# The Language of Predicate Logic

- Domain: a non-empty set of objects
- Constants: concrete objects in the domain
- Variables: placeholders for concrete objects in the domain
- Functions: takes objects in the domain as arguments and returns an object of the domain.
- Predicates: takes objects in the domain as arguments and returns true or false. They describe properties of objects or relationships between objects.
- Quantifiers: for how many objects in the domain is the statement true?

solutions

# A question on functions

Consider two translations of the sentence "every child is younger than its mother." But every person has one and only one (biological) mother. Not very elegant!!

for every child $x$ and every mother $y$ of $x$, $x$ is younger than $y$.

1. $(\forall x\, (\forall y\, ((Child(x) \wedge Mother(y, x)) \rightarrow Younger(x, y))))$
2. $(\forall x\, (Child(x) \rightarrow Younger(x, mother(x))))$

This function replaces the predicate Mother and the quantifier $\forall y$

Which of the following is the best answer?

a. Both are wrong.

b. 1 is correct and 2 is wrong.

c. 2 is correct and 1 is wrong.

d. Both are correct. 1 is better.

e. Both are correct. 2 is better. Shorter & more elegant.

The domain is the set of people. $Child(x)$ means $x$ is a child. $Mother(x, y)$ means $x$ is $y$'s mother. $Younger(x, y)$ means $x$ is younger than $y$. $mother(x)$ returns $x$'s mother.



solutions.

# The Language of Predicate Logic

The seven kinds of symbols:

| | | | |
|---|---|---|---|
| | 1. | Constant symbols. | Usually $c, d, c_1, c_2, \dots, d_1, d_2 \dots$ |
| | 2. | Variables. | Usually $x, y, z, \dots x_1, x_2, \dots, y_1, y_2 \cdots$ |
| lowercase | 3. | Function symbols. | Usually $f, g, h, \dots f_1, f_2, \dots, g_1, g_2, \dots$ |
| uppercase | 4. | Predicate symbols. | $P$, $Q$, ...$P_1$, $P_2$, ..., $Q_1$, $Q_2$, ... |
| | 5. | Connectives: | $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ |
| | 6. | Quantifiers: | $\forall$ and $\exists$ |
| | 7. | Punctuation: | '(', ')', and ',' |

Function symbols and predicate symbols have an assigned *arity*—the number of arguments required. For example,

- $f^{(1)}$: $f$ is a unary function.
- $P^{(2)}$: $P$ is a binary predicate.



Solutions

# Terms

Each term refers to an object of the domain.

We define the set of terms inductively as follows.

Base case
1. Each constant symbol is an atomic term.

Base case
2. Each variable is an atomic term.

Inductive case
3. $f(t_1, \ldots, t_n)$ is a term if $t_1, \ldots, t_n$ are terms and $f$ is an $n$-ary function symbol. (If $f$ is a binary function symbol, then we may write $\underbrace{(t_1 \; f \; t_2)}_{\text{infix notation}}$ instead of $\underbrace{f(t_1, t_2)}_{\text{prefix notation.}}$.)
4. Nothing else is a term.

e.g. $x + y$

$a * b$



solutions

# Which expressions are terms?

A term refers to an object of the domain.

Which of the following expressions is a term?

~~a.~~ $g(d, d)$ g has 3 arguments.

~~b.~~ $P(f(x, y), d)$ This is T/F, not an object of the domain.

~~c.~~ $f(x, g(y, z), d)$ g has 3 arguments, f has 2 arguments.

d. $g(x, f(y, z), d)$

Let $d$ be a constant symbol. Let $P$ be a predicate symbol with 2 arguments. Let $f$ be a function symbol with 2 arguments and $g$ be a function symbol with 3 arguments. Let $x$, $y$, and $z$ be variable symbols.



Solutions

# Is this a term?

True or False: The expression $(2 - f(x)) + (y * x)$ is a term.

Infix notation.

a. True

b. False

c. Not enough information to tell

The domain is the set of integers. $+$, $-$ and $*$ are binary functions. $f$ is a unary function. $x$ and $y$ are variables and $2$ is a constant.



solutions

# Well-Formed Predicate Logic Formulas

We define the set of well-formed formulas of predicate logic inductively as follows.

a predicate has at least 1 argument

1. $P(t_1, \ldots, t_n)$ is an atomic formula if $P$ is an $n$-ary predicate symbol and each $t_i$ is a term $(1 \leq i \leq n)$. this requires you to understand the definition of a term.
2. $(\neg\alpha)$ is a formula if $\alpha$ is a formula.
3. $(\alpha \star \beta)$ is a formula if $\alpha$ and $\beta$ are formulas and $\star$ is a binary connective symbol.
4. Each of $(\forall x\ \alpha)$ and $(\exists x\ \alpha)$ is a formula if $\alpha$ is a formula and $x$ is a variable.
5. Nothing else is a formula.



solutions

# Determine whether a formula is well-formed

$m$ is a constant and $x$ and $y$ are variables. $P^{(2)}$ and $Q^{(2)}$ are binary predicates. $f^{(1)}$ is a unary function.

Which of the following is a well-formed predicate logic formula?

~~a.~~ $(f(x) \rightarrow P(x, y))$ f(x) is not T/F. cannot be the first part of an →.

~~b.~~ $(\forall y\, P(m, f(y)))$ missing outmost bracket.

~~c.~~ $P(x, y) \rightarrow Q(Q(x))$ Q is binary and requires 2 arguments.

d. $Q(m, f(m))$

~~e.~~ $P(m, f(\underbrace{Q(x, y)}_{T/F}))$ f requires an object of the domain as its argument, but is given T/F.



Solutions

# Well-Formed Predicate Logic Formulas

e.g. Alice teaches CS245. In Pred Logic, let $T(x)$ mean $x$ teaches CS245.
In Prop Logic, define this to be $\underline{t}$. This is $\underline{T(Alice)}$.

We define the set of well-formed formulas of predicate logic inductively as follows.

Base case

△ 1. $\overbrace{P(t_1, \dots, t_n)}^{\text{corresponds to a propositional variable. it's T/F.}}$ is an atomic formula if $P$ is an $n$-ary predicate symbol and each $t_i$ is a term $(1 \le i \le n)$.

Inductive cases

2. $(\neg\alpha)$ is a formula if $\alpha$ is a formula.
3. $(\alpha \star \beta)$ is a formula if $\alpha$ and $\beta$ are formulas and $\star$ is a binary connective symbol.

△ 4. Each of $(\forall x\ \alpha)$ and $(\exists x\ \alpha)$ is a formula if $\alpha$ is a formula and $x$ is a variable. (similar to case 2 since $\forall x$ and $\exists x$ are unary, like $\neg$.)

5. Nothing else is a formula.

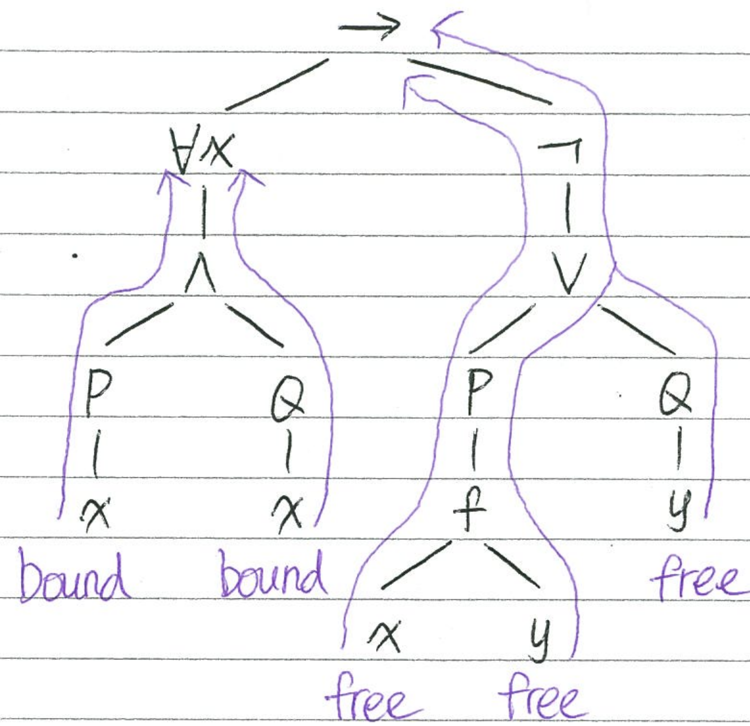
2, 3, 5 are the same as Prop Logic.



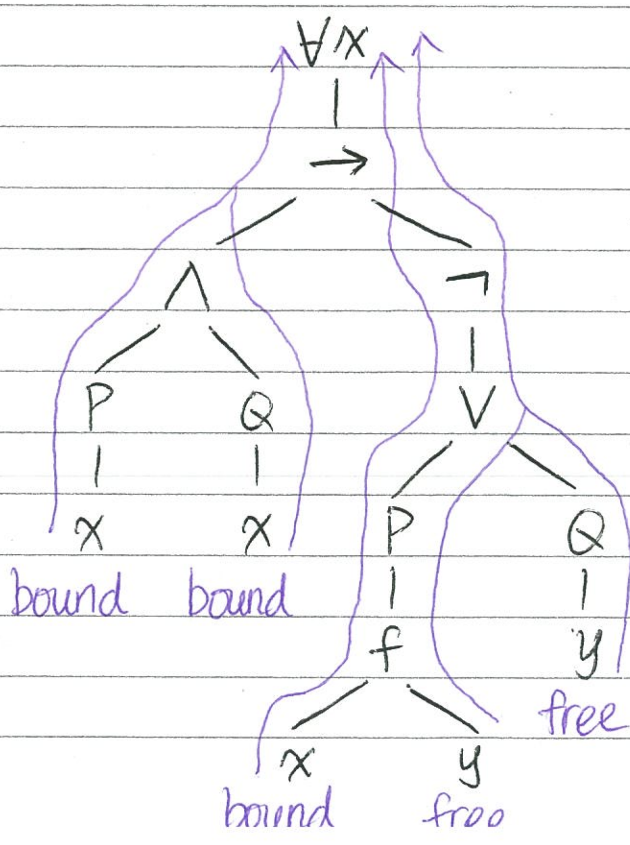
Solutions

Parse Trees, Free and Bound Variables

① $\alpha : (\forall x\ (P(x) \wedge Q(x))) \rightarrow (\neg (P(f(x, y)) \vee Q(y)))$




② $\beta : (\forall x\ ((P(x) \wedge Q(x)) \rightarrow (\neg (P(f(x, y)) \vee Q(y)))))$




solutions

# Substitution

Oct 16

Example : Variables: $\{x, y, z\}$ Functions: $\{f^{(2)}, g^{(2)}, h^{(2)}\}$
Predicates: $\{P^{(1)}, Q^{(1)}\}$.

Consider the following formulas:

$\alpha : ((\forall x (P(x) \land Q(x))) \to (\neg (P(f(x,y)) \lor Q(y))))$

Give the results of the following substitutions:

(1) $\alpha[g(x,y)/y]$ (2) $\alpha[h(x,y)/x]$

(1) $\alpha = ((\forall x (P(x) \land Q(x))) \to (\neg (P(f(x,y)) \lor Q(y))))$
There are two occurrences of $y$ and both occurrences of $y$ are free. The substitution will replace both occurrences.

$\alpha[g(x,y)/y]$
$= ((\forall x (P(x) \land Q(x))) \to (\neg (P(f(x, g(x,y))) \lor Q(g(x,y)))))$

(2) There are three occurrences of $x$ in $\alpha$. From the left, the first two occurrences of $x$ are bound and the third occurrence of $x$ is free. We only need to replace the third occurrence of $x$ by $h(x,y)$ in the substitution.

$\alpha[h(x,y)/x]$
$= ((\forall x (P(x) \land Q(x))) \to (\neg (P(f(h(x,y), y))) \lor Q(y))))$.

solutions

Substitution. Oct 17

Variables $\{x, y, z\}$ Functions $\{f^{(2)}, g^{(2)}, h^{(2)}\}$.
Predicates $\{P^{(1)}, Q^{(1)}\}$.

$\beta = (\forall x\, ((P(x) \wedge Q(x)) \rightarrow (\neg (P(f(x,y)) \vee Q(y)))))$.

State the results of the following substitutions.

(1) $\beta[g(x,y)/x]$ (2) $\beta[h(x,y)/y]$

(1) In $\beta$, all three occurrences of $x$ are bound.
Thus, any substitution does not affect $x$ in any way.

$\beta[g(x,y)/x] = \beta$

(2) In $\beta$, both occurrences of $y$ are free.
Thus, we need to replace both occurrences in the substitution

$\beta[h(x,y)/y]$
$= (\forall x\, ((P(x) \wedge Q(x)) \rightarrow (\neg (P(f(x, h(x,y))) \vee Q(h(x,y))))))$

- Wait, was the substitution OK? NO.
  - Before the substitution, the $y$'s were free.
  - After the substitution, the $x$ in $h(x,y)$ becomes bound by the leading quantifier $\forall x$. This is A PROBLEM!!!

This problem is called "capture".
We should avoid this by performing a careful substitution instead.

# Substitution: Why is "capture" problematic? Oct 16

Variables : $\{x, y, z\}$ Predicates $\{L^{(2)}\}$
Domain : the set of all people $L(x, y)$ means $x$ likes $y$.

Consider the formula $\gamma = (\forall x\ L(x, y))$

(1) Translate $\gamma$ to English.
Everyone likes $y$.

(2) State the result of the substitution: $\gamma[x/y]$

The only occurrence of $y$ in $\gamma$ is free.
So we need to replace it with the substitution.

$$\gamma[x/y] = (\forall x\ L(x, x))$$

The new $x$ is captured by the quantifier $\forall x$.

(3). Translate $\gamma[x/y]$ to English.
Everyone likes him/herself.

"Capture" caused the meaning of the formula to change because of the substitution. This is problematic!