects(W3, W2) ^
 ~Connects(W4, W2)  ^ ~Connects(W4, W3) ^ all w ~Connects(w, w)) ^

(all s
  (Unconflicted(s) <->
    (all w all a1 all a2 (On(a1,w,s) ^ On(a2,w,s)
                          -> Equals(a1,a2))))) ^

(Unconflicted(S1)) ^

(all a all w2
  (On(a,w2, S2) ->
    (exists w1
      (On(a,w1,S1) ^ Connects(w1,w2) ^ Allowed(a,w1,w2))))) ^

(all a1 all w1 all w2
    Allowed(a1,w1,w2) <->
      (Equals(w1, w2) v ~(exists a2 On(a2,w2,S1))))
```

10

Problem 2:

Next, we express the following:

- Every athlete is always on some workout station:

$$\forall a\, \forall s\, \exists w \;\; On(a, w, s) \tag{7}$$

```
(all a all s exists w
   On(a, w, s))
```

- No athlete is on two stations at the same time:

$$\forall s\, \forall a\, \forall w_1\, \forall w_2 \;\; \textsc{On}(a, w_1, s) \wedge \textsc{On}(a, w_2, s) \implies w_1 = w_2 \tag{8}$$

```
(all s all a all w1 all w2
  ((On(a, w1, s) ^ On(a, w2, s)) -> Equals(w1, w2)))
```

10

Problem 3:

The Java solver found the following satisfying assignment[1]:

```
Allowed_A1_W1_W1
Allowed_A1_W1_W3
Allowed_A1_W2_W1
Allowed_A1_W2_W3
Allowed_A1_W3_W1
Allowed_A1_W3_W3
Allowed_A1_W4_W1
Allowed_A1_W4_W3
Allowed_A2_W1_W1
Allowed_A2_W1_W3
Allowed_A2_W2_W1
Allowed_A2_W2_W3
Allowed_A2_W3_W1
Allowed_A2_W3_W3
Allowed_A2_W4_W1
Allowed_A2_W4_W3
Connects_W1_W2
Connects_W2_W3
Connects_W3_W4
Connects_W4_W1
On_A1_W1_S2
On_A1_W4_S1
On_A2_W2_S1
On_A2_W3_S2
Unconflicted_S1
Unconflicted_S2
```

In this satisfying assignment, A_1 is at station W_4 initially, then moves to W_1 in S_2, and A_2 moves from W_2 to W_3. Both S_1 and S_2 are unconflicted, and the values for Allowed are as we expect: both A_1 and A_2 are allowed to move anywhere except for W_2 and W_4, the original positions of the two athletes.

10

Problem 4:

To show that there are no possible conflicts, we add an axiom saying that S_2 is conflicted:

```
(~(Unconflicted(S2)))
```

Then, performing a search for a satisfying assignment will tell us if there is a possible assignment such that S_1 is unconflicted and S_2 is conflicted: i.e. if it is possible to create a conflict when none existed before. The SAT solver failed to find a satisfying assignment, and it is complete, so we know that none existed. Therefore, there are no possible conflicts in this domain.

It would not have been correct to use WalkSAT for this test, since WalkSAT is not a complete solver; it may fail to find a satisfying assignment even if one exists. So it cannot be used to prove that a set of axioms is never satisfiable.

10

Problem 5:

We now allow athletes to stay on their current station. We begin by adding connections from each station to itself; this involves a $\forall w$ Connects(w, w) clause to (1) and removing the opposite from (2):

---

[1] To make satisfying assignments readable, I created a postprocessor that parses the assignment and displays only the instantiations that are true, in alphabetical order. False instantiations are not listed. All satisfying assignments shown in this document are in this format.

```
(Connects(W1, W2) ^
 Connects(W2, W3) ^
 Connects(W3, W4) ^
 Connects(W4, W1) ^
 (all w Connects(w,w))) ^

(~Connects(W1, W3) ^ ~Connects(W1, W4) ^ ~Connects(W2, W4) ^
 ~Connects(W2, W1) ^ ~Connects(W3, W1) ^ ~Connects(W3, W2) ^
 ~Connects(W4, W2) ^ ~Connects(W4, W3)) ^
```

But this does not suffice: the station is already occupied (by the same athlete), so our current definition of ALLOWED will not allow the athlete to stay in place. We must modify it to allow a move if the station is currently occupied by the same athlete trying to move there, in addition to if it is empty. This makes our axiom:

$$\forall a_1\, \forall w_1\, \forall w_2\, \text{ALLOWED}(a_1, w_1, w_2) \iff (w_1 = w_2) \lor (\exists a_2\, \text{ON}(a_2, w_2, S_1)) \tag{9}$$

```
(all a1 all w1 all w2
   Allowed(a1,w1,w2) <->
      (Equals(w1, w2) v ~(exists a2 On(a2,w2,S1)))) ^
```

To verify that this is correct, we add the following test axiom, which requires that some athlete stays in place:

```
      (exists a exists w (On(a, w, S1) ^ (On(a, w, S2))))
```

Running the solver gives a satisfying assignment, in which A_1 stays in place in W_2, and A_2 moves from W_3 to W_4.

The domain is still unconflicted. To verify this, we add the $\neg$UNCONFLICTED(S_2) axiom again, as before, and attempt to find a satisfying assignment. No satisfying assignment exists, so the domain is unconflicted.

10

Problem 6:

Now we consider Layout 2. We replace (1) and (2) with the following:

```
(Connects(W1, W2) ^
 Connects(W1, W3) ^
 Connects(W2, W3) ^
 Connects(W3, W2) ^
 Connects(W2, W4) ^
 Connects(W3, W4) ^
 (all w Connects(w,w))) ^

(~Connects(W1, W4) ^ ~Connects(W2, W1) ^ ~Connects(W3, W1) ^
 ~Connects(W4, W2)  ^ ~Connects(W4, W3) ^ ~Connects(W4, W1))
```

We test for conflictedness using the $\neg$UNCONFLICTED(S_2) axiom, which finds a conflicting assignment. The full set of axioms and the output of the solver follow:

```
(Connects(W1, W2) ^
 Connects(W1, W3) ^
 Connects(W2, W3) ^
 Connects(W3, W2) ^
 Connects(W2, W4) ^
 Connects(W3, W4) ^
 (all w Connects(w,w))) ^

(~Connects(W1, W4) ^ ~Connects(W2, W1) ^ ~Connects(W3, W1) ^
```

```
 ~Connects(W4, W2)  ^ ~Connects(W4, W3) ^ ~Connects(W4, W1)) ^

(all s
  (Unconflicted(s) <->
    (all w all a1 all a2 (On(a1,w,s) ^
                          On(a2,w,s) -> Equals(a1,a2))))) ^

(Unconflicted(S1)) ^

(all a all w2
  (On(a,w2, S2) ->
    (exists w1
      (On(a,w1,S1) ^ Connects(w1,w2) ^ Allowed(a,w1,w2))))) ^

(all a1 all w1 all w2
   Allowed(a1,w1,w2) <->
      (Equals(w1, w2) v ~(exists a2 On(a2,w2,S1)))) ^

(all a all s exists w
   On(a, w, s)) ^

(all s all a all w1 all w2
  ((On(a, w1, s) ^ On(a, w2, s)) -> Equals(w1, w2))) ^

(~(Unconflicted(S2)))
```

```
Allowed_A2_W2_W1
Allowed_A2_W2_W2
Allowed_A2_W2_W4
Allowed_A2_W3_W1
Allowed_A2_W3_W3
Allowed_A2_W3_W4
Allowed_A2_W4_W1
Allowed_A2_W4_W4
Connects_W1_W1
Connects_W1_W2
Connects_W1_W3
Connects_W2_W2
Connects_W2_W3
Connects_W2_W4
Connects_W3_W2
Connects_W3_W3
Connects_W3_W4
Connects_W4_W4
On_A1_W2_S1
On_A1_W4_S2
On_A2_W3_S1
On_A2_W4_S2
Unconflicted_S1
```

So we see that in S_1, A_1 and A_2 are on stations W_2 and W_3 respectively, but both try to move to W_4 in S_2, creating a conflict.

Problem 7:

10

We change the definition of ALLOWED to avoid this situation. Now, an athlete can either stay in the same place, or move to a station that no other athlete can move to. (Note that this also forbids moving to a station that another athlete is already on, since the other athlete could stay in place.)

$$\forall a_1\, \forall w_1\, \forall w_2\; \textsc{Allowed}(a_1, w_1, w_2) \iff \\ (w_1 = w_2) \lor (\exists a_2\, \exists w_3\; a_2 \neq a_1 \land \textsc{On}(a_2, w_3, S_1) \land \textsc{Connects}(w_3, w_2)) \quad (10)$$

```
(all a1 all w1 all w2
   Allowed(a1,w1,w2) <->
      (Equals(w1, w2) v
       ~(exists a2 exists w3
          (~Equals(a2, a1) ^ On(a2, w3, S1) ^ Connects(w3, w2)))))
```

To verify the sanity of this approach, we search for a satisfying assignment. We obtain the following:

Athlete	S_1	S_2
A_1	W_1	W_1
A_2	W_3	W_4

A_1 is unable to move from W_1 because it can only move to W_2 or W_3, and A_2 could either stay in W_3 or move to ~~W_2~~. W_4 (not W_2 because ~~A_2~~ ~~can~~ ~~no~~ A_1 can move there)

We can already see that this will cause deadlock. If we constrain A_1 to be on W_2 and A_2 to be on W_3 in S_1, neither can move:

Athlete	S_1	S_2
A_1	W_2	W_2
A_2	W_3	W_3

But the situation can never become conflicted. Adding the $\neg$UNCONFLICTED(S_2) axiom, the solver fails to find a satisfying assignment, meaning that the domain can never become conflicted.

Problem 8:

We formalize the notion of deadlock as follows: S_1 is deadlocked if every athlete cannot move (and can only stay at the same station). That is, for every athlete, the only station CONNECTed to its current situation that it is ALLOWED to move to can only be its current station.

In FOL:

this should be 'all' – you want to constrain all possible starting configurations of the athletes, not just one.

10/12

$$\text{DEADLOCKED}(S_1) \iff$$
$$\forall a\, \exists w_1\, \text{ON}(a, w_1, S_1) \wedge [\forall w_2\, (\text{CONNECTS}(w_1, w_2) \wedge \text{ALLOWED}(a, w_1, w_2)) \implies w_1 = w_2] \quad (11)$$

```
(Deadlocked(S1) <->
  (all a exists w1
    (On(a, w1, S1) ^
     (all w2 ((Connects(w1, w2) ^
              Allowed(a, w1, w2)) -> Equals(w1, w2)))))) ^
```

Using the CONNECTS definitions for Layout 1, and the definition of ALLOWED given in (10), there can be no deadlock. We test this by adding the axiom DEADLOCKED(S_1) and searching for a satisfying assignment. The solver fails to find such an assignment, so deadlock is not possible.

Problem 9:

Now we return to the CONNECTS definitions for Layout 2. We show that it is conflicted by requiring DEADLOCKED(S_1) and finding a satisfying assignment. The solver finds the following deadlocked assignment:

Athlete	S_1	S_2
A_1	W_2	W_2
A_2	W_3	W_3

A_1 cannot move from W_2 because the only adjacent stations are W_3, which is occupied by A_2, and W_4, which A_2 could also move to. A_2 cannot move from W_3 for an equivalent reason.

The assignment follows:

7/8

```
Allowed_A2_W2_W1
Allowed_A2_W2_W2
Allowed_A2_W3_W1
Allowed_A2_W3_W3
Allowed_A2_W4_W1
Allowed_A2_W4_W4
Connects_W1_W1
Connects_W1_W2
Connects_W1_W3
Connects_W2_W2
Connects_W2_W3
Connects_W2_W4
Connects_W3_W2
Connects_W3_W3
Connects_W3_W4
Connects_W4_W4
Deadlocked_S1
On_A1_W2_S1
On_A1_W2_S2
On_A2_W3_S1
On_A2_W3_S2
Unconflicted_S1
Unconflicted_S2
```

Where's the rest of the assignment?

In particular, why do we only see the Allowed predicates for A2?

Problem 10:

We break symmetry by introducing a notion of *priority* between athletes. An athlete is now allowed to move to a new station if there exists no other athlete *of lower priority* that could move there.

We arbitrarily assign priorities, with A_1 having higher priority than A_2. Note that this axiom defines a relation between individual athletes, but could be easily extended if there were more than two athletes.

```
      (Priority(A1, A2) ^ ~Priority(A2, A1) ^ (all a ~Priority(a, a))) ^
```

We define a utility predicate BLOCKED. An athlete a is blocked from moving to a station w if there exists some other athlete with higher priority on a station that connects to w.

10/10

```
(all a1 all w1
  Blocked(a1, w1) <->
   (exists a2 exists w2
          (~Equals(a2, a1) ^ Priority(a2, a1) ^
           On(a2, w2, S1) ^ Connects(w2, w1)))) ^
```

Now we redefine ALLOWED. An athlete a is now allowed to move from w_1 to w_2 if it is not blocked from moving to w_2. Moreover, it is not allowed to stay at the same station unless every adjacent station is blocked. This keeps the athletes moving if possible.

maybe problematic - what happens if an athlete is on a dead-end?

```
(all a1 all w1 all w2
   Allowed(a1,w1,w2) <->
      ((Equals(w1, w2)  ^ (all w3 (Connects(w1, w3) -> Blocked(a1, w3)))) v
       (~Equals(w1, w2) ^ ~(Blocked(a1, w2))))) ^
```

The result is that, if we constrain A_1 to begin on W_2 and A_2 to begin on W_3 in S_1, A_1 is able to move to W_4.

Athlete	S_1	S_2
A_1	W_2	W_4
A_2	W_3	W_3

Furthermore, by adding a DEADLOCKED(S_1) axiom, we can test whether deadlock is ever possible. There are no satisfying assignments, so deadlock is avoided.

Problem 11:

10/10

This implication should not be a biconditional. If it were, then for all w_2, if there exists a w_1 such that a is on w_1 in S_1, w_1 connects to w_2, and a is allowed to move from w_1 to w_2, then a would be on w_2 in S_1. This is certainly true for *some* w_2, but not for *all* w_2. It's possible to have multiple choices for where a would move, and the biconditional would mean that a would then make all possible moves, which wouldn't make sense (and would be prevented by the axiom that asserts that each athlete is always on one station.)

Problem 12:

A better definition of liveness that incorporates fairness might require that every athlete gets to move on *some* step, i.e.

10/10

$$\forall a\ \exists s\ \exists w_1\ \exists w_2\ \ \mathrm{ON}(a, w_1, s) \wedge \mathrm{CONNECTS}(w_1, w_2) \wedge \mathrm{ALLOWED}(a, w_1, w_2) \tag{12}$$

But we could not try to prove or disprove this condition using the finite-domain method. This would require us to consider many states, since we are only ensuring that every athlete gets to move on some step. Our finite universe with only two states would not suffice, since it only considers one starting position. In fact, any finite number n of states would not suffice, since there might be some athlete that does not get to move until state $n+1$. So we would need to use some other method for first-order logic theorem proving. good

117/110

Just for completeness, note that the full set of axioms used in this project are available from `http://drkp.net/svn/classes/6.825/proj1/`.