Discrete_Mathematics_Tutorial_7

16 October 2025 19:34

Discrete_M...

CS2020A Discrete Mathematics

Tutorial 7 Submission

Submitted By

Kalakuntla Parjanya	**112401018**
Prachurjya Pratim Goswami	**102401023**
Chavva Srinivasa Saketh	**112401009**
Madeti Tarini	**112401020**
Vedant Singh	**142401041**

Problem 1. For each of the first order formula below over the domain $D = \{1, \ldots, 100\}$,

a) Draw its parse tree

b) Write a single python function using nested loops which evaluates the formula

c) Write a python program using multiple functions each using a single loop to

c) Write a python program using multiple functions each using a single loop to evaluate the formula

(You can assume that all the predicates are implemented and you can call them by their names supplying the correct number of arguments)

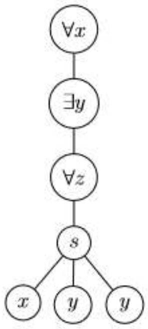
a) $\forall x \exists y \forall z\ S(x, y, z)$

b) $\forall x(P(x) \to \exists y(Q(y) \land \forall z(R(z) \lor S(x, y, z))))$

Solution.

a) $\forall x \exists y \forall z\ S(x, y, z)$

(i) **Parse Tree:**



(ii) **Single Python Function:**

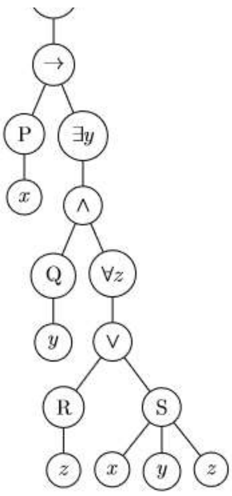
```
def AxEyAzS():
    for x in D:
    found_y = False
    for y in D:
        ok = True
        for z in D:
            if not S(x, y, z):
                ok = False
                break
        if ok:
            found_y = True
            break
    if not found_y:
        return False
return True
```


(iii) **Multiple Python Functions:**

```
# S(x,y,z)  : S
# P(x,y)    : forall z S
# Q(x)      : there exists y for all z S
# R()       : forall x there exists y forall z S

def P(x,y):
    for z in D:
        if not S(x,y,z):
            return false
    return true

def Q(x):
    for y in D:
        if Q(x,y):
            return true
    return false

def R():
    for x in D:
        if not Q(x):
            return false
    return true
```

b) $\forall x(P(x) \to \exists y(Q(y) \land \forall z(R(z) \lor S(x, y, z))))$

(i) **Parse Tree:**



(ii) **Single Python Function:**

```
def fun():
    for x in D:
        if P(x):
            found_y = false
            for y in D:
                if not Q(y):
                    break
                ok = true
                for z in D:
                    if not(R(z) or S(x,y,z)):
                        ok = false
                        break
                    if ok:
                        found_y = true
                        break
            if not found_y:
                return false
    return true
```

(iii) **Multiple Python Functions:**

```
# f(x,y)   : forall z (R(z) or S(x,y,z))
# g(x)     : there exists y (Q(y) and f(x,y))
# h()      : forall x (P(x) -> g(x))

def f(x,y):
    for z in D:
        if not(R(z) or S(x,y,z)):
            return false
    return true

def g(x):
    for y in D:
        if(Q(y) and f(x,y)):
            return true
    return false

def h():
    for x in D:
        if P(x) and not g(x):
            return false
    return true
```

Problem 2. Definition 1. A function $f : \mathbb{R} \to \mathbb{R}$ is called *continuous* if for every real number $\epsilon > 0$ and for every $x \in \mathbb{R}$ there exists a real number $\delta > 0$ such that for every $y \in \mathbb{R}$ with $|x - y| < \delta$, we have that $|f(x) - f(y)| < \epsilon$.

Definition 2. A function $f : \mathbb{R} \to \mathbb{R}$ is called *uniformly continuous* if for every real number $\epsilon > 0$ there exists a real number $\delta > 0$ such that for every $x, y \in \mathbb{R}$ with $|x-y| < \delta$, we have that $|f(x) - f(y)| < \epsilon$.

a) Express the two definitions above as first order sentences. Apart from the logical symbols and the function symbol f, you have to define all the other symbols that you use in the formulae. Do not use shorthands like $\forall(\epsilon > 0)$.

b) Draw parse-trees for these formulae side by side.

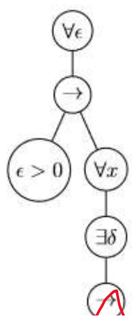
c) Which one do you think is a stronger requirement, that is does one of the continuity definitions follow as a logical consequence of the other?

Solution.

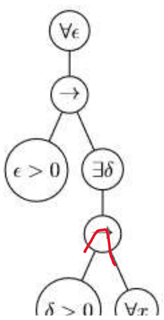
a) $\mathbb{D} = \mathbb{R}$
Statement 1: $\forall\epsilon[(\epsilon > 0) \to \forall x\exists\delta((\delta > 0) \to \forall y((-\delta < (x - y) < \delta) \to (-\epsilon < (f(x) - f(y)) < \epsilon)))]$

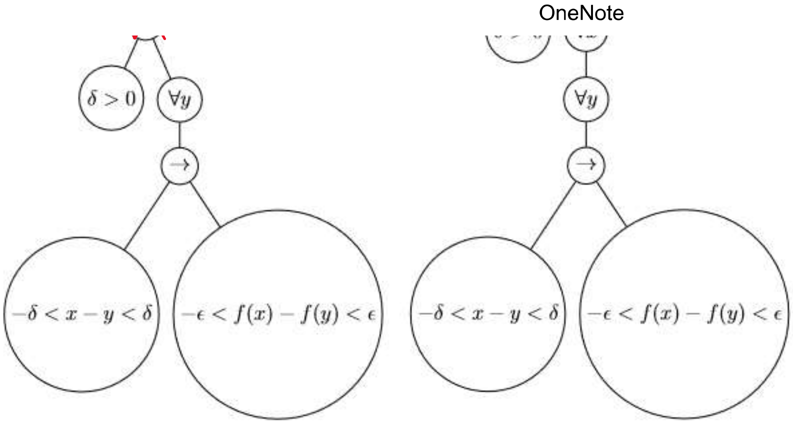
Statement 2: $\forall\epsilon[(\epsilon > 0) \to \exists\delta((\delta > 0) \to \forall x\forall y((-\delta < (x - y) < \delta) \to (-\epsilon < (f(x) - f(y)) < \epsilon)))]$

b) Statement 1 Parse tree:



Statement 2 Parse tree:





c) If there exists a particular δ for all y(statement 2), then for all y there always exists a δ(statement 1).
i.e, Statement 2 $\rightarrow$ Statement 1.
Thus, Statement 2 is a stronger requirement.

Problem 3. Let P be any ternary predicate, and

$$\begin{aligned}
\alpha_1 &= \forall x \forall y \exists z\ P(x,y,z) \\
\alpha_2 &= \forall x \exists z \forall y\ P(x,y,z) \\
\alpha_3 &= \exists z \forall x \forall y\ P(x,y,z) \\
\alpha_4 &= \forall x \exists y \exists z\ P(x,y,z) \\
\alpha_5 &= \forall x \exists z \exists y\ P(x,y,z) \\
\alpha_6 &= \exists z \forall x \exists y\ P(x,y,z)
\end{aligned}$$

a) Identify and count all ordered pairs (i,j) such that $\alpha_i \implies \alpha_j$ and justify why each of the logical consequences you identified is true.

b) Identify and count all ordered pairs (i,j) such that α_j is not a logical consequence of α_i and find a model for P in which α_i evaluates to True and α_j evaluates to False.

Solution.

a) Pairs where $\alpha_i \implies \alpha_j$

There are **21** such ordered pairs. They are:

- (1,1), (1,4), (1,5), (1,6)
- (2,1), (2,2), (2,4), (2,5)
- (3,1), (3,2), (3,3), (3,4), (3,5), (3,6)
- (4,4), (4,5)
- (5,4), (5,5)
- (6,4), (6,5), (6,6)

Justification?

b) Pairs where α_j is NOT a logical consequence of α_i

There are **15** such ordered pairs. They are:

- (1,2), (1,3), (1,6)
- (2,3), (2,6)
- (4,1), (4,2), (4,3), (4,6)
- (5,1), (5,2), (5,3), (5,6)
- (6,1), (6,2), (6,3)

Model?