### Course: CS420 - Artificial Intelligence

REVIEW EXERCISE 04

**Question 1.** Given a knowledge base KB as follows, {**P R**, **¬S P**, **¬S**, **R Q**}. Consider the pseudo-code function PL-RESOLUTION given in the lecture to check whether **KB entails Q**.

Present your work to the table below, in which the first column contains KB ∧ ¬α in CNF, and every of the next columns includes new sentences added to KB after each loop. Note that

* Duplicated sentences are omitted from the table
* *Circle the unit clauses that lead to the contradiction and hence the function ends successfully, if possible*
* Process the clauses in order, that is first pair clause 1 with clause 2, 3, 4… then pair clause 2 with clause 3, 4,… and so on.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CNF sentences | Loop 1 | Loop 2 | Loop 3 | Loop 4 |
| 1. ¬P ∨ R | 1. R ∨ S | 1. R |  |  |
| 1. P ∨ S | 1. ¬P ∨ Q | 1. ¬P |  |  |
| 1. ¬S | 1. P | 1. S ∨ Q |  |  |
| 1. Q ∨ ¬R | 1. ¬R | 1. S |  |  |
| 1. ¬Q |  | 1. Q |  |  |

**Circle the correct option**, IS or IS NOT.

Following the result of resolution, the sentence Q **IS / IS NOT** entailed by KB.

**Question 2.** Repeat Question 1. but this time you check whether **KB entails ¬Q**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CNF sentences | Loop 1 | Loop 2 | Loop 3 | Loop 4 |
| 1. ¬P ∨ R | 1. R ∨ S | 1. R |  |  |
| 1. P ∨ S | 1. ¬P ∨ Q | 1. S ∨ Q |  |  |
| 1. ¬S | 1. P | 1. Q |  |  |
| 1. Q ∨ ¬R |  |  |  |  |
| 1. Q |  |  |  |  |

There is no new clause generated after Loop 2. There is no contradiction, either. Thus, **KB does not entails** **¬Q.**

**Question 3.** Are the above problems solved by using Forward chaining or Backward chaining? Give your reason.

NO, there exists a sentence that is not a definite clause, i.e ¬S ⇒ P

**Question 4.** Consider the following text. “*Heather attended the meeting* or Heather was not invited. If *the boss wanted Heather at the meeting*, then *she was invited*. Heather did not attend the meeting. If the boss did not want Heather there, and the boss did not invite her there, then *she is going to be fired*.”

Use resolution to prove that **Heather is going to be fired**.*Hint: clauses in italic are good candidates for propositions.*

Let each of following propositions denote the facts represented in the corresponding clause.

* Proposition A represents for “Heather attended the meeting.”
* Proposition I represents for “Heather was invited.”
* Proposition W represents for “The boss wanted Heather at the meeting.”
* Proposition F represents for “Heather is going to be fired.”

Then the propositional KB in CNF will be

1. A ∨ ¬I
2. ¬W ∨ I
3. ¬A
4. W ∨ I ∨ F

Apply resolution to KB ∧ ¬α

1. **¬F** Negation of conclusion
2. W ∨ I from sentences 4 and 5
3. I from sentences 2 and 6
4. A from sentences 1 and 7
5. • from sentences 3 and 8

Conclusion: Therefore, Heather is going to be fired.

**Question 5.** Consider the following knowledge base of definite clauses.

1. C ∧ D → Y
2. R ∧ Z → C
3. ¬B ∨ D
4. ¬D ∨ ¬R ∨ Z
5. B
6. R → D
7. D → R

Prove **Y** using backward chaining

**Y** requires two sub-goals, C and D (from 1)

D requires the sub-goal B (from 3). B is given (from 5). Thus **D is satisfied.**

Crequires two sub-goals, R and Z (from 2). R requires D (from 6), which is proved. Z requires two sub-goals, D and R, which are both proved. Thus, **C is satisfied**.

Finally, Y can be proved because all required propositions can be obtained from KB.

Prove **Y** using forward chaining (only trigger a rule once for simplicity).

8. Dis obtained from 3 and 5.

9. R is obtained from 7 and 8.

10. Z is obtained from 4 and 8-9 (rewrite 4 in implication form)

10\*. D is obtained **again** from 6 and 9.

11. C is obtained from 2 and 9-10

12. Y is obtained from 1 and 8-11.

Finally, Y can be proved because it is generated from KB.

**Question 6.** Convert the following English sentences into FOL sentences, using only the predicates given inside the square brackets.

1. All green apples are sour. [Green1, Apple1, Sour1]

∀x Green(x) ∧ Apple(x) → Sour(x)

1. All babies love some green apples. [Baby1, Loves2, Apple1, Green1]

∀x Baby(x) → [∃y Green(y) ∧ Apple(y) ∧ Loves(x, y)]

1. Some babies do not love any sour apple. [Baby1, Loves2, Apple1, Sour1]

∃x Baby(x) ∧ [∀y Sour(y) ∧ Apple(y) → ¬Loves(x, y)]

1. Mary eats only one apple. [Apple1, Eat2]

∃x Apple(x) ∧ Eat(Mary, x) ) ∧ [∀y Apple(y) ∧ ¬(x = y) → ¬Eat(Mary, y)]

**Question 7.** Find (if it were possible) the MGU for each of the following pairs of FOL statements. Note that uppercase letters represent constants, while lowercase ones denote varibles, predicate/function names.

1. P( g(h(x)) , f(g(h(B))) , f(x) ) and P( y , f(y), z)
2. P( g(h(x)) , f(h(y)) , y ) and P( g(z) , f(z) , h(A) )
3. P( x , h(B) , h(x) ) and P( f(g(y)) , y , h(f(g(h(A)))) )
4. P( x , g(x) , z ) and P( f(y) , g(f(B)) , h(y) )
5. P( f(g(x)) , g(B) , h(x) ) and P( f(y) , y , h(C) )
6. P( x , h(x) , h(y) ) and P( f(g(z)) , h(f(g(B))) , h(z) )
7. θ = {y / g(h(B)) , x / B , z / f(B) }
8. θ = { x / h(A) , y / h(A) , z / h(h(A)) }
9. No MGU
10. θ ={ x / f(B) , y / B , z / h(B)}
11. No MGU
12. θ ={ x / f(g(B)) , z / B , y / B }

**Question 8.** Consider the following text. *“Anyone passing his history exam and winning the lottery is happy. But anyone who studies or is lucky can pass all his exams. John did not study but John is lucky. Anyone who is lucky wins the lottery.”*

1. For each of the axiom above, write the FOL sentence that best expresses its intended meaning, using only the following predicates

PASS(x, y): “x passes the y exam” HAPPY(x): “x is happy” LUCKY(x): “x is lucky”

STUDY(x): “x studies” WINLOT(x): “x wins the lottery”

1. ∀x PASS(x, history exams) ∧ WINLOT(x) ⇒ HAPPY(x)
2. ∀x STUDY(x) ∨ LUCKY(x) ⇒ ∀y PASS(x, y)
3. ¬STUDY(John) ∧ LUCKY(John)
4. ∀x LUCKY(x) ⇒ WINLOT(x)
5. Convert the above FOL clauses to clausal form
6. ¬PASS(x, history exams) ∨ ¬WINLOT(x) ∨ HAPPY(x)
7. ¬STUDY(x) ∨ PASS(x,y)
8. ¬LUCKY(x) ∨ PASS(x,y)
9. ¬STUDY(John)
10. LUCKY(John)
11. ¬ LUCKY(x) ∨ WINLOT(x)
12. Use resolution to answer the question “Is John happy?”
13. ¬HAPPY(John) Negation of conclusion
14. ¬PASS(John, history exams) ∨ ¬WINLOT(John) from 1 and 7 θ = { x/John}
15. WINLOT(John) from 5 and 6 θ = { x/John}
16. ¬PASS(John, history exams) from 8 and 9 θ = { x/John}
17. PASS(John, y) from 3 and 5 θ = { x/John}
18. • from 10 and 11 θ = { x/John, y/history}

Conclusion: Therefore, John is happy.

**Question 9.** Consider the following KB.

|  |
| --- |
| 1. Buffalo(x) ∧ Pig(y) → Faster(x,y) 4. Buffalo(Bob)  2. Pig(y) ∧ Slug(z) → Faster (y,z) 5. Pig(Pat)  3. Faster(x,y) ∧ Faster (y, z) → Faster(x, z) 6. Slug(Steve) |

Use forward chaining in first-order logic to prove **Faster(Bob, Steve)**. If several rules apply, use the one with the smallest number. Do not forget to indicate the unification at every step.

7. Faster(Bob, Pat) from 1 and 4-5 θ = {x / Bob, y / Pat}

8. Faster(Pat, Steve) from 2 and 5-6 θ = {x / Bob, y / Pat, z / Steve}

9. Faster(Bob, Steve) from 3 and 7-8 θ = {x / Bob, y / Pat}

Thus, KB entails Faster(Bob, Steve) using forward chaining.