NOTE: equations look whack in the browser. To see them properly, go to Editing->Open in Desktop App

1

a

i)

[]

|

p(X) <-

/ \

p(X) <- q(X) p(X) <- s(X)

\ /

p(X) <- q(X), s(X)

Any h θ-subsumes any h’ that is reachable from h in the lattice.

ii)

We use a top-down approach starting from the top of the lattice. at the top is too general, since it entails the negative example. One layer lower, we have two possibilities: entails the positive example, since we ha

ve , but not the negative one, since we don’t have . The same argument goes for .

Thus, the two most general inductive solutions are and .

b

i)

We start with the seed and construct an abductive proof.

Let the abductive triple be

Where

|  |  |  |
| --- | --- | --- |
| |    |    |   |  | | --- | | | |   |    |   |  | | --- | | | |   |  □ |

The abductive answer is .

We can now find mode body predicates from using deduction. Since all mode body predicates are already included in B as atoms, no deduction is necessary. Here we choose the mode body predicates that conform with the head’s input variables (namely R is r1), and are needed for the kernel set K:

We can thus construct the ground kernel set.

By replacing the constants with variables, we get the wanted set:

ii)

Both clause heads are needed to cover the positive example, as requires both and .

In order to entail , we need to have and thus , as only r2 has coins. The kernel set requires to . Since we have , is actually entailed. The second condition requires . Since there is no such ground clause in B, it is that stops from being entailed. Thus, we only need the second clause for the first negative example.

For the example , we have . The first clause in K requires and , the latter of which does not exist. The second clause in K requires , which exists in B as . Thus, the first clause is needed to not entail the second negative example.

Thus, T does not accept an inductive solution more general than K.

iii)

No, since Progol5 can only generate a single clause per seed example, but we need two clauses for a solution.

We can add the contrapositives to B:

If we try to SLD derive the mode head openDoor using Progol5, we fail:

|

|

|

|

As you can see, we need a clause for , where Progol5 gets stuck.

2

a

i)

The kernel set is defined as follows:

Where the Δs are derived abductively. The ground kernel set are the ground clauses:

ii)

A ground kernel set is more general than a ground bottom set. One reason is that a bottom set consists of only one clause, while a kernel set can have multiple clauses. There are non-OPL tasks that may require multiple clauses to prove a single seed example. These tasks are in general non-derivable by Bottom Generalisation because in order to prove the conjunction the two clauses you would need to entail from Bot(B,e+ ) a disjunction of the negated heads. By definition of Bot(B, e+) each of these negated disjuncts would need to be proved (by resolution) from BU{~e+} and this is often not the case for solutions that require multiple clauses to prove a single seed example.

iii)

b

i)

Each mode declaration gains a unique identifier:

We include a rule for each head declaration:

We include a rule for each body declaration:

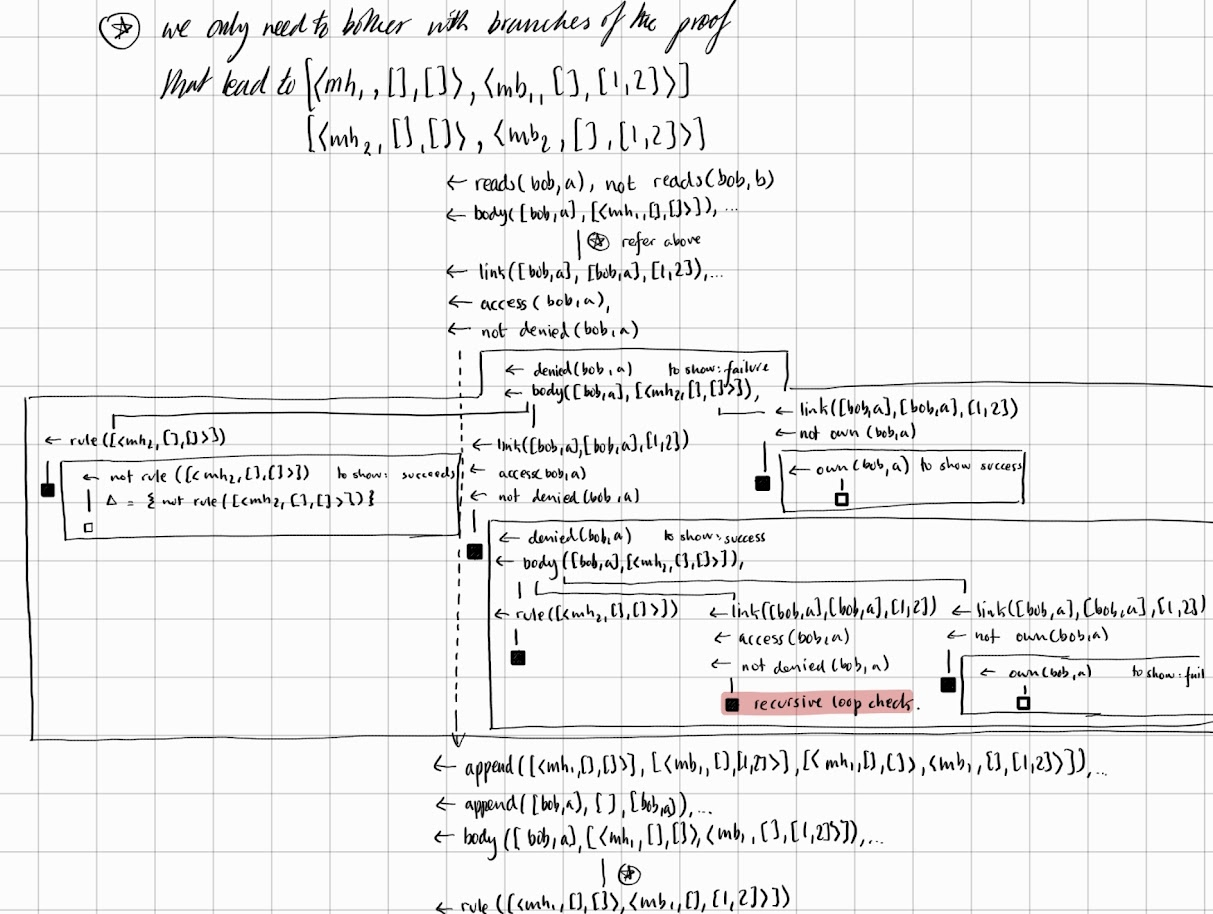
In addition, we include the base case rule:

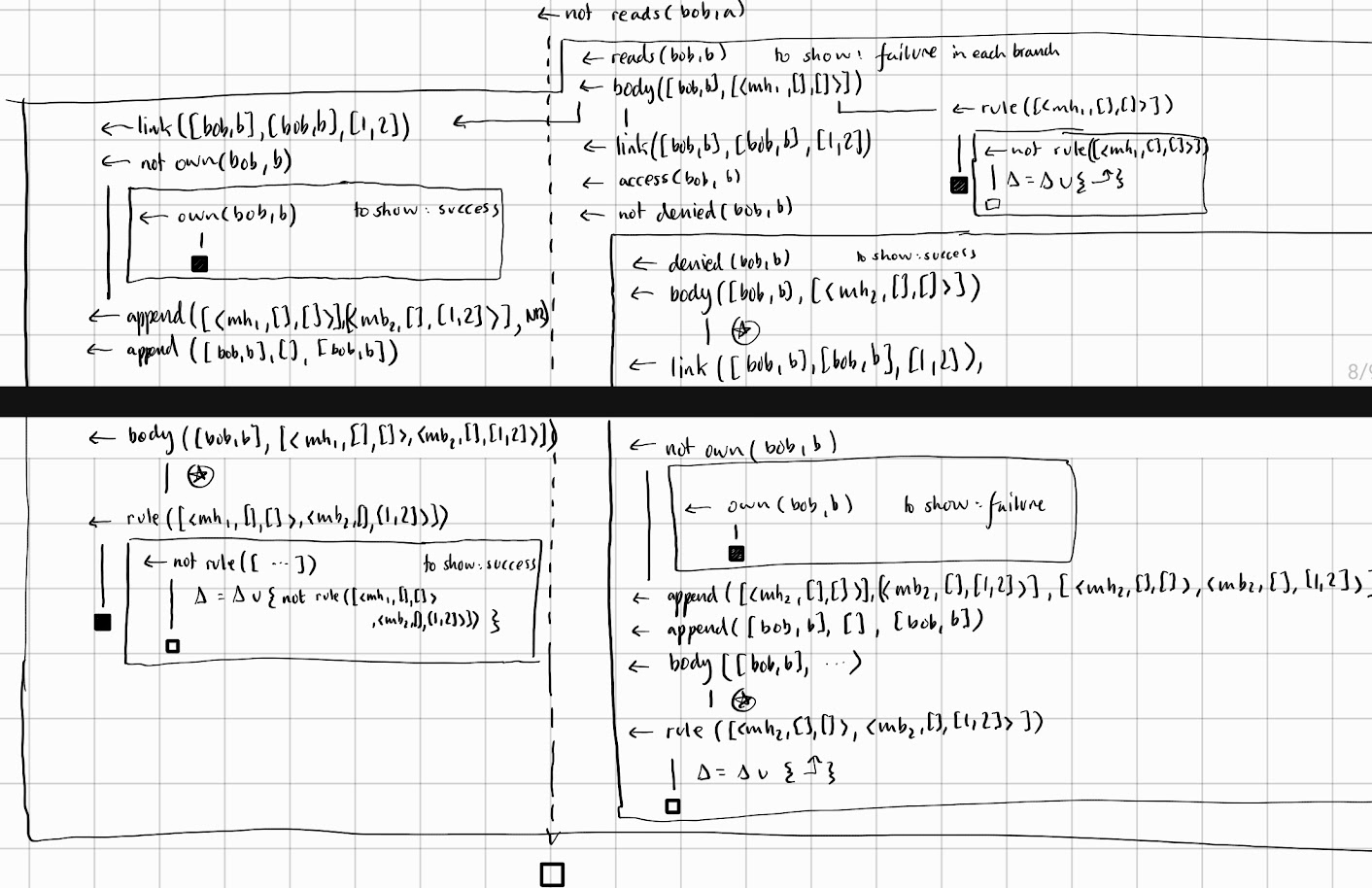
T is given by the set of the above rules.

The set of abducibles is given by the set of ground atoms:

ii)

~~If you can’t read it, comment and I’ll try to neaten it (or someone else can share theirs)~~

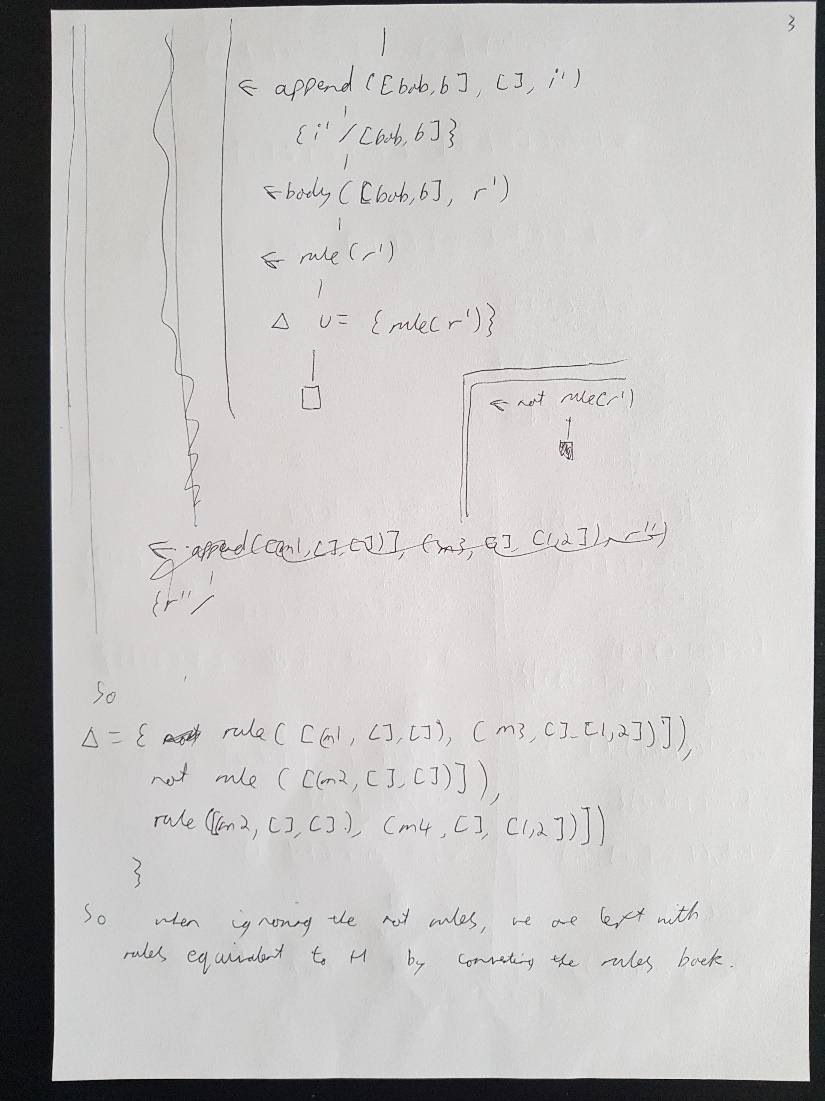
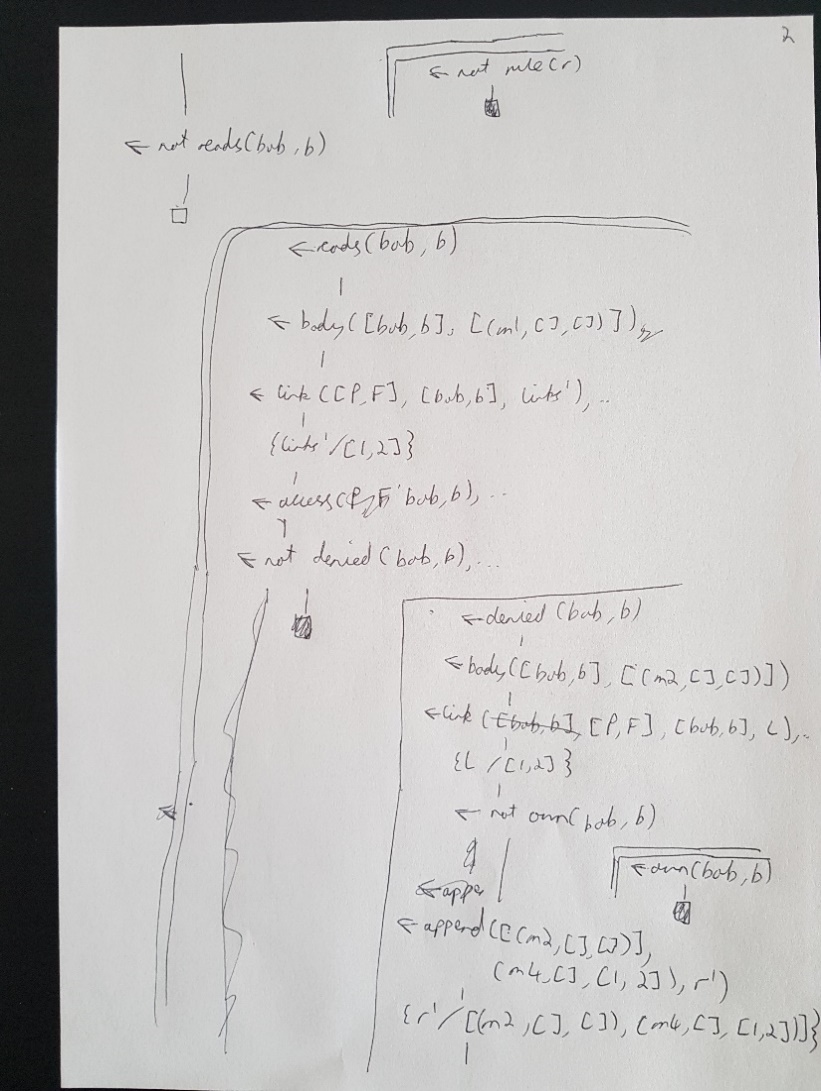
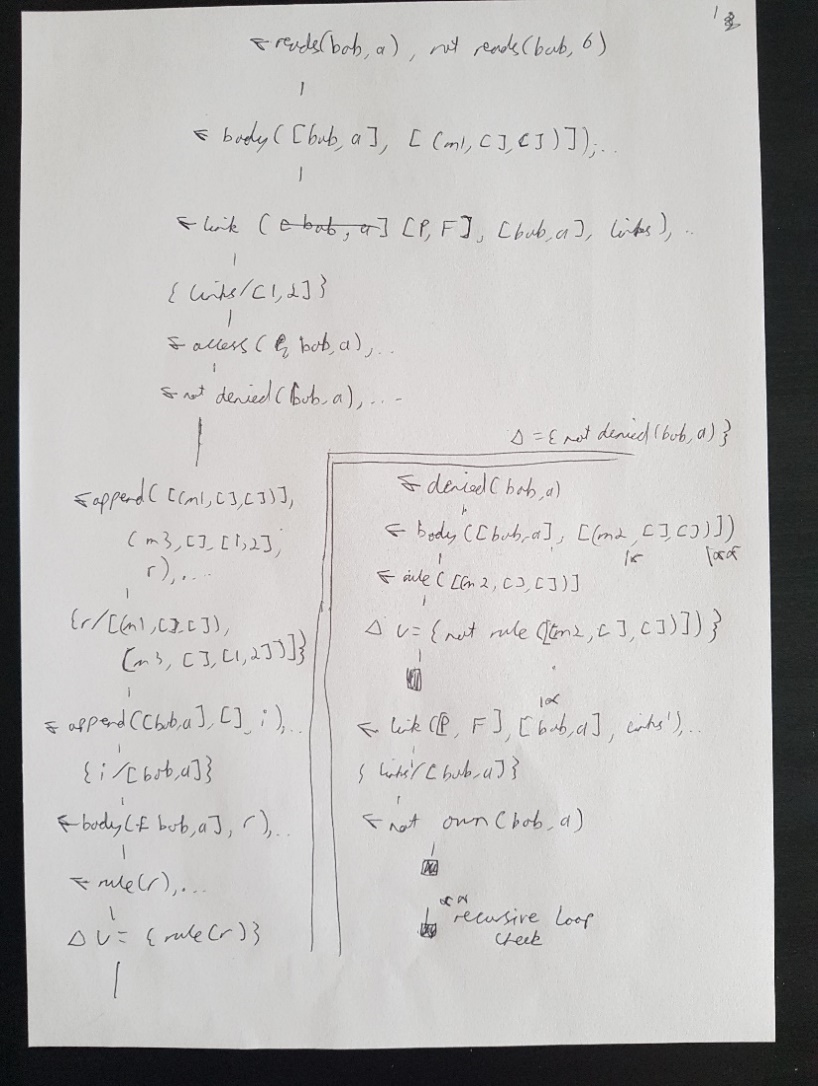
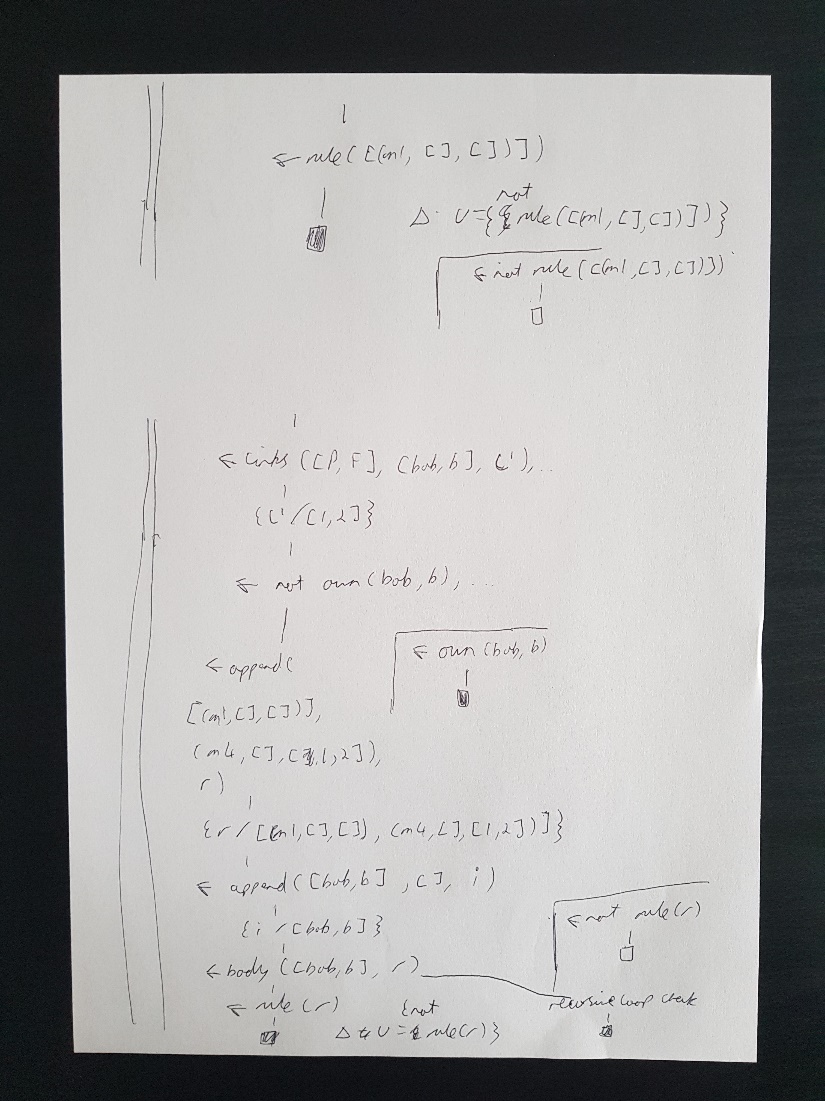
Screenshot cropped out



Therefore, we have learnt that

Therefore, translating back into the language of our program we get H.

(Old solution)

3

a

i)

p(X) <- t1(X)

p(X) <- t1(X), q(Y), t2(Y)

p(X) <- t1(X), q(Y), t2(Y), q(Z), t2(Z)

p(X) <- t1(X), q(Y), t2(Y), r(X, Y, Y)

p(X) <- t1(X), q(Y), t2(Y), q(Z), t2(Z), r(X, Y, Z)

Isn’t the above also missing: p(X)<- t1(X), t2(Y), t2(Z), r(X,Y,Z) ???

ii)

p(X) :- t1(X), rule(1)

p(X) :- t1(X), q(Y), t2(Y), rule(2)

p(X) :- t1(X), q(Y), t2(Y), q(Z), t2(Z), rule(3)

p(X) :- t1(X), q(Y), t2(Y), r(X, Y, Y), rule(4)

p(X) :- t1(X), q(Y), t2(Y), q(Z), t2(Z), r(X, Y, Z), rule(5)

:- not goal

goal :- p(a), p(b), not p(c)

% Background knowledge B needs writing out too (omitted)

{rule(1), rule(2), rule(3), rule(4), rule(5)}

minimise[rule(1)=1, rule(2)=2, rule(3)=3, rule(4)=3, rule(5)=4]

iii) 2021 – not assessed

b)i)

The answer sets of B are F = {q(1), q(2)}

A1 = F

A2 = F U {p(2)}

A3 = F U {p(1)}

A4 = F U {p(1), p(2)}

So the empty set is a brave inductive solution to any combination of p(1), p(2), not p(1), not p(2). And q(1) & q(2) have to be in E^+ else it isn’t satisfiable.

There are no such examples as the set of answer sets of B U <empty> is the superset of the set of answer sets of B U H\*.

ii)

E^- = {p(1), p(2)}

By part i, the empty set doesn’t cautiously entail it due to A2. Then the only answer set of B U H\* is {q(1), q(2)} so it cautiously entails it.

ii) E^+ = {/p(2)}, E^- = {p(1)}

By A1, the empty set isn’t a solution.

Then only answer set of B U H\* is {p(2), q(1), q(2)}

The answer sets of B2 U H\* are:

Answer: 1q(1) q(2)Answer: 2q(1) q(2) p(1)Answer: 3q(1) q(2) p(2)

No examples exist because the intersection of all answer sets of B2 U H\* is just the facts in B2. The empty hypothesis would hence also be a solution.

iv)v)vi) not assessed 2021

4a)i) F= {t(1), t(2)}

A1 = F U {p(1), q(2, 1), q(2,2)}

A2 = F U {p(1), p(2)}

A3 = F U {p(2), q(1, 2), q(1, 1)}

A4 = F U {q(1,1), q(2,1), q(1,2), q(2,2)}

ii) No as the answer sets that covers q(1,1) are A3 & A4 but q(1, 2) is also an element of A3 & A4

iii) No as q(1, 1) is an element of A3

b)i)

happy(X) <- passExam(X), r1.

happy(X)winLottery(X), r2.

passExam(mary).

0.2::winLottery(mary).

0.6::r1.

0.8::r2.

ii) There are 3 probabilistic facts and hence 2^3 = 8 possible worlds.

iii) On piazza, she said for any probability question construct the SLD derivation, then BDD, then compute the odds

