Họ và tên: Phan Trí Tài

MSSV: 20127318

HOMEWORK 6

(a) A B :

 (A => B) ∧ (B => A)

 (¬A ∨ B) ∧ (¬B ∨ A)

* 1. (A ∧ B)  (A ∨ B)

 ( ((A ∧ B) => (A ∨ B)) ∧ ((A ∨ B) => (A ∧ B)) )

 ( (¬(A ∧ B)) ∨ (A ∨ B)) ∧ (¬(A ∨ B) ∨ (A ∧ B)) )

 ( (¬A ∨ ¬B ∨ A ∨ B) ∧ ((¬A ∧ ¬B) ∨ (A ∧ B)) )

 ((¬A ∧ ¬B) ∨ (A ∧ B))

 ((¬A ∧ ¬B) ∨ A) ∧ ((¬A ∧ ¬B) ∨ B)

 (A ∨ ¬B) ∧ (¬A ∨ B)

* 1. A ∧ (A ⇒ B) ⇒ B

 ¬(A ∧ (¬A ∨ B)) ∨ B

 (¬A ∨ ¬ (¬A ∨ B)) ∨ B

 (¬A ∨ (A ∧ ¬B)) ∨ B

 (¬A ∨ ¬B) ∨ B

¬A ∨ 1

1

KB = {A ∧ B ⇒ C, D ∧ E ⇒ B, F ⇒ A, E ⇒ A, E, D}

Prove that the sentence C is entailed by KB by using

1. Forward chaining

R1: A ∧ B ⇒ C

R2: D ∧ E ⇒ B

R3: F ⇒ A R4: E ⇒ A

F1: E F2: D

• Execute:

o F3: A(F1,R4)

o F4: B(F1,F2,R2) o F5: C(R1,F3,F4)

* + We infer C
  + KB := C

1. Backward chaining

We have current goal is C.

Using the Backward Chaining rule, we have a new goals, A and B.

Using the Backward chaining rule for A as it conclusion. We find R3. R3 is such a rule, which only fire if it is satisfied D and E, so we have the new current goals is F include of A.

But we don’t find any rule which F as this solution so this goal failed.

Using the Backward chaining rule for A as it conclusion. We find R4. R4 is such a rule, which only fire if it is satisfied E, so we can reach this goal success.

Using the Backward chaining rule for B as it conclusion. We find R2. R2 is such a rule, which only fire if it is satisfied D and E, so we can reach this goal success.

Now, we have A and B so we will get goal is C.

* We infer C
* KB |= C

3:

Formalization:

+ Accomplice: A

+ Key: K

+ Car: C

KB = {(A => C) ∧ [(¬A ∧ ¬K) ∨ (A ∧ K)] ∧ K}

(\*): A => C ≡ ¬A ∨ C

(\*): [(¬A ∧ ¬K) ∨ (A ∧ K)] ≡ [(¬A ∧ ¬K) ∨ A] ∧ [(¬A ∧ ¬K) ∨ K]

# ≡ (¬K ∨ A) ∧ (¬A ∨ K)

|  |  |
| --- | --- |
| # | Clause |
|  | ¬A ∨ C |
| (¬K ∨ A) |
| (¬A ∨ K) |
| K |

A1: ¬A ∨ C

A2: (¬K ∨ A)

A3: (¬A ∨ K)

A4: K

A5: A (A4, A3)

A6: C (A5, A1)

=> KB |= C

=> The criminal came in a car

4.

1. ∀x∀y, (Cat(x) ∧ Mouse(y)) => Chase(x,y)
2. ∀x∃y, Cat(x) => (Mouse(y) ∧ Chase(x,y))
3. ∃x∀y, Cat(x) ∧ (Mouse(y) => Chase(x,y))
4. ∃x∃y, Cat(x) ∧ Mouse(y) ∧ Chase(x,y)
5. ∃y∀x, Mouse(y) ∧ (Cat(x) => Chase(x,y))
6. ∀y∃x, Mouse(y) => Cat(x) ∧ Chase(x,y)

5.

a)

* + Every child loves Santa :
    - , Child(x) => Loves(x,Santa)
  + Everyone who loves Santa loves any reindeer : o , Loves(x,Santa) => Reindeer(y) => Loves(x,y)
  + Rudolph is a reindeer, and Rudolph has a red nose: o Reindeer(Rudolph)  RedNose(Rudolph)
  + Anything which has a red nose is weird or is a clown: o , RedNose Weird Clown(x)
  + No reindeer is a clown :
    - ¬x, Reindeer Clown(x)
  + Scrooge does not love anything which is weird:
    - , Clown(x) => ¬Loves(Scrooge,x) b)
  + Every child loves Santa :
    - [Child(x) => Loves(x,Santa)] o  [¬ Child(x) Loves(x,Santa)] o CNF: [¬ Child(x)  Loves(x,Santa)]
  + Everyone who loves Santa loves any reindeer :
    - Reindeer(y) => Loves(x,y)
    - [Loves(x,Santa)  Reindeer(y) => Loves(x,y))] o [¬Loves(x,Santa) y ¬Reindeer(y)  Loves(x,y)]
    -  Loves(x,F(x))]
    -  o CNF: 
  + Rudolph is a reindeer, and Rudolph has a red nose: o Reindeer(Rudolph)  RedNose(Rudolph) o CNF : Reindeer(Rudolph)  RedNose(Rudolph)
  + Anything which has a red nose is weird or is a clown: o , RedNose(x) => Weird Clown
    - [¬RedNose Weird Clown o [¬RedNose Weird Clown o CNF: ¬RedNose Weird Clown
  + No reindeer is a clown :
    - ¬x, Reindeer Clown
    - ( Reindeer Clown o x[¬Reindeer ¬Clown o [¬Reindeer ¬Clown o CNF: [¬Reindeer ¬Clown
  + Scrooge does not love anything which is weird: o , Weird(x) => ¬Loves(Scrooge,x)
    - [¬Weird ¬Loves(Scrooge,x)] o [¬Weird ¬Loves(Scrooge,x)] o CNF: ¬ Weird  ¬Loves(Scrooge,x)

c)

|  |  |
| --- | --- |
| # | Clause |
| 1 | ¬ Child(x)  Loves(x,Santa) |
| 2 |  |
| 3 | Reindeer(Rudolph) |
| 4 | RedNose(Rudolph) |
| 5 | ¬RedNose  Weird  Clown |
| 6 | ¬Reindeer  ¬Clown |
| 7 | ¬ Weird  ¬Loves(Scrooge,x) |

A1: ¬ Child(x)  Loves(x,Santa)

A2: 

A3: Reindeer(Rudolph)

A4: RedNose(Rudolph)

A5: ¬RedNose(x) ∨ Weird(x) ∨ Clown(x)

A6: ¬Reindeer(x) ∨ ¬Clown(x)

A7: ¬ Weird (x) ∨ ¬Loves(Scrooge,x)

A8: ¬Clown(Rudolph)(A3,A6)

A9: Weird(Rudolph) (A8,A5,A4)

A10: ¬Loves(Scrooge, Rudolph) (A9,A7)

A11: ¬Loves(Scrooge,Santa) (A10,A2,A3)

A12: ¬ Child(Scrooge)(A11,A1)

 Scrooge is not a child