

Project #3 Solutions

EECS 592

WN 2020

Updated solutions on 4/28/2020 - problem 10 plan has been changed and is shown in blue.
The additional Room argument was missing on every action.

1. R&N Problem 8.10:

- a. Emily is either a surgeon or a lawyer

$$\text{Occupation}(\text{Emily}, \text{Surgeon}) \vee \text{Occupation}(\text{Emily}, \text{Lawyer})$$

- b. Joe is an actor, but he also holds another job

$$\text{Occupation}(\text{Joe}, \text{Actor}) \wedge [\exists x \text{ Occupation}(\text{Joe}, x) \wedge x \neq \text{Actor}]$$

- c. All surgeons are doctors

$$\forall x \text{ Occupation}(x, \text{Surgeon}) \Rightarrow \text{Occupation}(x, \text{Doctor})$$

- d. Joe does not have a lawyer (i.e. is not a customer of any lawyer)

$$\forall x \text{ Occupation}(x, \text{Lawyer}) \Rightarrow \neg \text{Customer}(\text{Joe}, x)$$

- e. Emily has a boss who is a lawyer

$$\exists x \text{ Boss}(x, \text{Emily}) \wedge \text{Occupation}(x, \text{Lawyer})$$

- f. There exists a lawyer all of whose customers are doctors

$$\exists x \text{ Occupation}(x, \text{Lawyer}) \wedge [\forall y \text{ Customer}(y, x) \Rightarrow \text{Occupation}(y, \text{Doctor})]$$

- g. Every surgeon has a lawyer

$$\forall x \text{ Occupation}(x, \text{Surgeon}) \Rightarrow [\exists y \text{ Customer}(x, y) \wedge \text{Occupation}(y, \text{Lawyer})]$$

2. R&N Problem 8.20:

For this problem we assume we are working with natural numbers and equality is defined. Equality can be defined as $\forall x, y \ x = y \iff \neg(x < y) \wedge \neg(y < x)$ and natural number axioms are shown in the book (8.3.3).

- a. "x is an even number"

$$\forall x \text{ Even}(x) \iff \exists y \ x = y + y$$

b. "x is prime"

$$\forall x \text{ Prime}(x) \iff [(1 < x) \wedge (\forall y, z \ x = y * z \Rightarrow (y = 1) \vee (z = 1))]$$

c. Golbach's conjecture

$$\forall x [Even(x) \wedge (2 < x)] \Rightarrow [\exists y, z \ Prime(y) \wedge Prime(z) \wedge x = y + z]$$

3. R&N Problem 9.6:

Horse(x) : x is a horse

Cow(x) : x is a cow

Pig(x) : x is a pig

Mammal(x) : x is a mammal

Offspring(x, y) : x is the offspring of y

Parent(x, y) : x is the parent of y

a. Horses, cows, and pigs are mammals

$$\forall x \text{ Horse}(x) \Rightarrow \text{Mammal}(x)$$

$$\forall x \text{ Cow}(x) \Rightarrow \text{Mammal}(x)$$

$$\forall x \text{ Pig}(x) \Rightarrow \text{Mammal}(x)$$

b. An offspring of a horse is a horse

$$\forall x, y [Offspring(x, y) \wedge \text{Horse}(y)] \Rightarrow \text{Horse}(x)$$

c. Bluebeard is a horse

$$\text{Horse(Bluebeard)}$$

d. Bluebeard is Charlie's parent

$$\text{Parent(Bluebeard, Charlie)}$$

e. Offspring and parent are inverse relations

$$\forall x, y \text{ Parent}(x, y) \iff \text{Offspring}(y, x)$$

f. Every mammal has a parent

$$\forall x \text{ Mammal}(x) \Rightarrow \exists y \text{ Parent}(y, x)$$

4. R&N Problem 9.13:

- a. Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query $\exists h \text{ Horse}(h)$, where clauses are matched in the order given.

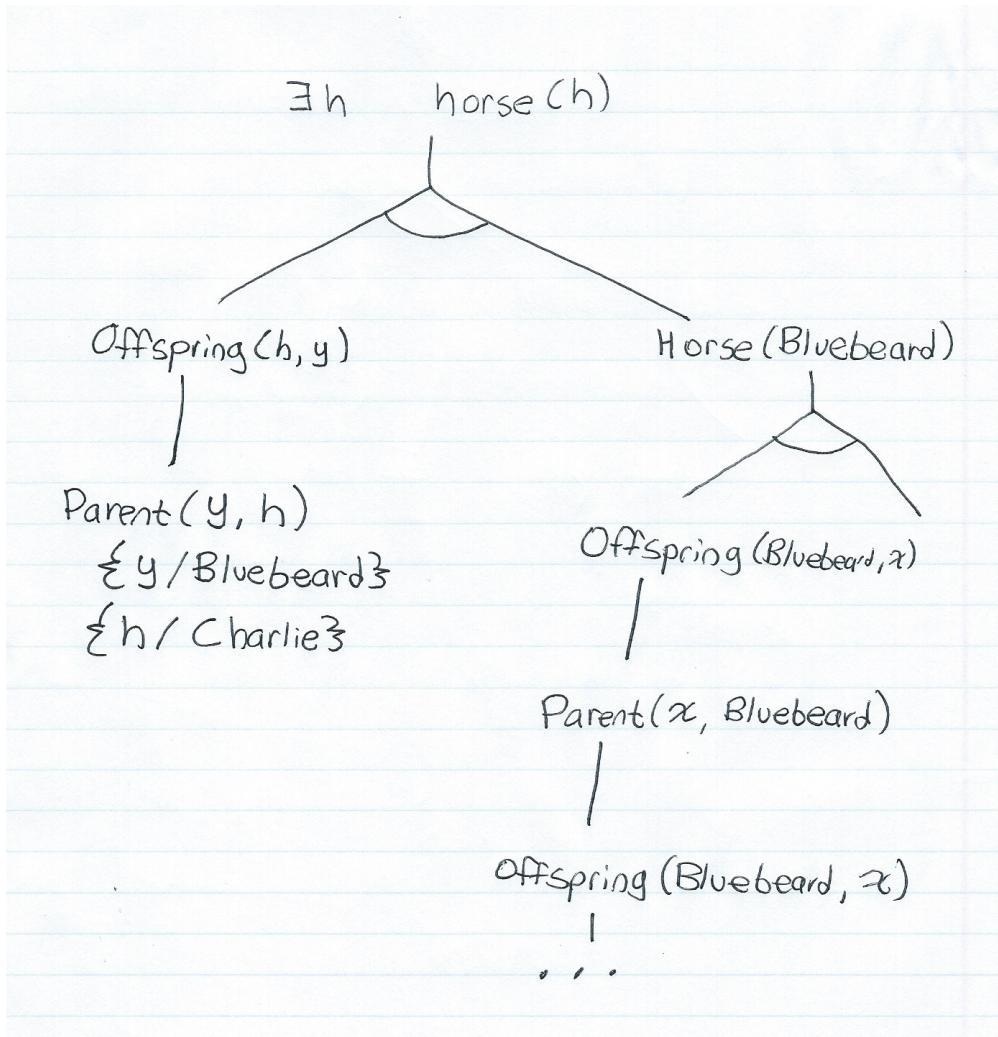


Figure 1: Backwards chaining for problem 4

- b. What do you notice about this domain?

The domain is recursive and the backwards chaining algorithm will never finish since it is stuck in a cycle.

- c. How many solutions for h actually follow from your sentences?

Based on our knowledge base there are 2 solutions, *Charlie* and *Bluebeard*.

- d. Can you think of a way to find all of them?

Ref. [1] details methods and proves their optimality and completeness for finding multiple solutions to a repeating inference problem. Theorems 3.1 and 3.5 handle the single answer and multiple answer cases respectively. The main idea is that repeated states can be pruned from the search tree, since they will never provide an answer.

In our tree, $Offspring(Bluebeard, x)$ appearing a second time would count as a repeated state, causing the algorithm to recurse back up to $Horse(Bluebeard)$ and try a different implication to satisfy it, which in this case would be $Horse(Bluebeard)$ exactly and then $Horse(Charlie)$ would be solved for. On an additional call (via higher level control), the goal could be satisfied directly with $Horse(Bluebeard)$ since that would be next statement in the specified order and thus we would get our 2 solutions.

5. R&N Problem 9.23: From "Horses are animals," it follows that "The head of a horse is the head of an animal." Demonstrate that this inference is valid by carrying out the following steps

$HeadOf(h, x) : h \text{ is the head of } x$
 $Horse(x) : x \text{ is a Horse}$
 $Animal(x) : x \text{ is an animal}$

- a. Translate the premise and the conclusion into the language of first-order logic.

Premise:

$$\forall x \ Horse(x) \Rightarrow Animal(x)$$

Conclusion:

$$\forall x, h \ Horse(x) \wedge HeadOf(h, x) \Rightarrow [\exists a \ Animal(a) \wedge HeadOf(h, a)]$$

6. Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion

$Hound(x) : x \text{ is a hound}$
 $Howl(x) : x \text{ howls at night}$
 $Cat(x) : x \text{ is a cat}$
 $Mouse(x) : x \text{ is a mouse}$
 $Has(x, y) : x \text{ has a } y$
 $LightSleeper(x, y) : x \text{ is a light sleeper}$
 $Sam : \text{constant}$

- a. All hounds howl at night

$$\forall x \ Hound(x) \Rightarrow Howl(x)$$

$$\neg Hound(x) \vee Howl(x)$$

$\mathbf{R}_1 : \neg Hound(x) \vee Howl(x)$

b. Anyone who has any cats will not have any mice

$$\forall r, c, m \ Has(r, c) \wedge Cat(c) \wedge Mouse(m) \Rightarrow \neg Has(r, m)$$

$$\neg Has(r, c) \vee \neg Cat(c) \vee \neg Mouse(m) \vee \neg Has(r, m)$$

$$\mathbf{R}_2 : \neg Has(r, c) \vee \neg Cat(c) \vee \neg Mouse(m) \vee \neg Has(r, m)$$

c. Light sleepers do not have anything that howls at night

$$\forall q, h \ LightSleeper(q) \wedge Howl(h) \Rightarrow \neg Has(q, h)$$

$$\neg LightSleeper(q) \vee \neg Howl(h) \vee \neg Has(q, h)$$

$$\mathbf{R}_3 : \neg LightSleeper(q) \vee \neg Howl(h) \vee \neg Has(q, h)$$

d. Sam has either a cat or a hound

$$\exists p \ Has(Sam, p) \wedge (Cat(p) \vee Hound(p))$$

$$Has(Sam, C_1) \wedge (Cat(C_1) \vee Hound(C_1))$$

$$\mathbf{R}_4 : Has(Sam, C_1)$$

$$\mathbf{R}_5 : Cat(C_1) \vee Hound(C_1)$$

e. (Conclusion) If Sam is a light sleeper, then Sam does not have any mice

$$\forall n \ LightSleeper(Sam) \wedge Mouse(n) \Rightarrow \neg Has(Sam, n)$$

$$\neg LightSleeper(Sam) \vee \neg Mouse(n) \vee \neg Has(Sam, n)$$

\neg Conclusion:

$$\neg [\neg LightSleeper(Sam) \vee \neg Mouse(n) \vee \neg Has(Sam, n)]$$

$$LightSleeper(Sam) \wedge Mouse(n) \wedge Has(Sam, n)$$

$$\mathbf{R}_6 : LightSleeper(Sam)$$

$$\mathbf{R}_7 : Mouse(n)$$

$$\mathbf{R}_8 : Has(Sam, n)$$

Resolution steps to prove conclusion shown in Figure 2.

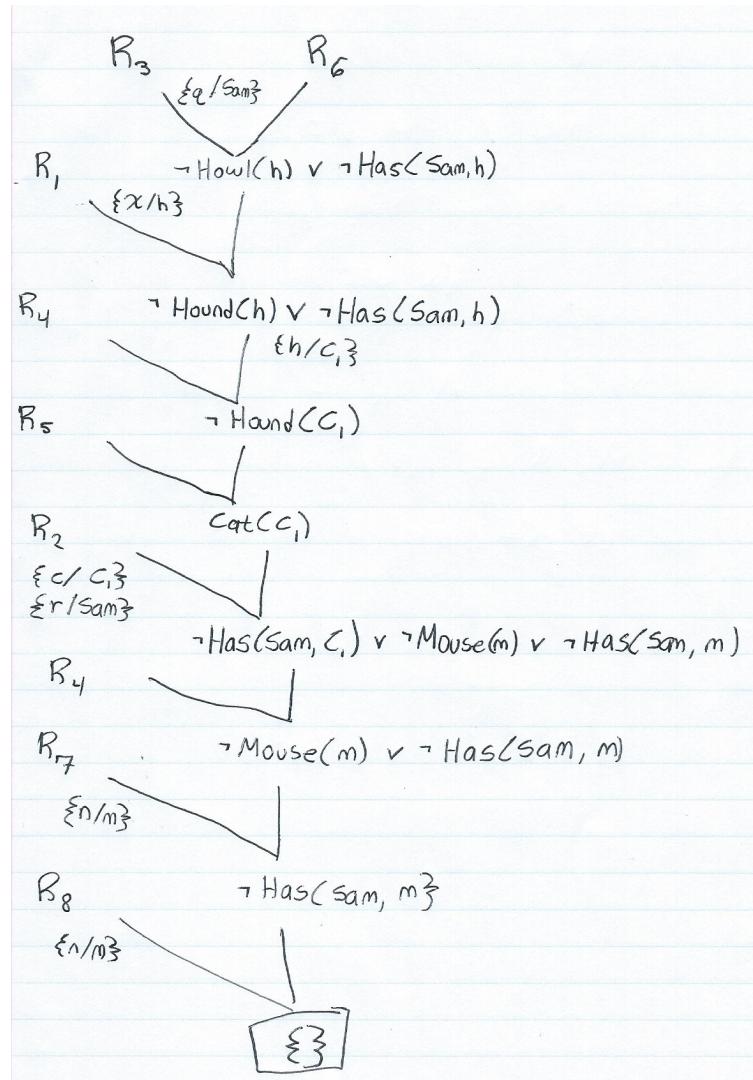


Figure 2: Resolution steps for problem 6

7. Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion

$Has(x, y) : x \text{ has a } y$
 $FeelWarm(x) : x \text{ feels warm}$
 $Drunk(x) : x \text{ is drunk}$
 $IsWarm(x) : x \text{ is warm}$
 $Costume(x) : x \text{ is a costume}$
 $Furry(x) : x \text{ is furry}$
 $AI(x) : x \text{ is an AI student}$
 $CS(x) : x \text{ is a CS student}$
 $Robot(x) : x \text{ is a robot costume}$

- a. Everyone who feels warm either is drunk, or every costume they have is warm

$$\begin{aligned} \forall x, y \, FeelWarm(x) &\Rightarrow [Drunk(x) \vee (Has(x, y) \wedge Costume(y) \Rightarrow IsWarm(y))] \\ \neg FeelWarm(x) \vee [Drunk(x) \vee (\neg Has(x, y) \vee \neg Costume(y) \vee IsWarm(y))] \\ \boxed{\mathbf{R}_1 : \neg FeelWarm(x) \vee Drunk(x) \vee \neg Has(x, y) \vee \neg Costume(y) \vee IsWarm(y)} \end{aligned}$$

- b. Every costume that is warm is furry

$$\begin{aligned} \forall c \, Costume(c) \wedge IsWarm(c) &\Rightarrow Furry(c) \\ \boxed{\mathbf{R}_2 : \neg Costume(c) \vee \neg IsWarm(c) \vee Furry(c)} \end{aligned}$$

- c. Every AI student is a CS student

$$\begin{aligned} \forall s \, AI(s) &\Rightarrow CS(s) \\ \boxed{\mathbf{R}_3 : \neg AI(s) \vee CS(s)} \end{aligned}$$

- d. Every AI student has some robot costume

$$\begin{aligned} \forall p, \exists r \, AI(p) &\Rightarrow [Has(p, r) \wedge Robot(r)] \\ \neg AI(p) \vee [Has(p, Rof(p)) \wedge Robot(Rof(p))] \\ (\neg AI(p) \vee Has(p, Rof(p))) \wedge (\neg AI(p) \vee Robot(Rof(p))) \\ \boxed{\mathbf{R}_4 : \neg AI(p) \vee Has(p, Rof(p))} \\ \boxed{\mathbf{R}_5 : \neg AI(p) \vee Robot(Rof(p))} \end{aligned}$$

Also, we need to say every robot costume is a costume (to make things work).

$$\forall q, \ Robot(q) \Rightarrow Costume(q)$$

$$\mathbf{R}_6 : \neg Robot(q) \vee Costume(q)$$

e. No robot costume is furry

$$\forall m, \ Robot(m) \Rightarrow \neg Furry(m)$$

$$\mathbf{R}_7 : \neg Robot(m) \vee \neg Furry(m)$$

f. (Conclusion) If every CS student feels warm, then every AI student is drunk

$$[\forall n CS(n) \Rightarrow FeelWarm(n)] \Rightarrow [\forall t AI(t) \Rightarrow Drunk(t)]$$

$$[\forall n \neg CS(n) \vee FeelWarm(n)] \Rightarrow [\forall t \neg AI(t) \vee Drunk(t)]$$

$$\neg [\forall n \neg CS(n) \vee FeelWarm(n)] \vee [\forall t \neg AI(t) \vee Drunk(t)]$$

\neg Conclusion:

$$[\forall n \neg CS(n) \vee FeelWarm(n)] \wedge \neg [\forall t \neg AI(t) \vee Drunk(t)]$$

$$[\forall n \neg CS(n) \vee FeelWarm(n)] \wedge [\exists t AI(t) \wedge \neg Drunk(t)]$$

$$[\neg CS(n) \vee FeelWarm(n)] \wedge AI(T_1) \wedge \neg Drunk(T_1)$$

$$\mathbf{R}_8 : \neg CS(n) \vee FeelWarm(n)$$

$$\mathbf{R}_9 : AI(T_1)$$

$$\mathbf{R}_{10} : \neg Drunk(T_1)$$

Resolution steps to prove conclusion shown in Figure 3.

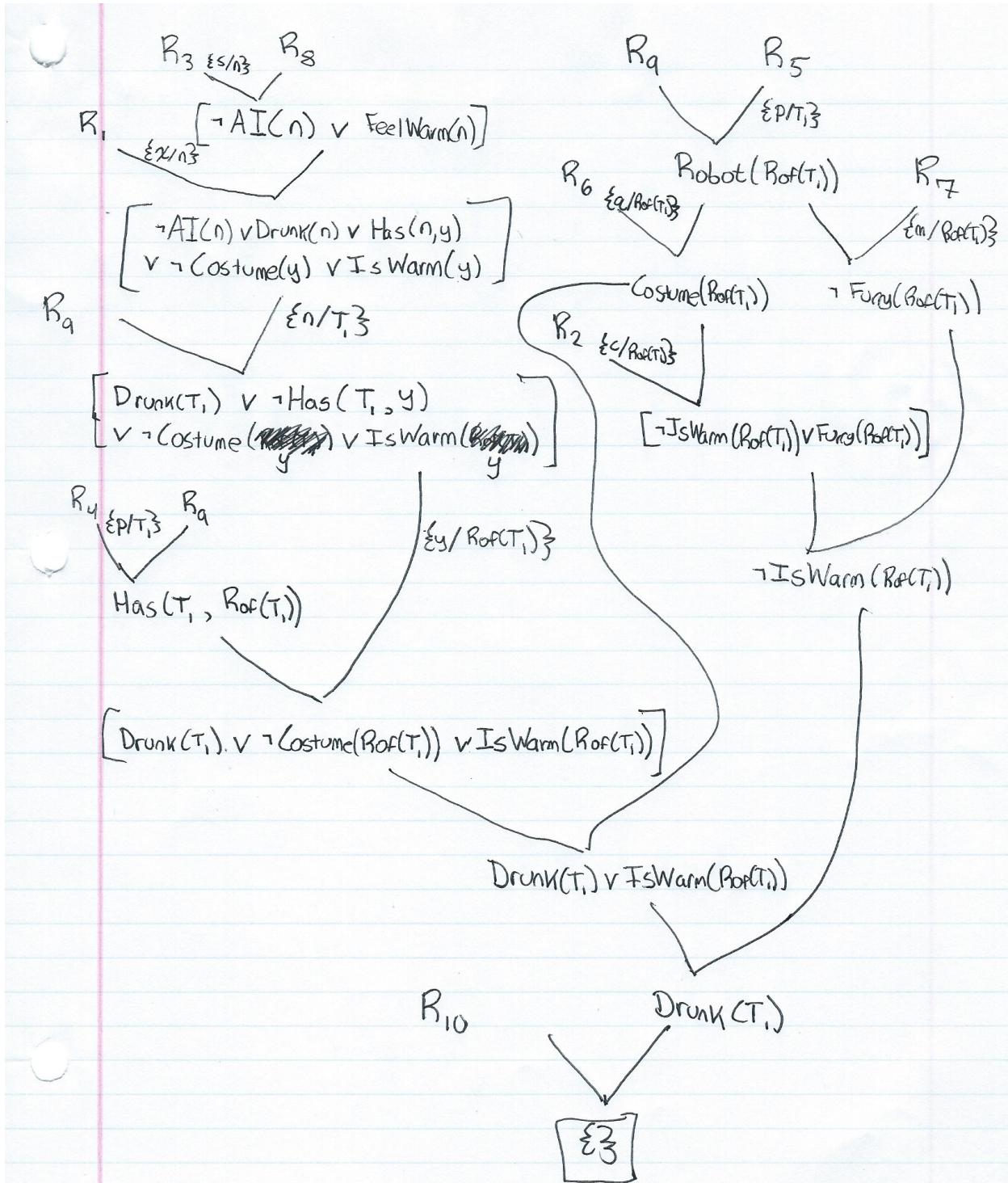


Figure 3: Resolution steps for problem 7

8. Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion

$Child(x)$: x is a child
 $Loves(x, y)$: x loves y
 $Candy(x)$: x is a candy
 $Nutr(x)$: x is a nutrition fanatic
 $Eats(x, y)$: x eats y
 $Pumpkin(x)$: x is a pumpkin
 $Buys(x, y)$: x buys y
 $Carves(x, y)$: x carves y
 $Stuart$: constant
 $Lifesavers$: constant

- a. Every child loves every candy

$$\begin{aligned} \forall x, c \ Child(x) \wedge Candy(c) &\Rightarrow Loves(x, c) \\ \boxed{\mathbf{R}_1 : \neg Child(x) \vee \neg Candy(c) \vee Loves(x, c)} \end{aligned}$$

- b. Anyone who loves some candy is not a nutrition fanatic

$$\begin{aligned} \forall a \ [\exists b \ Candy(b) \wedge Loves(a, b)] &\Rightarrow \neg Nutr(a) \\ \forall a \ \neg [\exists b \ Candy(b) \wedge Loves(a, b)] \vee \neg Nutr(a) \\ \forall a \ [\forall b \ \neg Candy(b) \vee \neg Loves(a, b)] \vee \neg Nutr(a) \\ \boxed{\mathbf{R}_2 : \neg Candy(b) \vee \neg Loves(a, b) \vee \neg Nutr(a)} \end{aligned}$$

- c. Anyone who eats any pumpkin is a nutrition fanatic

$$\begin{aligned} \forall y, p \ Pumpkin(p) \wedge Eats(y, p) &\Rightarrow Nutr(y) \\ \boxed{\mathbf{R}_3 : \neg Pumpkin(p) \vee \neg Eats(y, p) \vee Nutr(y)} \end{aligned}$$

- d. Anyone who buys any pumpkin either carves it or eats it

$$\begin{aligned} \forall z, q \ Pumpkin(q) \wedge Buys(z, q) &\Rightarrow [Carves(z, q) \vee Eats(z, q)] \\ \boxed{\mathbf{R}_4 : \neg Pumpkin(q) \vee \neg Buys(z, q) \vee Carves(z, q) \vee Eats(z, q)} \end{aligned}$$

e. Stuart buys a pumpkin

$$\exists r \text{ } Pumpkin(r) \wedge Buys(Stuart, r)$$

$$Pumpkin(R_1) \wedge Buys(Stuart, R_1)$$

$\mathbf{R}_5 : Pumpkin(R_1)$

$\mathbf{R}_6 : Buys(Stuart, R_1)$

f. Lifesavers is a candy

$\mathbf{R}_7 : Candy(Lifesavers)$

g. (Conclusion) If Stuart is a child, then Stuart carves some pumpkin

$$Child(Stuart) \Rightarrow [\exists d \text{ } Pumpkin(d) \wedge Carves(Stuart, d)]$$

$$\neg Child(Stuart) \vee [\exists d \text{ } Pumpkin(d) \wedge Carves(Stuart, d)]$$

\neg Conclusion:

$$Child(Stuart) \wedge \neg [\exists d \text{ } Pumpkin(d) \wedge Carves(Stuart, d)]$$

$$Child(Stuart) \wedge [\forall d \neg Pumpkin(d) \vee \neg Carves(Stuart, d)]$$

$\mathbf{R}_8 : Child(Stuart)$

$\mathbf{R}_9 : \neg Pumpkin(d) \vee \neg Carves(Stuart, d)$

Resolution steps to prove conclusion shown in Figure 4.

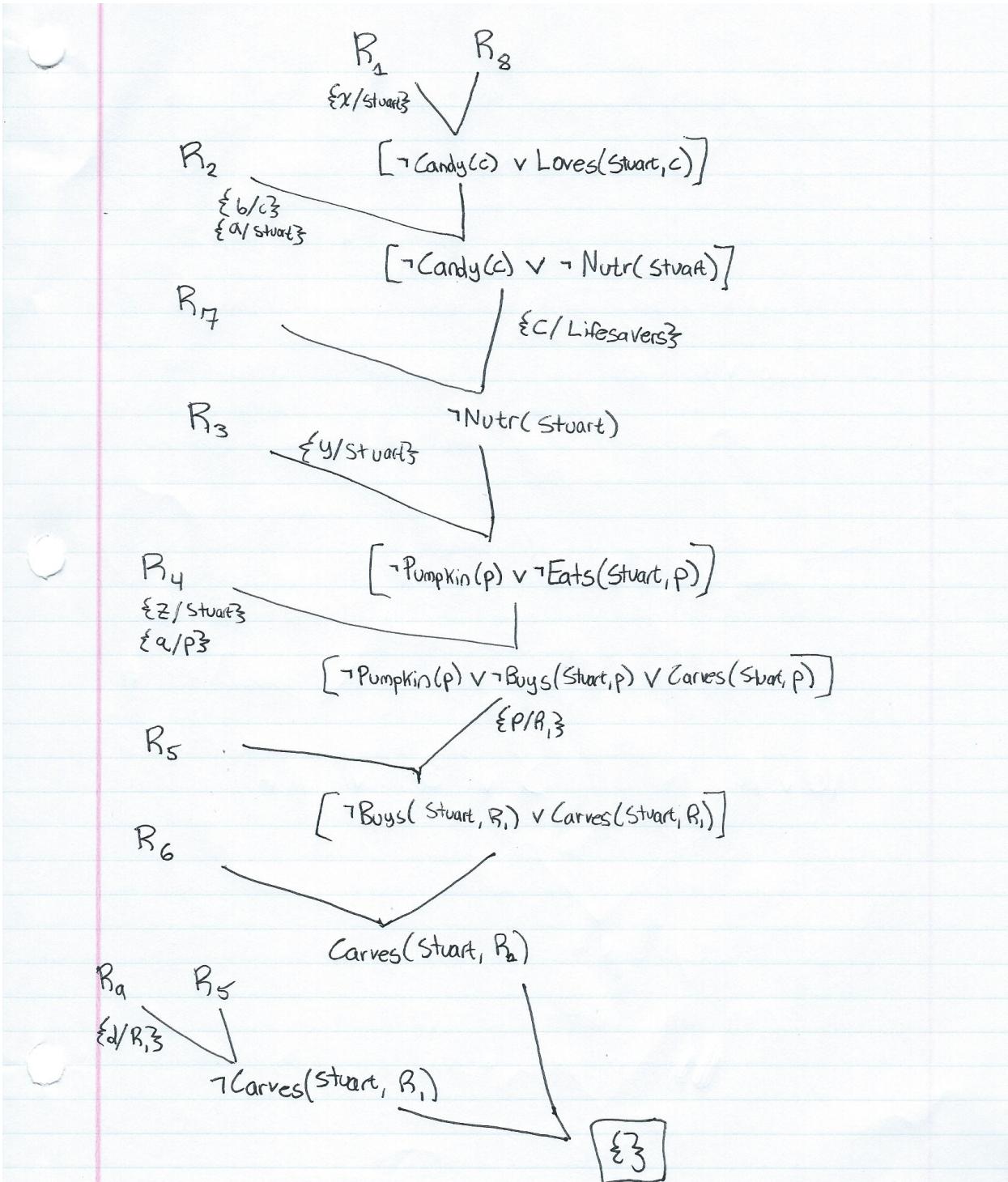


Figure 4: Resolution steps for problem 8

9. R&N 10.2 (Flying)

$Fly(P1, JFK, JFK)$
 $Fly(P1, JFK, SFO)$
 $Fly(P2, SFO, SFO)$
 $Fly(P2, SFO, JFK)$

10. R&N 10.4 (Shakey the Robot)

$Action(Go(x, y, r)),$
 PRECOND : $At(Shakey, x) \wedge In(x, r) \wedge In(y, r)$
 EFFECT : $At(Shakey, y) \wedge \neg At(Shakey, x))$
 $Action(Push(b, x, y, r),$
 PRECOND : $At(Shakey, x) \wedge Pushable(b) \wedge At(b, x) \wedge In(x, r) \wedge In(y, r)$
 EFFECT : $At(b, y) \wedge At(Shakey, y) \wedge \neg At(b, x) \wedge \neg At(Shakey, x))$
 $Action(ClimbUp(x, b),$
 PRECOND : $At(Shakey, x) \wedge At(b, x) \wedge Climbable(b) \wedge On(Shakey, Floor)$
 EFFECT : $On(Shakey, b) \wedge \neg On(Shakey, Floor))$
 $Action(ClimbDown(b, x),$
 PRECOND : $On(Shakey, b)$
 EFFECT : $On(Shakey, Floor) \wedge \neg On(Shakey, b))$
 $Action(TurnOn(s, b, x),$
 PRECOND : $On(Shakey, b) \wedge At(Shakey, x) \wedge At(s, x) \wedge TurnedOff(s)$
 EFFECT : $TurnedOn(s) \wedge \neg TurnedOff(s)$
 $Action(TurnOff(s, b, x),$
 PRECOND : $On(Shakey, b) \wedge At(Shakey, x) \wedge At(l, x) \wedge TurnedOn(s)$
 EFFECT : $TurnedOff(s) \wedge \neg TurnedOn(s)$

Initial State:

$In(Switch_1, Room_1) \wedge In(Door_1, Room_1) \wedge In(Door_1, Corridor)$
 $In(Switch_2, Room_2) \wedge In(Door_2, Room_2) \wedge In(Door_2, Corridor)$
 $In(Switch_3, Room_3) \wedge In(Door_3, Room_3) \wedge In(Door_3, Corridor)$
 $In(Switch_4, Room_4) \wedge In(Door_4, Room_4) \wedge In(Door_4, Corridor)$
 $In(Shakey, Room_3) \wedge At(Shakey, X_{Start}) \wedge On(Shakey, Floor)$
 $In(Box_1, Room_1) \wedge In(Box_2, Room_1) \wedge In(Box_3, Room_1) \wedge In(Box_4, Room_1)$
 $Climbable(Box_1) \wedge Climbable(Box_2) \wedge Climbable(Box_3) \wedge Climbable(Box_4)$
 $Pushable(Box_1) \wedge Pushable(Box_2) \wedge Pushable(Box_3) \wedge Pushable(Box_4)$
 $At(Box_1, X_1) \wedge At(Box_2, X_2) \wedge At(Box_3, X_3) \wedge At(Box_4, X_4)$
 $TurnedOn(Switch_1) \wedge TurnedOff(Switch_2) \wedge TurnedOff(Switch_3) \wedge TurnedOn(Switch_4)$

Plan:

*Go(X_S, Door₃, Room₃)
Go(Door₃, Door₁, Corridor)
Go(Door₁, X₂, Room₁)
Push(Box₂, X₂, Door₁, Room₁)
Push(Box₂, Door₁, Door₂, Corridor)
Push(Box₂, Door₂, Switch₂, Room₂)*

References

- [1] D. E. Smith, M. R. Genesereth, and M. L. Ginsberg, “Controlling recursive inference,” *Artificial Intelligence*, vol. 30, no. 3, pp. 343–389, 1986.