v1.

a)

i)

A hypothesis is complete if it covers all of the positive examples. It is consistent if it doesn’t cover any of the negative examples and doesn’t contradict itself or the background knowledge.

ii)

H1: q(X, X) <- s(X, X).

This is inconsistent because together with s(a, a) which is in the background knowledge it entails r(a, a) which is a negative example.

H2: …

Any H that subsumes Bot(B, e+) would work.

Bot(B, e+) is the set of ground literals whose negation can be entailed by B ⋀ ¬e+.

B ⋀ ¬e+ ⊨ s(a, b) ⋀ t(b) ⋀ s(a, a) ⋀ ¬r(a, b) ⋀ ¬q(a, b)

Bot(B, e+) = ¬(s(a, b) ⋀ t(b) ⋀ s(a, a) ⋀ ¬r(a, b) ⋀ ¬q(a, b))

= ¬s(a, b) ⋁ ¬t(b) ⋁ ¬s(a, a) ⋁ r(a, b) ⋁ q(a, b)

h┴ = q(X, Y) <- s(X, Y), t(Y), s(X, X), ¬r(X, Y)

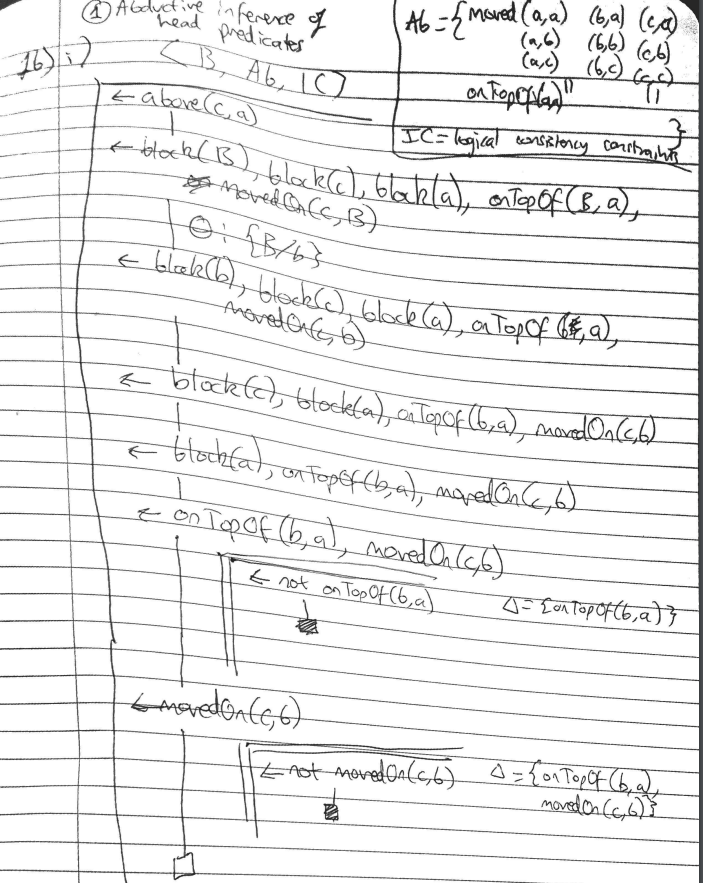
H2: q(X, Y) <- s(X, Y), t(Y), s(X, X)

b)

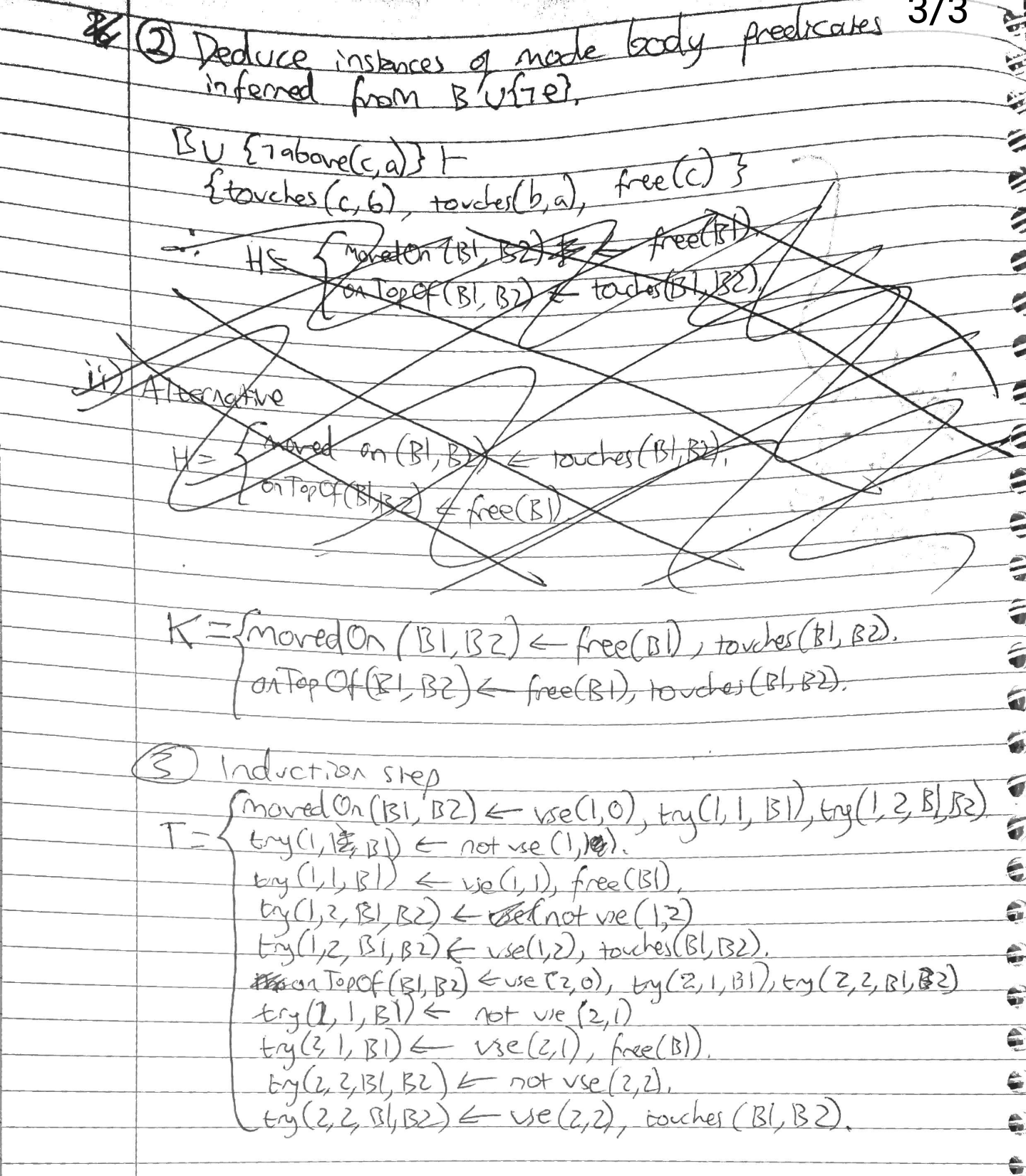
i)

Abductive task <B, Ab, IC> where B is given background knowledge, Ab = {movedOn(a,a), movedOn(a,b), movedOn(a,c), movedOn(b,a), movedOn(b,b), movedOn(b,c), movedOn(c,a), movedOn(c,b), movedOn(c,c), onTopOf(a,a), onTopOf(a,b), onTopOf(a,c), onTopOf(b,a), onTopOf(b,b), onTopOf(b,c), onTopOf(c,a), onTopOf(c,b), onTopOf(c,c)} and IC is logical consistency constraints.

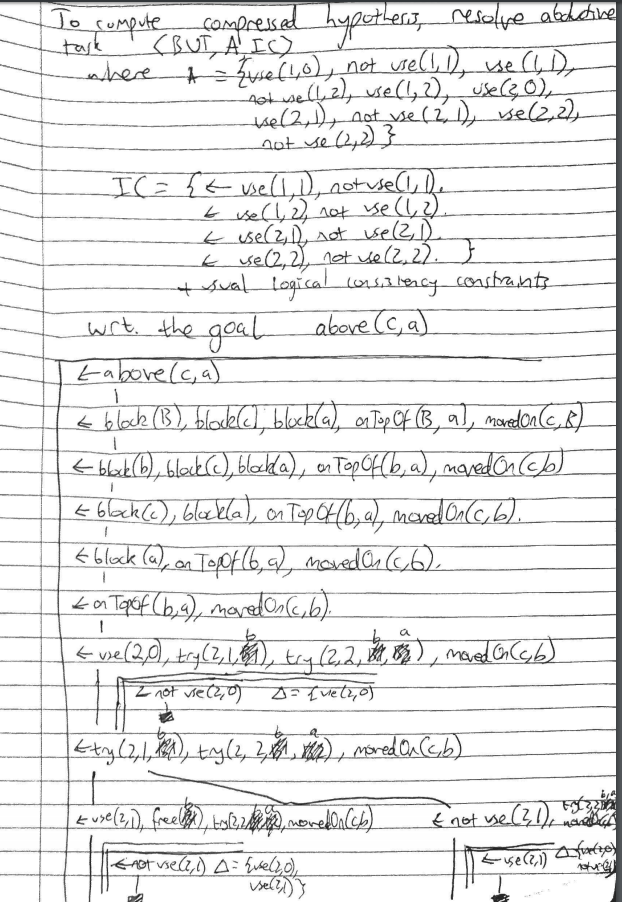
The goal of the abduction is the seed above(c,a).

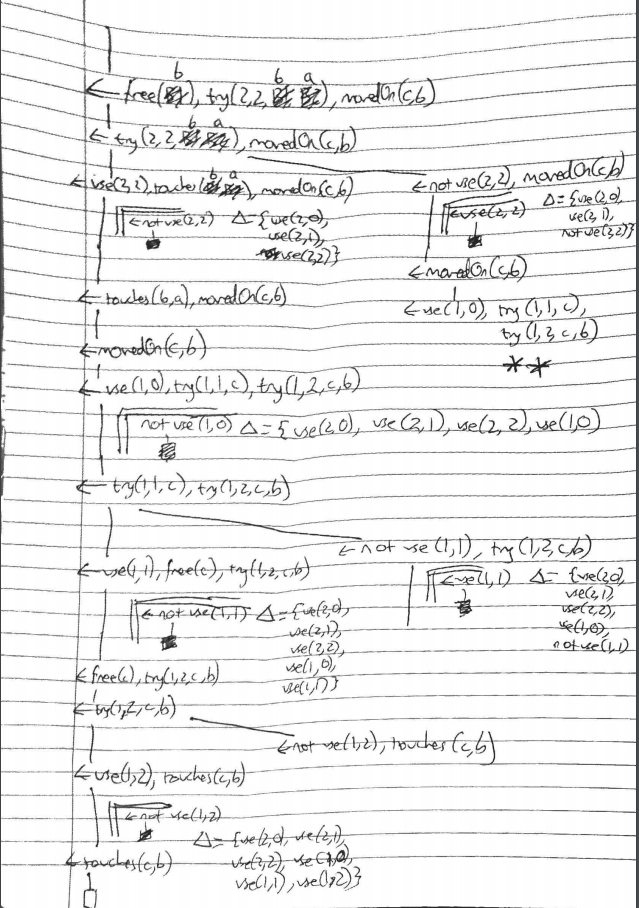


For the body predicates inferred, in the tutorial she just says “It is easy to see that the set of consequences of B ∪ {¬e} that conform with the mode bode declarations is…” in both examples, but in my coursework she put -3 marks and “how did you get this” so does anyone know?



I started to do the proof below but it got way too long, is there a quicker way?





ii)

movedOn(B1, B2) <- free(B1), touches(B1, B2)

onTopOf(B1, B2)

iii)

Yes. I think.

No, because you need to learn 2 rules in one iteration of the coverage loop, which can't be done because it has an incomplete startset.

2.

a)

i)

T =

sentence(V1, V2) <- body([V1, V2], [(m1, [], [])]).

nominal(V1, V2) <- body([V1, V2], [(m2, [], [])]).

body(InputSoFar, RulesSoFar) <-

vp(V1, V2),

link([V1, V2], InputSoFar, Links),

append(RuleSoFar, (m3, [], Links), NewRule),

append(InputSoFar, [], NewInputs),

body(NewInputs, NewRule).

body(InputSoFar, RuleSoFar) <-

noun(V1, V2),

link([V1, V2], InputSoFar, Links),

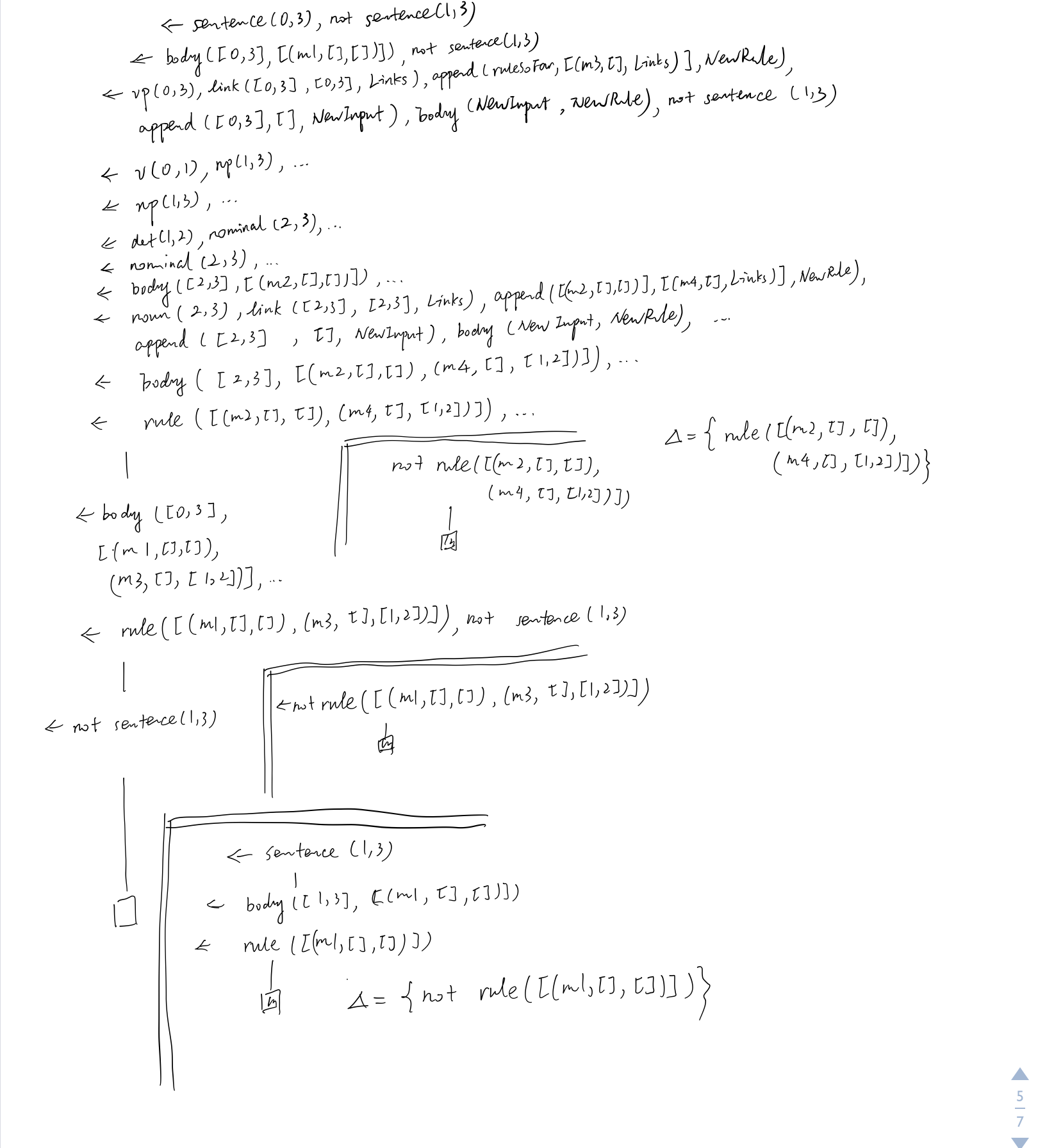
append(RuleSoFar, (m4, [], Links), NewRule),

append(InputSoFar, [], NewInputs),

body(NewInputs, NewRule).

body(InputSoFar, RuleSoFar) <- rule(RuleSoFar).

ii)



b)

To get the bottom set we see what is entailed by B U {ᄀe+}.

B U {ᄀe+} ⊨ {pos(0). pos(1). pos(2). pos(3). v(0, 1). det(1, 2). noun(2,3). ᄀsentence(0,3) }

ᄀBot(B, e+) = pos(0) ⋀ pos(1) ⋀ pos(2) ⋀ pos(3) ⋀ v(0, 1) ⋀ det(1, 2) ⋀ noun(2,3) ⋀ ᄀsentence(0,3).

Bot(B, e+) = sentence(0, 3) <- pos(0), pos(1), pos(2), pos(3), v(0, 1), det(1, 2), noun(2,3).

It is clear to see that H does not subsume the bottom set, so H is not derivable by bottom generalisation.

To get the head of the kernel set we solve abductive task <B, Ab, IC>

Where Ab = {

sentence(0, 0), sentence(0, 1), sentence(0, 2), sentence(0, 3),

sentence(1, 0), sentence(1, 1), sentence(1, 2), sentence(1, 3),

sentence(2, 0), sentence(2, 1), sentence(2, 2), sentence(2, 3),

sentence(3, 0), sentence(3, 1), sentence(3, 2), sentence(3, 3),

nominal(0, 0), nominal(0, 1), nominal(0, 2), nominal(0, 3),

nominal(1, 0), nominal(1, 1), nominal(1, 2), nominal(1, 3),

nominal(2, 0), nominal(2, 1), nominal(2, 2), nominal(2, 3),

nominal(3, 0), nominal(3, 1), nominal(3, 2), nominal(3, 3),

}

And IC is the logical consistency constraints.

From this we get the head is sentence(0,3).

To get the body, we see what is entailed by B U {ᄀe+}. It is easy to see that

B U {ᄀe+} ⊨ {pos(0). pos(1). pos(2). pos(3). v(0, 1). det(1, 2). noun(2,3). ᄀsentence(0,3)}

So the kernel set K =

sentence(0,3) <- pos(0).

sentence(0,3) <- pos(1).

sentence(0,3) <- pos(2).

sentence(0,3) <- pos(3).

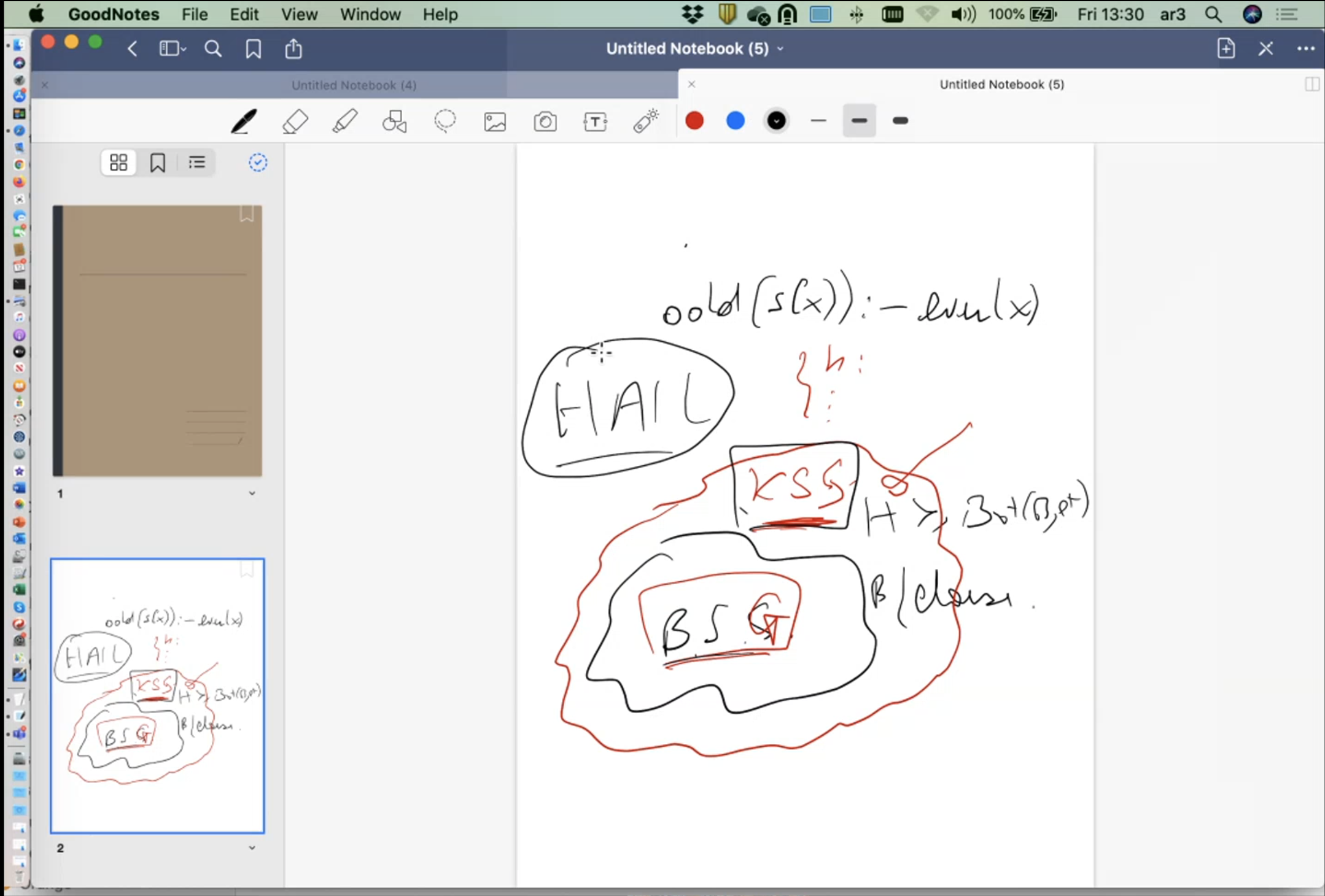
sentence(0,3) <- v(0, 1).

sentence(0,3) <- det(1,2).

sentence(0,3) <- noun(2,3).

Clearly H does not subsume K so H is not derivable by Kernel Set Subsumption.

I think we could also have started with showing that H is not derivable by Kernel Set Subsumption, which allows us to say for free that H is not derivable by Bottom Generalisation neither as Bottom Generalisation is a subset of Kernel Set Subsumption. (see revision lecture:)



3.

a)

i)

ground(B U H) =

q(1) <- not p(1), s(1).

q(2) <- not p(2), s(2).

r(1, a).

r(2, b).

s(1).

s(2).

t(a).

t(b).

p(1) <- not q(1), not r(1, a), s(1).

p(2) <- not q(2), not r(2, a), s(2).

Since the grounding contains r(1, a) and, not r(1, a) is in the body of a rule, we know the head of that rule can never be satisfied so no answer set will contain p(1). Since we have not p(1) and s(1) we also know that every answer set must contain q(1) because the body of that rule is always satisfied. All the ground literal heads with no body [r(1,a), r(2,b), s(1), s(2), t(a), t(b)] are in every answer set because that’s always true.

p(X) and q(X) can’t both be true at the same time for any X because of the first rule in B and the rule in H and the fact that s(1) and s(2) are always true. So the answer set either contains p(2) or q(2).

So the possible answer sets are:

1: {r(1,a), r(2,b), s(1), s(2), t(a), t(b), q(1), p(2)}

2: {r(1,a), r(2,b), s(1), s(2), t(a), t(b), q(1), q(2)}

\*\*\*no proof required but I included the explanation to be helpful\*\*\*

ii)

Yes because there is at least one answer set (the first one) where q(1) is true and q(2) is false.

iii)

No because there is an answer set where q(2) is not false (the second one). For it to be a cautious inductive solution, the positive examples must be true in every answer set and the negative examples false in every answer set.

b) I checked this in clingo so I think it’s right :-)

i)

p(X) :- s(X).

p(X) :- s(X), not q(X).

p(X) :- s(X), not r(X, C).

p(X) :- s(X), not q(X), not r(X, C).

ii)

% background

q(X) :- not p(X), s(X).

r(1, a).

r(2, b).

s(1).

s(2).

t(a).

t(b).

% S\_M

p(X) :- s(X), rule(1).

p(X) :- s(X), not q(X), rule(2).

p(X) :- s(X), not r(X, C), rule(3, C).

p(X) :- s(X), not q(X), not r(X, C), rule(4, C).

#minimize[rule(1)=1, rule(2)=2, rule(3, a)=2, rule(3, b)=2, rule(4, a)=3, rule(4, b)=3].

{rule(1), rule(2), rule(3, a), rule(3, b), rule(4, a), rule(4, b)}.

goal :- q(1), not q(2).

:- not goal.

iii)

{r(1,a) r(2,b) s(1) s(2) t(a) t(b) rule(3,a) p(2) q(1)}

or

{r(1,a) r(2,b) s(1) s(2) t(a) t(b) rule(2) p(2) q(1)}

c)

i)

1: {b(1), a(1)}

2: {b(1), c(1)}

3: {b(2), a(2)}

4: {b(2), c(2)}

ii)

Not sure if this is right:

<B2, SM, {<Ø, Ø>}, {<Ø, {b(1), b(2)}>}

This says (I think) that there should be at least one answer set and there should be no answer set which doesn’t contain b(1) and doesn’t contain b(2) i.e. every answer set should contain at least one of b(1) b(2).

iii) Again not sure if this is right:

There is no ground literal that is in every answer set so there cannot be a cautious induction task.

4a)i)

Lives(Herman, Berlin)

ii)

Lives(Herman, Germany)

iii)

Lives(Herman, X)

iv)

b) i)a)

Modeb(n, p)

Modeh(n, p)

?

b) ?

ii)a)?

b)?

iii)?

c) i)

<- nate(s(s(0)))

|

<- nate(s(0))

|

<- nate(0)

|

[]

ii)

0.5^3=0.125

iii) N = s(s(... m times s(0) ...))

P(N) = 0.5^(m+1)