Here's an explanation for each answer:

a. ∀x (Child(x) → Loves(x, Santa))

This statement represents the sentence "Every child loves Santa." It uses the universal quantifier ∀x to say that for every x (which represents an individual), if x is a child, then x loves Santa.

b. ∀x (Loves(x, Santa) → (∀y (Reindeer(y) → Loves(x, y))))

This statement represents the sentence "Everyone who loves Santa loves any reindeer." It uses the universal quantifier ∀x to say that for every x, if x loves Santa, then for every y (representing a reindeer), x loves y.

c. Reindeer(Rudolph) ∧ HasRedNose(Rudolph)

This statement represents the sentence "Rudolph is a reindeer, and Rudolph has a red nose." It asserts that Rudolph is a reindeer (Reindeer(Rudolph)) and also has a red nose (HasRedNose(Rudolph)).

d. ∀x (HasRedNose(x) → (Weird(x) ∨ Clown(x)))

This statement represents the sentence "Anything which has a red nose is weird or is a clown." It uses the universal quantifier ∀x to say that for every x, if x has a red nose, then x is either weird or a clown.

e. ∀x (Reindeer(x) → ¬Clown(x))

This statement represents the sentence "No reindeer is a clown." It uses the universal quantifier ∀x to say that for every x, if x is a reindeer, then x is not a clown.

f. ∀x (Weird(x) → ¬Loves(Scrooge, x))

This statement represents the sentence "Scrooge does not love anything which is weird." It uses the universal quantifier ∀x to say that for every x, if x is weird, then Scrooge does not love x.

g. ¬Child(Scrooge)

This statement represents the conclusion "Scrooge is not a child." It simply states that Scrooge is not a child (¬Child(Scrooge)).

Đây là một lời giải thích cho mỗi câu trả lời:

Một. ∀x (Con(x) → Yêu(x, ông già Noel))

Câu nói này tiêu biểu cho câu "Mọi đứa trẻ đều yêu ông già Noel." Nó sử dụng lượng từ chung ∀x để nói rằng với mọi x (đại diện cho một cá nhân), nếu x là một đứa trẻ, thì x yêu ông già Noel.

b. ∀x (Yêu(x, ông già Noel) → (∀y (Tuần lộc(y) → Yêu(x, y))))

Câu nói này tiêu biểu cho câu “Ai yêu ông già Noel thì yêu tuần lộc nào”. Nó sử dụng lượng từ chung ∀x để nói rằng với mọi x, nếu x yêu ông già Noel, thì với mọi y (đại diện cho một con tuần lộc), x yêu y.

c. Tuần Lộc(Rudolph) ∧ Mũi Đỏ(Rudolph)

Tuyên bố này đại diện cho câu "Rudolph là một con tuần lộc, và Rudolph có mũi đỏ." Nó khẳng định rằng Rudolph là một con tuần lộc (Reindeer(Rudolph)) và cũng có chiếc mũi đỏ (HasRedNose(Rudolph)).

d. ∀x (HasRedNose(x) → (Kỳ lạ(x) ∨ Chú hề(x)))

Câu nói này tiêu biểu cho câu "Con gì có mũi đỏ là kỳ dị hoặc là thằng hề". Nó sử dụng lượng từ chung ∀x để nói rằng với mọi x, nếu x có mũi đỏ, thì x hoặc là kỳ dị hoặc là một chú hề.

đ. ∀x (Tuần lộc(x) → ¬Clown(x))

Tuyên bố này đại diện cho câu "Không có tuần lộc là một chú hề." Nó sử dụng lượng từ chung ∀x để nói rằng với mọi x, nếu x là tuần lộc, thì x không phải là chú hề.

f. ∀x (Weird(x) → ¬Loves(Scrooge, x))

Tuyên bố này đại diện cho câu "Scrooge không yêu bất cứ điều gì kỳ lạ." Nó sử dụng bộ định lượng phổ quát ∀x để nói rằng với mọi x, nếu x là lạ, thì Scrooge không yêu x.

g. ¬Con (Scrooge)

Tuyên bố này đại diện cho kết luận "Scrooge không phải là một đứa trẻ." Nó chỉ đơn giản nói rằng Scrooge không phải là một đứa trẻ (¬Child(Scrooge)).

To check if "R" is derived from the given knowledge base in CNF form, we need to perform resolution refutation. Resolution refutation is a proof technique in propositional logic that aims to derive a contradiction (such as "R ∧ ¬R") by applying resolution steps.

Let's set up the resolution refutation proof:

1. Negate the goal: ¬R

2. Convert the knowledge base to a set of clauses:

Clause 1: (P ∨ Q) ∨ Q

Clause 2: True

Clause 3: (R ∧ ¬S) ∨ (S ∧ ¬Q)

3. Add the negated goal as an additional clause: ¬R

Now, let's proceed with the resolution refutation:

1. Apply resolution between Clause 1 and ¬R:

(P ∨ Q) ∨ Q

¬R

----------------

(P ∨ Q)

2. Apply resolution between Clause 3 and (P ∨ Q):

(R ∧ ¬S) ∨ (S ∧ ¬Q)

(P ∨ Q)

----------------

(R ∧ ¬S) ∨ (S ∧ ¬Q)

3. Apply resolution between Clause 2 and (R ∧ ¬S) ∨ (S ∧ ¬Q):

True

(R ∧ ¬S) ∨ (S ∧ ¬Q)

----------------

True

The resolution refutation ends with the derived clause "True," indicating that the original goal ¬R is not contradicted. Therefore, "R" is not derived from the given knowledge base in CNF form.

Để kiểm tra xem "R" có được lấy từ cơ sở tri thức đã cho ở dạng CNF hay không, chúng ta cần thực hiện bác bỏ độ phân giải. Bác bỏ giải quyết là một kỹ thuật chứng minh trong logic mệnh đề nhằm mục đích rút ra một mâu thuẫn (chẳng hạn như "R ∧ ¬R") bằng cách áp dụng các bước giải quyết.

Hãy thiết lập bằng chứng bác bỏ giải pháp:

1. Phủ định mục tiêu: ¬R

2. Chuyển đổi cơ sở tri thức thành một tập hợp các mệnh đề:

Mệnh đề 1: (P ∨ Q) ∨ Q

Mệnh đề 2: Đúng

Mệnh đề 3: (R ∧ ¬S) ∨ (S ∧ ¬Q)

3. Thêm mục tiêu bị phủ định dưới dạng mệnh đề bổ sung: ¬R

Bây giờ, hãy tiến hành bác bỏ giải pháp:

1. Áp dụng cách giải quyết giữa Khoản 1 và ¬R:

(P ∨ Q) ∨ Q

¬R

----------------

(P ∨ Q)

2. Áp dụng cách giải quyết giữa Khoản 3 và (P ∨ Q):

(R ∧ ¬S) ∨ (S ∧ ¬Q)

(P ∨ Q)

----------------

(R ∧ ¬S) ∨ (S ∧ ¬Q)

3. Áp dụng cách giải giữa Mệnh đề 2 và (R ∧ ¬S) ∨ (S ∧ ¬Q):

ĐÚNG VẬY

(R ∧ ¬S) ∨ (S ∧ ¬Q)

----------------

ĐÚNG VẬY

Việc bác bỏ giải pháp kết thúc bằng mệnh đề dẫn xuất "True", chỉ ra rằng mục tiêu ban đầu ¬R không bị mâu thuẫn. Do đó, "R" không được lấy từ cơ sở tri thức đã cho ở dạng CNF.

4. Cho 1 KB như sau: KB = { (P → Q) → Q (P→P) → R, (R→S) → ¬ (S→Q)}

biến đổi tập cơ sở tri thức trên về dạng hội chuẩn (CNF)

To transform the given knowledge base into Conjunctive Normal Form (CNF), we need to follow a few steps:

1. Eliminate the implication using the logical equivalence: p → q ≡ ¬p ∨ q.

2. Apply De Morgan's laws: ¬(p ∧ q) ≡ ¬p ∨ ¬q and ¬(p ∨ q) ≡ ¬p ∧ ¬q.

3. Distribute ∨ over ∧ using the distributive law: p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r).

Let's apply these steps to the given knowledge base:

1. (P → Q) → Q

Applying the implication elimination:

¬(P → Q) ∨ Q

2. (P → P) → R

Applying the implication elimination:

¬(P → P) ∨ R

Since P → P is a tautology (always true), we can simplify it to:

True ∨ R

This simplifies to:

True

3. (R → S) → ¬(S → Q)

Applying the implication elimination:

¬(R → S) ∨ ¬(S → Q)

Now, let's combine these expressions into CNF:

1. ¬(P → Q) ∨ Q --> (¬¬P ∨ Q) ∨ Q --> (P ∨ Q) ∨ Q

This is already in CNF.

2. True --> True ∧ True

This is already in CNF.

3. ¬(R → S) ∨ ¬(S → Q) --> (¬(¬R ∨ S)) ∨ ¬(¬S ∨ Q) --> (R ∧ ¬S) ∨ (S ∧ ¬Q)

This is already in CNF.

Therefore, the transformed CNF form of the given knowledge base is:

CNF = { (P ∨ Q) ∨ Q, True, (R ∧ ¬S) ∨ (S ∧ ¬Q) }

Để chuyển đổi cơ sở tri thức đã cho thành Dạng chuẩn liên kết (CNF), chúng ta cần thực hiện theo một số bước:

1. Loại bỏ hàm ý bằng cách sử dụng tương đương logic: p → q ≡ ¬p ∨ q.

2. Áp dụng các định luật De Morgan: ¬(p ∧ q) ≡ ¬p ∨ ¬q và ¬(p ∨ q) ≡ ¬p ∧ ¬q.

3. Phân phối ∨ trên ∧ sử dụng luật phân phối: p ∨ (q ∧ r) ≡ (p ∨ q) ∧ (p ∨ r).

Hãy áp dụng các bước này cho cơ sở tri thức đã cho:

1. (P → Q) → Q

Áp dụng phép loại bỏ hàm ý:

¬(P → Q) ∨ Q

2. (P → P) → R

Áp dụng phép loại bỏ hàm ý:

¬(P → P) ∨ R

Vì P → P là một phép lặp (luôn luôn đúng), chúng ta có thể đơn giản hóa nó thành:

Đúng ∨ R

Điều này đơn giản hóa thành:

ĐÚNG VẬY

3. (R → S) → ¬(S → Q)

Áp dụng phép loại bỏ hàm ý:

¬(R → S) ∨ ¬(S → Q)

Bây giờ, hãy kết hợp các biểu thức này thành CNF:

1. ¬(P → Q) ∨ Q --> (¬¬P ∨ Q) ∨ Q --> (P ∨ Q) ∨ Q

Điều này đã có trong CNF.

2. Đúng -> Đúng ∧ Đúng

Điều này đã có trong CNF.

3. ¬(R → S) ∨ ¬(S → Q) --> (¬(¬R ∨ S)) ∨ ¬(¬S ∨ Q) --> (R ∧ ¬S) ∨ (S ∧ ¬Q )

Điều này đã có trong CNF.

Do đó, dạng CNF được chuyển đổi của cơ sở tri thức đã cho là:

CNF = { (P ∨ Q) ∨ Q, Đúng, (R ∧ ¬S) ∨ (S ∧ ¬Q) }

A picture containing diagram, line, circle, design

Description automatically generated

1a. Hãy liệt kê:   
- danh sách các node theo thứ tự được mở rộng,  
- đường đi tìm được

A picture containing diagram, circle, screenshot, line

Description automatically generated

A picture containing circle, diagram

Description automatically generated

BFS: S->B->G

DFS: S->A->C->G , DEEP =4,COST=9

UCS: S->(2)A->(3)C->(2)G

(A và B là con của S, => A không thể là cha của B

GREEDY: S(10)->B(5)->G ,COST=9

A\*: S(10)->(5+4)B(5)->(5+0)G(0) ,COST =9

1b.Heuristic

A picture containing diagram, line, plot

Description automatically generated