XINRUI ZHANG EECS 592 Project 3. zxinrui

1. R&N Problem 8.10

a. $Occupation(Emily, Surgeon) \lor Occupation(Emily, Lawyer)$

b. $Occupation(Joe, Actor) \lor [\exists x\ Occupation(Joe, x) \land \lnot(x = Actor)]$

c. $\forall x\ Occupation(x, Surgeon) \Rightarrow Occupation(x, Doctor)$

d. $\lnot \exists x\ Customer(Joe, x) \land Occupation(x, Lawyer)$

e. $\exists x\ Boss(x, Emily) \land Occupation(x, Lawyer)$

f. $\exists x\ Occupation(x, Lawyer) \land (\forall y\ Customer(y, x) \Rightarrow Occupation(y, Doctor))$

g. $\forall x\ \exists y\ Occupation(x, Surgeon) \Rightarrow Customer(x, y) \land Occupation(y, Lawyer)$

2. R&N Problem 8.22

$$[\forall k\ Key(k) \Rightarrow (\exists t_1\ Lost(k, t_1) \land Before(Now, t_1))]$$
$$\land [\forall S_1 \forall S_2\ Sock(S_1) \land Sock(S_2) \land Pair(S_1, S_2) \Rightarrow [\exists t_2 (Lost(S_1, t_2) \lor Lost(S_2, t_2)) \land Before(Now, t_2)]]$$

3. Only b is legitimate.

a uses Everest which is in KB.

c applied twice Existential Instantiation which is not allowed.

XINRUI ZHANG EECS 592 Project 3 ZXinrui

4. R&N Problem 9.6

a. $\forall x$ Horse (x) $\vee$ Cow (x) $\vee$ Pig (x) $\Rightarrow$ Mammal (x)

b. $\forall x \forall y$ Horse (x) $\wedge$ Offspring (y, x) $\Rightarrow$ Horse (y)

y is x's offspring

c. Horse (Bluebeard)

d. Parent (Bluebeard, Charlie) Bluebeard is Charlie's parent.

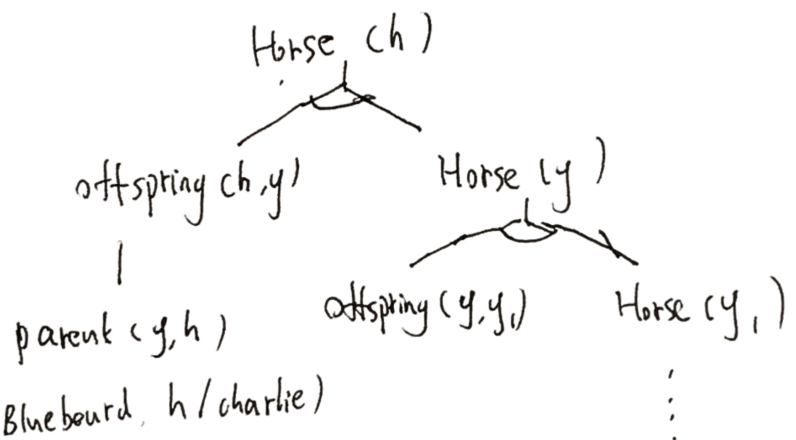
e. $\forall x \forall y$ Offspring (x, y) $\Leftrightarrow$ Parent (y, x)

f. $\forall x$ Mammal (x) $\Rightarrow$ ($\exists$ y Parent (y, x)

y is x's parent

5. R&N Problem 9.13

a.



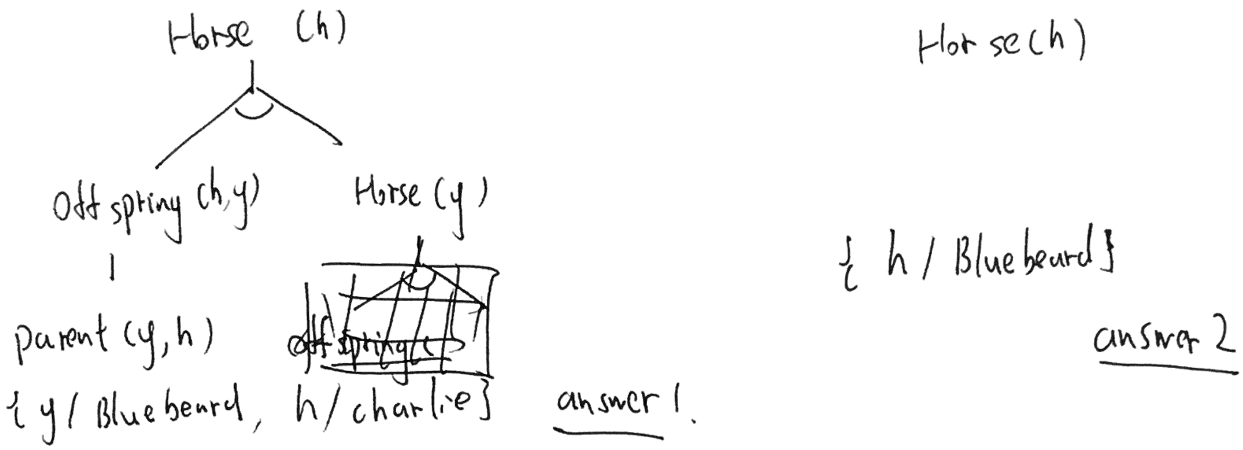
~~infinitely~~ infinite loop

b. Because of this infinite loop, backward ~~chaining~~ chaining is not able to solve this problem.

c. {h/ Bluebeard} and {h/ Charlie}

d. If the portion of the space is redundant which means it will not produce any novel answers to the original problem, the portion can be eliminate.

According to Corollary 3.4: The depth of repetition in a search space can be limited to one less than the total number of answers desired for the problem.



6.

a. $\forall x$ ~~Haws~~ $Hounds(x) \Rightarrow Howlatnight(x)$

b. $\forall x \forall y \; Cat(y) \wedge Have(x,y) \Rightarrow \neg(\exists z \; Mouse(z) \wedge Have(x,z))$

c. $\forall x \; Lightsleeper(x) \Rightarrow \neg(\exists y \; Have(x,y) \wedge Howlatnight(y))$

d. $\exists x \; Have(Sam, x) \wedge (cat(x) \vee Hound(x))$

e. $Lightsleeper(Sam) \Rightarrow \neg(\exists z \; Mouse(z) \wedge Have(Sam, z))$

6. CNF

a. $\neg$ Hound (x) $\vee$ Howlatnight (x)

b. $\neg$ cat (y) $\vee$ $\neg$ Have (x, y) $\vee$ $\neg$ Mouse (z) $\vee$ $\neg$ Have (x, z)

c. $\neg$ Lightsleeper (x) $\vee$ $\neg$ Have (x, y) $\vee$ $\neg$ Howlatnight (y)

d. Have (Sam, X1) $\wedge$ (cat (X1) $\vee$ Hound (X1))

$\neg$e: Lightsleeper (Sam) $\wedge$ Mouse (X2) $\wedge$ Have (Sam, X2)

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

$\neg$ cat (y) $\vee$ $\neg$ Have (x, y) $\vee$ $\neg$ Mouse (z) $\vee$ $\neg$ Have (x, z)

$\neg$ Lightsleeper (x) $\vee$ $\neg$ Have (x, y) $\vee$ $\neg$ Howlatnight (y)

Have (Sam, X1)

cat (X1) $\vee$ Hound (X1)

Lightsleeper (Sam)

Mouse (X2)

Have (Sam, X2)

cat (X1) $\vee$ Howlatnight (X1)

$\neg$ cat (y) $\vee$ $\neg$ Have (x, y) $\vee$ $\neg$ Have (x, X2)

$\neg$ cat (y) $\vee$ $\neg$ Have (Sam, y)

Howlatnight (X1) $\vee$ $\neg$ Have (Sam, X1).

$\neg$ Have (Sam, y) $\vee$ $\neg$ Howlatnight (y)

Howlatnight (X1)

$\neg$ Howlatnight (X1)

empty

conclusion proved

XINRUI ZHANG GGCS 592 Project 3 zxinrui

7\. a. $\forall x\ Feelwarm(x) \Rightarrow Drunk(x) \vee (\forall y\ costume(y) \wedge Have(x,y) \Rightarrow warm(y))$

b. $\forall x\ Costume(x) \Rightarrow (Warm(x) \Rightarrow furry(x))$

c. $\forall x\ AI(x) \Rightarrow CS(x)$

d. $\forall x\ AI(x) \Rightarrow (\exists y\ Costume(y) \wedge Robot(y) \wedge Have(x,y))$

e. $\neg(\exists x\ costume(x) \wedge Robot(x) \wedge Furry(x))$

f. $(\forall x\ CS(x) \Rightarrow Feelwarm(x)) \Rightarrow (\forall y\ AI(y) \Rightarrow Drunk(y))$

cnf:

a. $\neg Feelwarm(x) \vee Drunk(x) \vee \neg Costume(y) \vee \neg Have(x,y) \vee warm(y)$

b. $\neg costume(x) \vee \neg Warm(x) \vee Furry(x)$

c. $\neg AI(x) \vee CS(x)$

~~d. ¬AI(x) ∨ Costume(Y1) ∧ Robot~~

d. $(\neg AI(x) \vee Costume(Y1)) \wedge (\neg AI(x) \vee Robot(Y1)) \wedge (\neg AI(x) \vee Have(x, Y1))$

e. $\neg costume(x) \vee \neg Robot(x) \vee \neg Furry(x)$

¬f. $(\neg CS(x) \vee Feelwarm(x)) \wedge AI(Y2) \wedge \neg Drunk(Y2)$

a, b, c, e as above.

d (1) $\neg AI(x) \vee Costume(Y1)$

(2) $\neg AI(x) \vee Robot(Y1)$

(3) $\neg AI(x) \vee Have(X, Y1)$

¬f (1) $\neg Cs(x) \vee Feelwarm(x)$

(2). $AI(Y2)$

(3) $\neg Drunk(Y2)$

XINRUI ZHANG   Project 3   EECS 592   zxinrui

¬f(2) AI(Y2)   d(1) ¬AI(X) ∨ costume(Y1)   d(2) ¬AI(X) ∨ Robot(Y1)

costume(Y1)   e. ¬costume(X) ∨ ¬Robot(X) ∨ ¬Furry(X)   Robot(Y1)

¬Furry(Y1)   b. ¬costume(X) ∨ ¬Warm(X) ∨ Furry(X)

$\boxed{\neg \text{Warm}(Y_1)}$

¬f(2) AI(Y2)   d(3) ¬AI(X) ∨ Have(X, Y1)

a. ¬Feelwarm(X) ∨ Drunk(X) ∨ ¬costume(Y) ∨ ¬Have(X, Y) ∨ warm(Y)

costume(Y1)   Have(Y2, Y1)   ¬f(3) Drunk(Y2)

¬Feelwarm(Y2) ∨ warm(Y1)

¬f(2) AI(Y2)   c. ¬AI(X) ∨ CS(X)

CS(Y2)   ¬f(1) ¬CS(X) ∨ Feelwarm(X)

Feelwarm(Y2).

$\boxed{\text{warm}(Y_1).}$

empty

conclusion proved

XINRUI ZHANG Project 3 EECS 592 zxinrui.

8. a. $\forall x \forall y \quad Child(x) \wedge Candy(y) \Rightarrow loves(x, y)$

b. $\forall x \; [\exists y \; loves(x, y) \wedge Candy(y) \Rightarrow \neg Nufanatic(x)]$

c. $\forall x \; [\exists y \; Eats(x, y) \wedge Pumpkin(y) \Rightarrow Nufanatic(x)]$

d. $\forall x \forall y \quad Buys(x, y) \wedge Pumpkin(y) \Rightarrow Carves(x, y) \vee Eats(x, y)$

e. $\exists x \; Buys(stuart, x) \wedge Pumpkin(x)$

f. ~~Lifesaver~~ $Candy(Lifesavers)$

g. $Child(Stuart) \Rightarrow \exists x \quad Carves(stuart, x) \wedge Pumpkin(x)$

CNF:

a. $\neg child(x) \vee \neg Candy(y) \vee loves(x, y)$

b. $\neg Loves(x, y) \vee \neg Candy(y) \vee \neg Nufanatic(x)$

c. $\neg Eats(x, y) \vee \neg Pumpkin(y) \vee Nufanatic(x)$

d. $\neg Buys(x, y) \vee \neg Pumpkin(y) \vee Carves(x, y) \vee Eats(x, y)$

e. $Buys(stuart, x1) \wedge Pumpkin(x1)$

f. $Candy(lifesavers)$

$\neg$g. $Child(stuart) \wedge (\neg Pumpkin(x) \vee \neg Carves(stuart, x))$

XINRUI ZHANG EECS 592 project 3 zxinrui

a. ¬ Child (x) ∨ ¬ candy (y) ∨ loves (x, y)

b. ¬ loves (x, y) ∨ ¬ candy (y) ∨ ¬ Nutanatic (x)

c. ¬ Eats (x, y) ∨ ¬ Pumpkin (y) ∨ Nutanatic (x)

d. ¬ Buys (x, y) ∨ ¬ Pumpkin (y) ∨ carves (x, y) ∨ Eats (x, y)

e. (1) Buys (stuart, x1)
(2) Pumpkin (x1)

f. Candy (life savers).

¬g. (1) Child (Stuart)
(2). ¬ Pumpkin (x) ∨ ¬ Carves (stuart, x)

b. ¬ loves (x, y) ∨ ¬ candy (y) ∨ ¬ Nutanatic (x). f. candy (Lifesavers)

¬ loves (x, lifesavers) ∨ ¬ Nutanatic (x)

a. ¬ child (x) ∨ ¬ candy (y) ∨ loves (x, y) f. candy (lifesavers).

¬g(1) child (Stuart) ¬ child (x) ∨ loves (x, lifesavers)

loves (stuart, lifesavers) ¬ Nutanatic (Stuart)

¬g(2) ¬ Pumpkin (x) ∨ ¬ Carves (stuart, x) e(2) Pumpkin (x1).

d. ¬ Buys (x, y) ∨ ¬ Pumpkin (y) ∨ Carves (x, y) ¬ Carves (stuart, x1) e(1). Buys (stuart, x1) e(2) Pumpkin (x1)

e(2) Pumpkin (x1) Eats (stuart, ~~[illegible]~~ x1) c. ¬ Eats (x, y) ∨ ¬ Pumpkin (y) ∨ Nutanatic (x)

Nutanatic (Stuart)

empty

conclusion proved

XINRUI ZHANG EECS 592 Project3 zxinrui

R&N 10.2

9. 1. action schema: Fly (P1, JFK, SFO)

initial state: At (P1, JFK) ∧ Plane (P1) ∧ Airport (JFK) ∧ Airport (SFO)

2. action schema Fly (P2, SFO, JFK)

initial state: At (P2, SFO) ∧ Plane (P2) ∧ Airport (JFK) ∧ Airport (SFO)

10. R&N 10.3

At (x, y) x at y

object1 (x) x is a object (box)

object2 (x) x is a object (banana)

Height (x, y) x have height y

~~grasp (x, y)~~ ~~x grasp y~~

Grasping (x, y) x is grasping y

Position (x) x is a position

Monkey (x) x is a monkey

M: monkey

B: box

BA: Banana

A, B, C: position

Low

High

a. initial state

Monkey (M) ∧ Position (A) ∧ Position (B) ∧ Position (C) ∧ Object1 (B) ∧ Object2 (BA)
∧ At (M, A) ∧ At (BA, B) ∧ At (B, C) ∧ Height (M, Low) ∧ Height (B, low)
∧ Height (BA, High) ~~∧ ¬ Grasping (M, BA)~~

b. six action schemas

1. Go (m, from, to)

Precond: Monkey (m) ∧ Position (from) ∧ Position (to) ∧ At (m, from) ∧ Height (M, Low)

Effect: ¬ At (m, from) ∧ At (m, to)

2. Push (m, object, from, to)

Precon: Monkey (m) ∧ Position (from) ∧ Prosition (to) ∧ Object1 (object) ∧ At (m, from) ∧ At (object, from) ∧ Height (m, Low) ∧ Height (object, Low)

Effect: ¬ At (m, from) ∧ ¬ At (object, from) ∧ At (m, to) ∧ At (object, to)

3. Climp Up (m, object, p)

Precon: Monkey (m) ∧ Object1 (object) ∧ ~~Positio~~ Position (P) ∧ At (m, P) ∧ At (object, p) ∧ Height (m, Low) ∧ Height (object, Low)

Effect: ¬ Height (m, Low) ∧ Height (m, high)

4. Climp Down (m, object, p)

Precon: Monkey (m) ∧ Object1 (object) ∧ Position (p) ∧ At (m, p) ∧ At (object, p) ∧ Height (m, high) ∧ Height (object, Low)

Effect: Height (m, Low) ∧ ¬ Heigh (m, high)

5. Grasp (m, object, p, height)

Precon: Monkey (m) ∧ object2 (object) ∧ Position (p) ∧ At (m, p) ∧ At (object, p) ∧ Height (m, h) ∧ Height (object, h)

Effect: Grasping (m, object) ∧ ¬ At (object, p) ∧ Height (object, h)

6. Ungrasp (m, object, p, h)

Precon: Monkey (m) ∧ object2 (object) ∧ position (p) ∧ At (m, p) ∧ Height (m, h) ∧ ~~gra~~ Grasping (m, object)

Effect: ¬ Grasping (m, object) ∧ At (object, p) ∧ Height (object, h)

10. c.

Goal: $Grasp(M, BA) \wedge At(B, p)$

Initial state:

$Monkey(M) \wedge Position(A) \wedge Position(B) \wedge Position(C) \wedge object1(B) \wedge Object2(BA) \wedge At(M, A) \wedge At(BA, B) \wedge At(B, p) \wedge Height(M, Low) \wedge Height(B, Low) \wedge Height(BA, High) \wedge ((p = A) \vee (p = B) \vee (p = C))$

This goal can not be solvet by a clasical planning system.

d. add a Heavy(x): x is heavy

Push heavy (m, object, from, to) ~~weight~~

Precon: $Monkey(M) \wedge Position(from) \wedge Position(to) \wedge Object1(object) \wedge At(m, from) \wedge At(object, from) \wedge Height(m, Low) \wedge Height(object, Low) \wedge Heavy(object)$

Effect: nothing changed

Push light (m, object, from, to)

Precon: $Monkey(m) \wedge Position(from) \wedge Position(to) \wedge Object1(object) \wedge At(m, from) \wedge At(object, from) \wedge Height(m, Low) \wedge Height(object, Low) \wedge \neg Heavy(object)$

Effect: $\neg At(m, from) \wedge \neg At(object, from) \wedge At(m, to) \wedge At(object, to)$