Ali Awari – aka87

Homework #2

1.

A ∧ (B ∨ C) ⇔ (A ∧ B) ∨ (A ∧ C)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | B ∨ C | A ∧ (B ∨ C) | A ∧ B | A ∧ C | (A ∧ B) ∨ (A ∧ C) |
| T  T  T  T  F  F  F  F | T  T  F  F  T  T  F  F | T  F  T  F  T  F  T  F | T  T  T  F  T  T  T  F | **T**  **T**  **T**  **F**  **F**  **F**  **F**  **F** | T  T  F  F  F  F  F  F | T  F  T  F  F  F  F  F | **T**  **T**  **T**  **F**  **F F F F** |

A ∨ (B ∧ C) ⇔ (A ∨ B) ∧ (A ∨ C)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | B ∧ C | A ∨ (B ∧ C) | A ∨ B | A ∨ C | (A ∨ B) ∧ (A ∨ C) |
| T  T  T  T  F  F  F  F | T  T  F  F  T  T  F  F | T  F  T  F  T  F  T  F | T  F  F  F  T  F  F  F | **T**  **T**  **T**  **T**  **T**  **F**  **F**  **F** | T  T  T  T  T  T  F  F | T  T  T  T  T  F  T  F | **T**  **T**  **T**  **T**  **T F F F** |

2.

¬(A ∨ B) ⇔ (¬A) ∧ (¬B)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | ¬A | ¬B | A ∨ B | ¬(A ∨ B) | (¬A) ∧ (¬B) |
| T  T  F  F | T  F  T  F | F  F  T  T | F  T  F  T | T  T  T  F | **F**  **F**  **F**  **T** | **F**  **F**  **F**  **T** |

¬(A ∧ B) ⇔ (¬A) ∨ (¬B)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | ¬A | ¬B | A ∧ B | ¬(A ∨ B) | (¬A) ∧ (¬B) |
| T  T  F  F | T  F  T  F | F  F  T  T | F  T  F  T | T  F  F  F | **F**  **T**  **T**  **T** | **F**  **T**  **T**  **T** |

3.

What does the person’s rule tell you about the hidden side of each card?

Assuming their rule holds:

* That the first card ‘A’, has an even number on the hidden side.
* The second card ‘7’, has a non-vowel letter on the hidden side.
* The third card ‘10’, has a vowel on the hidden side.
* The fourth card ‘B’, has an odd number on the hidden side.

If you want to check or verify the person’s rule by flipping over a card or cards, which cards should you flip, and what should you conclude from the reverse side?

4.

(A ∨ B ∨ ¬C) ∧ (¬A ∨ B ∨ C) ∧ (A ∨ ¬B ∨ ¬C)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | ~A | ~B | ~C | A∨B∨~C | ~A∨B∨C | A∨~B∨~C | Original Statement |
| T  T  T  T  F  F  F  F | T  T  F  F  T  T  F  F | T  F  T  F  T  F  T  F | F F F F T T T T | F F T T F F T T | F T F T F T F T | T  T  T  T  T  T  F  T | T  T  T  F  T  T  T  T | T  T  T  T  F  T  T  T | **T**  **T**  **T**  **F**  **F**  **T**  **F**  **T** |

5.

(A xor B) ⇔ ¬(A ⇔ B)

|  |  |  |
| --- | --- | --- |
| A xor B | A ⇔ B | ¬(A ⇔ B) |
| F  T  T  F | T  F  F  T | **F**  **T**  **T**  **F** |

6.

* All men must die. I am not a man. Must I die?

***NO, “*I” is not in the domain of man.**

* All men must die. I must die. Am I a man?

***NOT necessarily, not stated “I” is a man.***

* All men must die. I must not die. Must I die?

***You are NOT a man, thus NO.***

* All men must die. Must Socrates die?

***We CANNOT conclude he must die, without knowing if whether Socrates is a man or not.***

* All men must die. John Tucker must die. Is John Tucker Socrates?

***NO.***

***\*\* Questions, 7,8,9 were skipped in the prompt.***

10.

Argue that modus ponens is actually just a special case of the resolution inference rule.

Modus Ponens is A => B, A : B.

Resolution is [A or B, NOT(B) or C], therefore A or C.

However, Resolution is also, [A or B, B => C] , therefore A or C.

Because the Resolution Rule uses an implies to condense the statement with two variables it can be seen that Modus Ponens is a special case of Resolution only using one variable instead of two.

11.

It can be argued that the resolution inference rule is in fact complete, in the sense that it is the only rule needed to infer everything that can be inferred. Why might other inference rules (not just modus ponens) be useful to know?

Modus Ponens is useful when you need to prove simple arguments, Resolution Inference is useful with three variables.

12.

1. B ∨ C, 2. ¬B ∨ ¬C, 3. A ∨ B, 4. A ∨ C, 5. ¬A ∨ ¬B ∨ ¬C

6. ¬A

Expression 3: [A ∨ B], given Expression 6, B.

Expression 4: [A ∨ C], given Expression 6, C.

Expression 1: [B ∨ C], given Expressions 3, 4, TRUE.

Expression 5: [¬A ∨ ¬B ∨ ¬C], given Expression 6, statement works.

**Expression 2: [¬B ∨ ¬C], Contradiction.**

13.

P = ∀x : A(x) ⇒ B(x), where A(x) is the statement that x ***is made of cheese***, and B(x) is the statement that x ***orbits the Earth***. Let the domain of interest be ***the set of satellites in orbit around the earth***. Is P true?

P is true if you find an x that suffices A(x) => B(x).

14.

∀n : n not divisible by 3 ⇒ n2 gives a remainder of 1 when divided by 3

– What would the initial step / formulation of a direct proof look like?

Euclid's algorithm, n=3a+b, b is either 0,1,2

Because n is now n2 you remove 2 as a possibility for a remainder.

And since, remainder 0 is not an option (divisible by 3), 1 is left.

– What would the initial step / formulation of a proof by contrapositive look like?

If n2 doesn’t give a remainder of 1 when divided by 3 then n is divisible by 3.

– What would the initial step / formulation of a proof by contradiction look like?

n ***is*** divisible by 3 => n2 gives a remainder of 1 when divided by 3

– What would the inductive step of a proof by induction look like?

Because n is an integer, n = 3k, n = 3k+1, or n = 3k+2

If n2 = (3k)2 = 3 \* 3k2, 3k2 is an integer, multiple of 3, remainder 0.

If n2 = (3k + 1)2 = 9x2 + 6x + 1 = 3(3k2 + 2x) + 1, remainder 1.

If n2 = (3k + 2)2 = 9x2 + 12x + 4 = 3(3k2 + 4x + 1) + 1, remainder 1.

– Which of these approaches seems most promising? Why?

Induction, it allows one to prove the hypothesis mathematically.

An alternative approach might be to consider proof by cases. In particular, for any n, n gives a remainder of 0, 1, or 2 when divided by 3. By considering these three cases separately, prove the claim for a given n.

Case 1: When n = 3, remainder is 0.

Case 2: When n = 4, remainder is 1.

Case 3: When n = 5, remainder is 1, if n wasn’t squared remainder is 2.

15.

Prove that over the domain of non-negative integers, ∀n : n2 + n is even.

Can you construct a proof by minimal counter example? What is the best approach to take here?

Suppose there is a smallest positive integer that cannot be even, call it *m*.

1 ≤ n < *m*

1 = 12 + 1 => 2: *m* ≥ 2

*k* < *m*

*m* = *k* + 1, where 1 ≤ *k* < *m*

(*k* + 1)2 + (*k* + 1) = *k*2 + 2*k* + 1 + *k* + 1 = *k*2 + 3*k* + 2 = *k*(*k* + 3) + 2

(-2*k* - 2) + *k*2 + *k* < *k*2 + 3*k* + 2 + (-2*k* - 2)

*k*2 + -*k* - 2 < *k*2 + *k*

Also true for m + 1, contradiction

16.

Consider again the Fibonacci numbers, defined by the sequence f(0) = 1, f(1) = 1, and f(n+2) = f(n)+f(n+1) for n ≥ 0. In the sections above, we proved that f(n) ≤ 2 n for all n. What is the smallest positive number α you can find such that the proof can be adapted to show

∀n ≥ 0 : f(n) ≤ αn ?

a = 1.1

17.

Show that for n ≥ 3 over the set of positive integers, we have:

1/1 + 1/2 + 1/3 + . . . + 1/n ≤ 2 ln(n).

You may find the inequality x ≤ −2 ln(1 − x) for 0 < x < 1 useful here.

Base Case: 1/1 + 1/2 + 1/3 + ... + 1/n <= 2 ln(n)

Base Case: 1/1 + 1/2 + 1/3 + ... + 1/n+1 <= 2 ln(n+1)

1/1 + 1/2 + 1/3 + … + 1/n ≤ 2 ln(n)

P(n) = 1/1 + 1/2 + 1/3 + … + 1/n

[1/1 + 1/2 + 1/3 + … + 1/(n+1)] ≤ 2 ln(n+1)

P(n) + 1/(n+1) ≤ 2 ln(n) + 1/(n+1)

P(n+1) ≤ 2 ln(n) + 1/(n+1)

Base Case #2 means that:

P(n+1) ≤ 2 ln(n) + 1/(n+1) ≤ 2 ln(n+1)

2 ln(n) + 1/(n+1) ≤ 2 ln(n + 1)

18.

Suppose that cats are sold in sets of 3 or sets of 5. Show that for any n ≥ 8, someone can buy exactly n cats.

**Base Case**:

*n* = 8, 3 + 5

*n* = 9, 3 + 3 + 3

*n* = 10, 5 + 5

*n* = 11, 5 + 3 + 3

**Inductive Step**:

Since *k* + 1 ≥ 8, (*k* + 1) – 3 ≥ 5

(*k* + 1) – 3) = 3*m* + 5*n*

(*k* + 1) – 3) = 3(*m* + 1) + 5*n*