1. Translate the following sentences into predicate logic. State what your predicate names mean.

\[ A(x, y) : x \text{ gets assigned task } y. \]
\[ F(x, y) : x \text{ is a friend of } y. \]
\[ H(x) : x \text{ is happy.} \]
\[ S(x, y) : x \text{ solves task } y. \]
\[ T(x) : x \text{ is a troubleshooter.} \]

Implicitly, a task is only assigned to one person:

A. \[ \forall y : \forall x : \forall z : (A(x, y) \land A(z, y)) \rightarrow x = z \]

1. Any happy person has all tasks that are assigned to them solved by someone.

\[ \forall x : H(x) \rightarrow \forall y : A(x, y) \rightarrow \exists z : S(z, y) \text{ or} \]
\[ \forall x : \forall y : (H(x) \land A(x, y)) \rightarrow S(z, y) \]

2. A task assigned to someone can only be solved by themselves or by a troubleshooter.

\[ \forall y : A(x, y) \rightarrow \forall z : S(z, y) \rightarrow ((z = x) \lor T(z)) \text{, or} \]
\[ \forall y : \forall x : \exists z : (A(x, y) \land S(z, y)) \rightarrow ((z = x) \lor T(z)) \]

3. A troubleshooter solves precisely any tasks assigned to any friend of theirs.

\[ \forall x : T(x) \rightarrow \forall y : S(x, y) \leftrightarrow \exists z : A(z, y) \land F(z, x) \]

C. If someone does not solve all their assigned tasks and they are happy, they have a troubleshooter as friend.

\[ \forall z : (\exists y : A(x, y) \land \neg S(x, y) \land H(x)) \rightarrow \exists z : T(z) \land F(z, x) \]

1.2 Prove the conclusion C from the premises 1-3. Assume the friendship relation is antireflexive and symmetric. Indicate the use of proof rules as clearly as possible.

We start a direct proof by assuming the premise of the implication in C:

- (4) \[ \exists y : A(x, y) \land \neg S(x, y) \land H(x) \] (Assumption (x arbitrary))
- (5) \[ A(x, b) \land \neg S(x, b) \land H(x) \] (Existence (4) (b: new constant))
- (6) \[ H(x) \] (Separation (5))

We show that there is some a who solves the task b that x does not solve.

- (7) \[ H(x) \rightarrow \forall y : A(x, y) \rightarrow \exists z : S(z, y) \] (Specification (to x))
- (8) \[ \forall y : A(x, y) \rightarrow \exists z : S(z, y) \] (Modus Ponens (6, 7))
- (9) \[ A(x, b) \rightarrow \exists z : S(z, b) \] (Specification, y = b)
- (10) \[ A(x, b) \] (Separation (5))
- (11) \[ \exists z : S(z, b) \] (Modus Ponens (9, 10))
- (12) \[ S(a, b) \] (Existence (11) (a: new constant))

We prove by contradiction that a is not x, and then that a must be a troubleshooter.

- (13) \[ \neg S(x, b) \] (Separation (5))
- (14) \[ S(a, b) \land \neg S(x, b) \] (Conjunction (12, 13))
- (15) \[ x = a \] (Assume)
- (16) \[ S(a, b) \land \neg S(a, b) \] (Substitution (14, 15))
- (17) \[ \neg (x = a) \] (Contradiction (15, 16))
- (18) \[ A(x, b) \land S(a, b) \] (Conjunction (10, 12))
- (19) \[ A(x, b) \land S(a, b) \rightarrow (a = x) \lor T(a) \] (3 specification (2))
- (20) \[ (a = x) \lor T(a) \] (Modus Ponens (18, 19))
- (21) \[ T(a) \] (Zero Rule for \( \lor \)) (17, 20)

We show that x and a must be friends.

- (22) \[ T(a) \rightarrow \forall y : S(a, y) \leftrightarrow \exists z : A(z, y) \land F(z, a) \] (Specification (3))
- (23) \[ \forall y : S(a, y) \leftrightarrow \exists z : A(z, y) \land F(z, a) \] (Modus Ponens (21, 22))
- (24) \[ S(a, b) \leftrightarrow \exists z : A(z, b) \land F(z, a) \] (Specification (23))
- (25) \[ S(a, b) \rightarrow \exists z : A(z, b) \land F(z, a) \] (Equivalence, Separation (24))
- (26) \[ \exists z : A(z, b) \land F(z, a) \] (Modus Ponens (12, 25))
- (27) \[ A(c, b) \land F(c, a) \] (Existence (26))
- (28) \[ A(c, b) \] (Separation (27))
- (29) \[ c = x \] (Uniqueness of task assignment (A, 10, 28))
- (30) \[ F(x, a) \] (Separation (27), Substitution c = x)
- (31) \[ F(a, x) \] (Symmetry of F)
We end by introducing quantifiers and discharging the assumption (premise of C).

\[(32) \ T(a) \land F(a, x) \quad \text{Conjunction (21, 31)}\]

\[(33) \exists z : T(z) \land F(z, x) \quad \text{Existence (32)}\]

\[(34) \exists y : A(x, y) \land \neg S(x, y) \land H(x)) \implies \exists z : T(z) \land F(z, x) \quad \text{Direct Proof (4, 33)}\]

\[(35) \forall z : (\exists y : A(x, y) \land \neg S(x, y) \land H(x)) \implies \exists z : T(z) \land F(z, x) \quad \text{Generalization (34)}\]

2.1 Convert the following formula to conjunctive normal form. Show your steps.

\[((a \to b) \to \neg c) \land (\neg (a \oplus d) \land (b \to \neg (c \to a)) \land (\neg (b \lor d) \to \neg c) \land (d \to \neg (a \lor c))\]

\[\neg(a \to b) \lor \neg c \land (a \to d) \land (d \to a) \land (\neg b \lor (c \land \neg a)) \land (b \lor d \to \neg c) \land (d \to \neg (a \land \neg c))\]

\[\((a \land \neg b) \lor \neg c \land (\neg a \lor d) \land (a \lor \neg d) \land (\neg b \lor c) \land (\neg b \land \neg a) \land (b \lor d \land \neg c) \land (d \lor \neg a) \land (\neg d \lor \neg c)\]

2.2 Determine all truth assignments that satisfy the formula. Show your work.

We can simplify \((\neg b \lor \neg c) \land (\neg b \lor c) = \neg b \land (\neg c \lor c) = \neg b\).

We duplicate \((\neg d \lor \neg a)\) by idempotence and simplify again using distributivity and excluded middle:

\[\neg(a \lor d) \land (a \lor \neg d) = \neg a \land (a \lor \neg d) \land (a \lor \neg d) = \neg a\].

These three unit clauses give \(a = b = d = F\).

The remaining clauses are: \((a \lor \neg c) \land (\neg b \lor \neg a) \land (b \lor d \lor \neg c) \land (d \lor \neg a) \land \neg d \lor \neg c)\).

Substituting the found values, we get \(-c\), thus \(c = F\), and we have a unique satisfying assignment.

3. Consider the set \(P\) of all partitions of a finite set \(S\). Define a relation on \(P\) as follows: \(R(C_1, C_2)\) iff for any set \(X \in C_1\) there is a set \(Y \in C_2\) such that \(Y \subset X\) (proper subset). Show that \(R\) is a strict order (antireflexive, antisymmetric, and transitive).

Take an arbitrary partition \(C\) and let \(X_m \in C\) be a set of minimum cardinality (since the underlying set \(S\) is finite, this exists). Then \(\neg R(C, C)\) since \(Y \in C\) with \(Y \subset X_m\) with would imply \(|Y| < |X_m|\).

For antisymmetry, define \(m(C) = \min_{X \in C} |X|\). Again, this exists, \(S\) being finite. Then \(R(C_1, C_2) \implies m(C_2) < m(C_1)\) if we choose \(X \in C_1\) with \(|X| = m(C_1)\). Likewise, \(R(C_2, C_1) \implies m(C_2) < m(C_1)\). Thus, \(R(C_1, C_2) \land R(C_2, C_1)\) is a contradiction, and the condition for implication is vacuously true.

For transitivity, take arbitrary \(C_1, C_2, C_3\), and \(X \in C_1\). Then \(R(C_1, C_2)\) implies there is \(Y \subset C_2\) with \(Y \subset X\), and \(R(C_2, C_3)\) gives \(Z \subset C_3\) with \(Z \subset Y\). The transitivity of \(\subset\) gives \(Z \subset X\) gives \(R(C_1, C_3)\), and by generalization, \(R\) is transitive.

4.1 Let \(x, y, z, w\) be naturals. Assuming we know \(x\) is divisible by the three numbers \(y, z,\) and \(y + z + w\), prove or disprove that \(x\) is divisible by \(w\).

**False.** A counterexample is \(x = 5, y = 1, z = 1\) and \(w = 3\).

We see \(x\) is divisible by \(y, z\) and \(y + z + w\) but not by \(w\).

4.2 Let \(x, y, z, w\) be naturals. Assuming we know \(x\) divides the three numbers \(y, z\), and \(y + z + w\), prove or disprove that \(x\) divides \(w\).

**True.** We know for some integers \(k, m, n\) we have: \(kx = y, mx = z, nx = y + z + w\).

Then \(w = nx - y - z = nx - kx - mx = x(n - k - m)\), which shows \(w\) is divisible by \(x\).

5. Define the binary relation \(R\) on \(\mathbb{N}\) by \(R(x, y)\) iff \(\text{lcm}(x, y) = x \cdot y\). Argue whether this relation is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

If \(\text{lcm}(x, y) = xy\), then \(\gcd(x, y) = 1\), since \(\text{lcm}(x, y) \cdot \gcd(x, y) = xy\).

**Not Reflexive:** \(\gcd(2, 2) = 2\) so \((2, 2) \notin R\)

**Not Antireflexive:** \(\gcd(1, 1) = 1\) so \((1, 1) \in R\)

**Symmetric:** \(\gcd(x, y) = 1 \implies \gcd(y, x) = 1\) so \((x, y) \in R \implies (y, x) \in R\)

**Not Antisymmetric:** \((2, 3) \in R\) and \((3, 2) \in R\), but \(2 \neq 3\)

**Not Transitive:** \(\gcd(2, 3) = 1, \gcd(3, 4) = 1, \gcd(2, 4) \neq 1\), so \((2, 3) \in R, (3, 4) \in R, \) but \((2, 4) \notin R\)

6.1 Use the Euclidean Algorithm to compute \(g = \gcd(429, 357)\)

\[
429 \% 357 = 72 \\
357 \% 72 = 69 \\
72 \% 69 = 3 \\
69 \% 3 = 0
\]

Thus \(g = 3\).

6.2 Find \(m\) and \(n\) so that \(2g = 429m + 357n\).

Use the Extended Euclidean Algorithm.

\[
429 = 1 \cdot 429 + 0 \cdot 357 \\
357 = 0 \cdot 429 + 1 \cdot 357 \\
72 = 1 \cdot 429 - 1 \cdot 357 \\
69 = -4 \cdot 429 + 5 \cdot 357 \\
3 = 5 \cdot 429 - 6 \cdot 357
\]

Thus, \(6 = 10 \cdot 429 - 12 \cdot 357\). \(m = 10\) and \(n = -12\)
1.1 Translate the following sentences into predicate logic. State what your predicate names mean.

\( A(x, y) \): \( x \) gets assigned task \( y \). \( E(x) \): \( x \) is an expert. \( H(x) \): \( x \) is happy. \( S(x, y) \): \( x \) solves task \( y \).

Implicitly, a task is only assigned to one person:

A. \( \forall y : \forall x : \forall z : (A(x, y) \land A(z, y)) \rightarrow x = z \)

1. A person is happy precisely when all tasks that are assigned to them are solved by someone.

\[ \forall x : H(x) \leftrightarrow \forall y : A(x, y) \rightarrow \exists z : S(z, y) \]

2. A task assigned to someone can only be solved by themselves or by an expert.

\[ \forall y : \forall x : A(x, y) \rightarrow \exists z : S(z, y) \rightarrow ((x = x) \lor E(z)), \text{ or} \]

\[ \forall y : \forall x : \forall z : (A(x, y) \land S(z, y)) \rightarrow ((z = x) \lor E(z)) \]

3. An expert does not solve a task unless that task is assigned to a person who is not an expert and does not solve the task themselves.

\[ \forall x : E(x) \rightarrow \forall y : S(x, y) \rightarrow \exists z : A(z, y) \land \neg E(z) \land \neg S(z, y) \text{ or} \]

\[ \forall x : \forall y : (E(x) \land S(z, y)) \rightarrow \exists z : A(z, y) \land \neg E(z) \land \neg S(z, y) \]

C. Any happy expert has no assigned tasks.

\[ \forall x : (E(x) \land H(x)) \rightarrow \neg \exists y : A(x, y) \]

1.2 Prove the conclusion \( C \) from the premises 1-3. Indicate the use of proof rules as clearly as possible.

We do a proof by contradiction, negating \( C \):

\[ \neg C = \neg \forall x : (E(x) \land H(x)) \rightarrow \neg \exists y : A(x, y) = \exists x : E(x) \land H(x) \land \exists y : A(x, y) \]

(4) \( E(a) \land H(a) \land A(a, b) \)

(2\times \text{ Instantiation of } \neg C, a, b \text{ new constants})

From (1), task \( b \) is solved by someone

(5) \( H(a) \)

Separation (4)

(6) \( H(a) \rightarrow \forall y : A(a, y) \rightarrow \exists z : S(z, y) \)

Specification (1, \( x = a \))

(7) \( \forall y : A(a, y) \rightarrow \exists z : S(z, y) \)

Modus Ponens (5, 6)

(8) \( A(a, b) \rightarrow \exists z : S(z, b) \)

Specification (7, \( y = b \))

(9) \( A(a, b) \)

Separation (4)

(10) \( \exists z : S(z, b) \)

Modus Ponens (8, 9)

(11) \( S(c, b) \)

Instantiation (10, \( c \) new constant)

From (2), we show \( c \) is an expert

(12) \( (A(a, b) \land S(c, b)) \rightarrow (a = c) \lor E(c) \)

3\times \text{ Specification (2, } x = a, y = b, z = c)\)

(13) \( A(a, b) \land S(c, b) \)

Conjunction (9, 11)

(14) \( (a = c) \lor E(c) \)

Modus Ponens (12, 13)

(15) \( E(a) \)

Separation (4)

(16) \( a = c \rightarrow E(c) \)

Substitution (15)

(17) \( E(c) \)

Proof by Cases (14, 16, split on \( a = c \))

From (3), we derive a contradiction

(18) \( (E(c) \land S(c, b)) \rightarrow \exists z : (A(z, b) \land (\neg E(z) \land \neg S(z, b)) \)

(19) \( E(c) \land S(c, b) \)

2\times \text{ Specification (3, } x = c, y = b)\)

(20) \( \exists z : A(z, b) \land \neg E(z) \land \neg S(z, b) \)

Conjunction (11, 17)

(21) \( A(c, b) \land \neg E(c) \land \neg S(c, b) \)

Modus Ponens (18, 19)

(22) \( A(c, b) \)

Existence (20, new constant \( c \))

(23) \( c = a \)

Separation (21)

(24) \( \neg E(a) \land \neg S(a, b) \)

Uniqueness of assignment (A, 9, 22)

(25) \( \neg E(a) \)

Separation (21), Substitution \( c = a \)

(26) \( E(a) \land \neg E(a) \)

Separation (24)

We have derived a contradiction, which completes our proof of the conclusion.
2.1 Convert the following formula to conjunctive normal form. Show your steps.

\[(a \rightarrow \neg(b \rightarrow \neg d)) \land (\neg(a \lor c) \rightarrow \neg d) \land ((b \rightarrow a) \rightarrow \neg d) \land \neg(b + c) \land (c \rightarrow \neg(b \rightarrow d))\]

\[\neg a \lor (b \land \neg d) \land (a \lor c \land \neg d) \land (\neg(b \rightarrow a) \land \neg d) \land (b \land c) \land (c \rightarrow b) \land (\neg c \lor (\neg b \land \neg d))\]

\[\neg a \lor (b \land d) \land (a \lor c \land \neg d) \land ((b \land \neg a) \land \neg d) \land (b \lor c) \land (b \lor \neg c) \land (\neg c \lor \neg b) \land (\neg c \lor \neg d)\]

2.2 Determine all truth assignments that satisfy the formula. Show your work.

We can simplify: \((-a \lor d) \land (-a \lor \neg d) = -a \land (d \lor \neg d) = -a\).

We duplicate \((-b \lor \neg c)\) by idempotence and simplify again using distributivity and excluded middle:

\((-b \lor c) \land (\neg b \lor \neg c) = -b \land (b \lor \neg c) \land (\neg b \lor \neg c) = \neg c\). These three unit clauses give \(a = b = c = F\).

The remaining clauses are: \((-a \land \neg b) \land (a \lor c \land \neg d) \land (b \lor \neg d) \land (\neg a \lor \neg d) \land (\neg b \lor c) \land (b \lor \neg c) \land (\neg b \lor \neg c) \land (\neg c \lor \neg d)\)

3. Consider the set \(C\) of all sets of languages over a nonempty alphabet \(\Sigma\). Define a relation \(R\) on \(C\) as follows: \(R(X,Y)\) iff for any two languages \(L_1, L_2 \in X\) we have \(L_1 \cup L_2 \in Y\). Show that \(R\) is antysymmetric and transitive. Is it reflexive or antireflexive?

Take arbitrary \(X\) and \(Y\) and specify \(L_1\) and \(L_2\) to the same arbitrary language \(L \in X\). Then \(R(X,Y)\) gives us \(L \in Y\), thus \(X \subseteq Y\). Likewise, \(R(Y,X)\) \(\rightarrow Y \subseteq X\); their conjunction gives us \(X = Y\). Since \(X\) and \(Y\) were arbitrary, generalization gives us antisymmetry.

For transitivity, take arbitrary \(X, Y, Z\), and arbitrary \(L_1, L_2 \in X\), resulting in \(L_1 \cup L_2 \in Y\). We have seen that \(R(Y,Z)\) \(\rightarrow Y \subseteq Z\), thus \(L_1 \cup L_2 \in Z\). This gives us \(R(X,Z)\) by definition, and by generalization, \(R\) is transitive.

\(R\) is not reflexive, take \(X = \{\{a\}, \{b\}\}\); since \(\{a, b\} \notin X\), \(R(X,X)\) is false.

\(R\) is also not antireflexive, \(R(\emptyset, \emptyset) = true\) (or we can take any \(X\) closed under union, such as the powerset of any set).

4.1 Let \(x, y, z, w\) be naturals. Assuming we know \(x\) is divisible by the three numbers \(y, z\), and \(yz + w\), prove or disprove that \(x\) is divisible by \(w\).

**False.** A counterexample is \(x = 3\), \(y = 1\), \(z = 1\) and \(w = 2\). As we see \(y, z\) and \(yz + w\) all divide \(x\), but \(w\) does not divide \(x\).

4.2 Let \(x, y, z, w\) be naturals. Assuming we know \(x\) divides the three numbers \(y, z\), and \(yz + w\), prove or disprove that \(x\) divides \(w\).

**True.** We know for some integers \(k, m, n\) we have: \(kx = y, mnx = z, nx = yz + w\).

Then \(w = nx - yz = nmx - kmx^2 = x(n - kmx)\), which shows \(w\) is divisible by \(x\).

5. Define the binary relation \(R\) on \(\mathbb{N}\) by \(R(x,y)\) iff \(x\) is the inverse of \(y\) modulo 2. Argue whether this relation is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

\((x, y) \in R\) if \(x\) is inverse of \(y\) mod 2. So \(xy \equiv 1\) mod 2 which means \(x\) and \(y\) are both odd numbers.

**Not Reflexive:** \((2, 2) \notin R\) (or any other even number)

**Not Antireflexive:** \((1, 1) \in R\) (or any other odd number)

**Symmetric:** \(xy \equiv 1\) mod 2 \(\leftrightarrow yx \equiv 1\) mod 2.

**Not AntiSymmetric:** \((1, 3) \in R\) and \((3, 1) \in R\), but \(1 \neq 3\)

**Transitive:** \((x, y) \in R \land (y, z) \in R\) means \(x, y\) and \(z\) are odd numbers and since any two odd numbers are in \(R\) we can say \((x, z) \in R\).

6. Find all solutions to the system of congruences:

\[x \equiv 4 \pmod{11}, x \equiv 1 \pmod{3}\] and \(x \equiv 5 \pmod{8}\)

First we check if all moduli are pairwise prime: \(\gcd(11, 3) = 1, \gcd(11, 8) = 1, \gcd(3, 8) = 1\)

Since they are relatively prime we can use the CRT to find the solution:

\[m_1 = 11, m_2 = 3, m_3 = 8, \quad M = 11 \times 3 \times 8 = 264, \quad M/m_1 = 24, M/m_2 = 88, M/m_3 = 33.\]

We compute the inverses: \(-5 \text{ of } 24 \text{ modulo } 11, 1 \text{ of } 88 \text{ modulo } 3, 1 \text{ of } 33 \text{ modulo } 8.

Then, \(x = (4 \times 24 \times -5) + (1 \times 88 \times 1) + (5 \times 33 \times 1) = -227\). Thus, \(x \equiv -227 \equiv 37 \pmod{264}\).