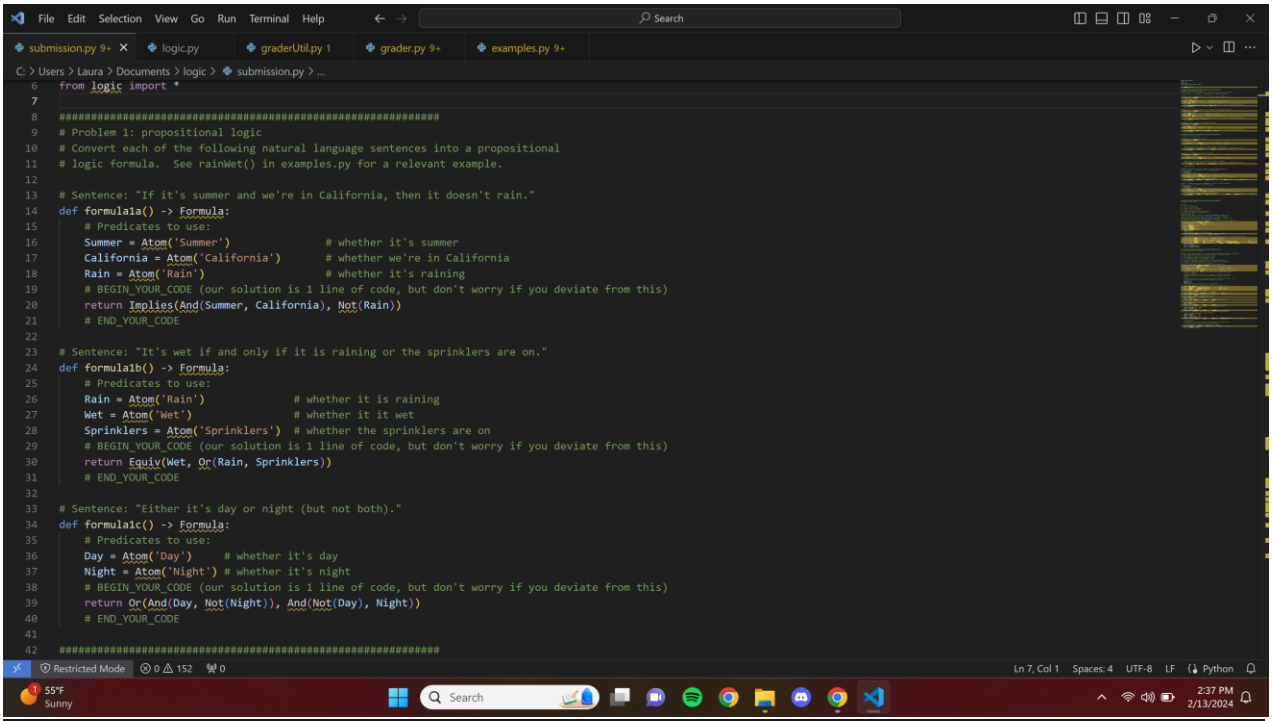
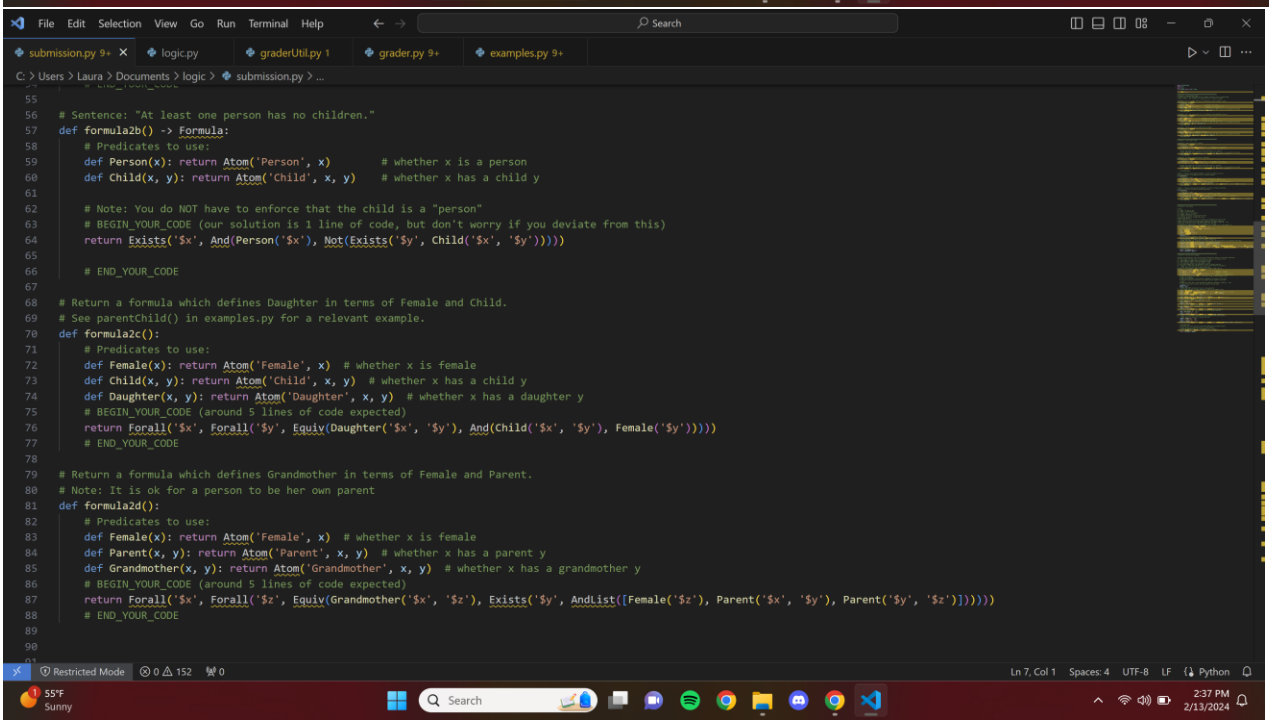
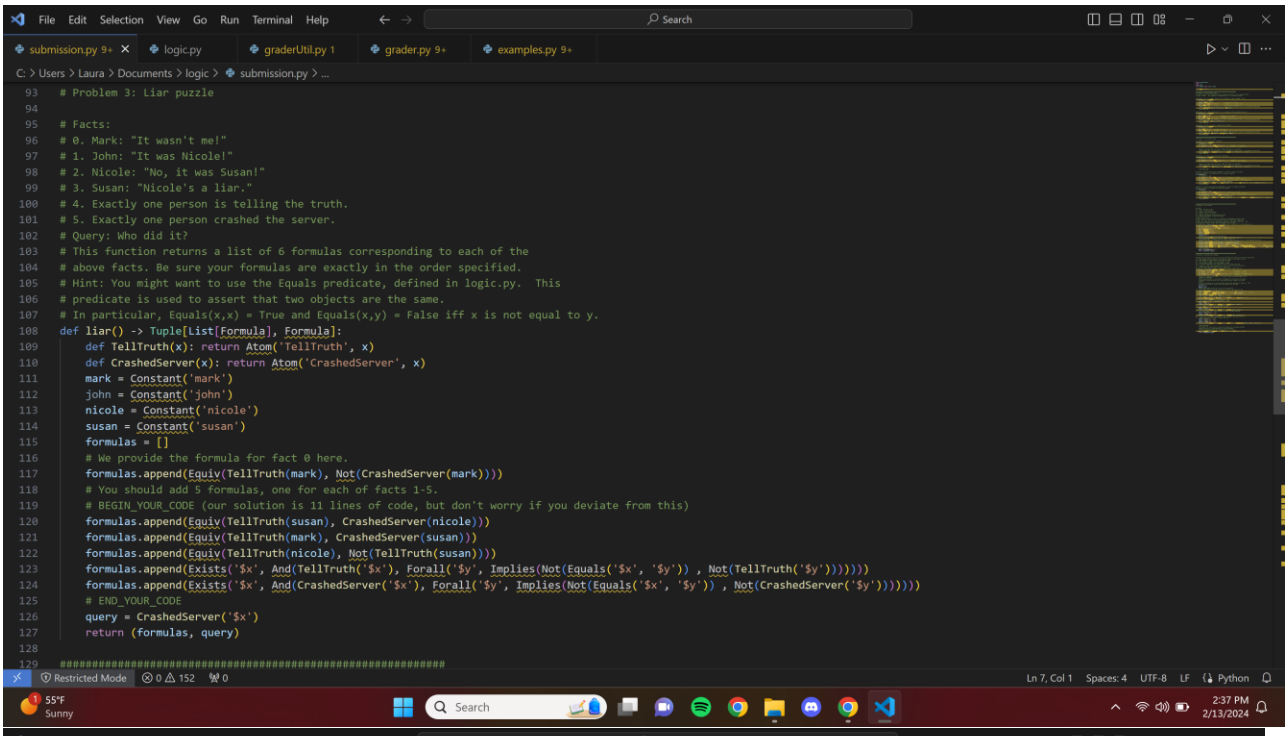
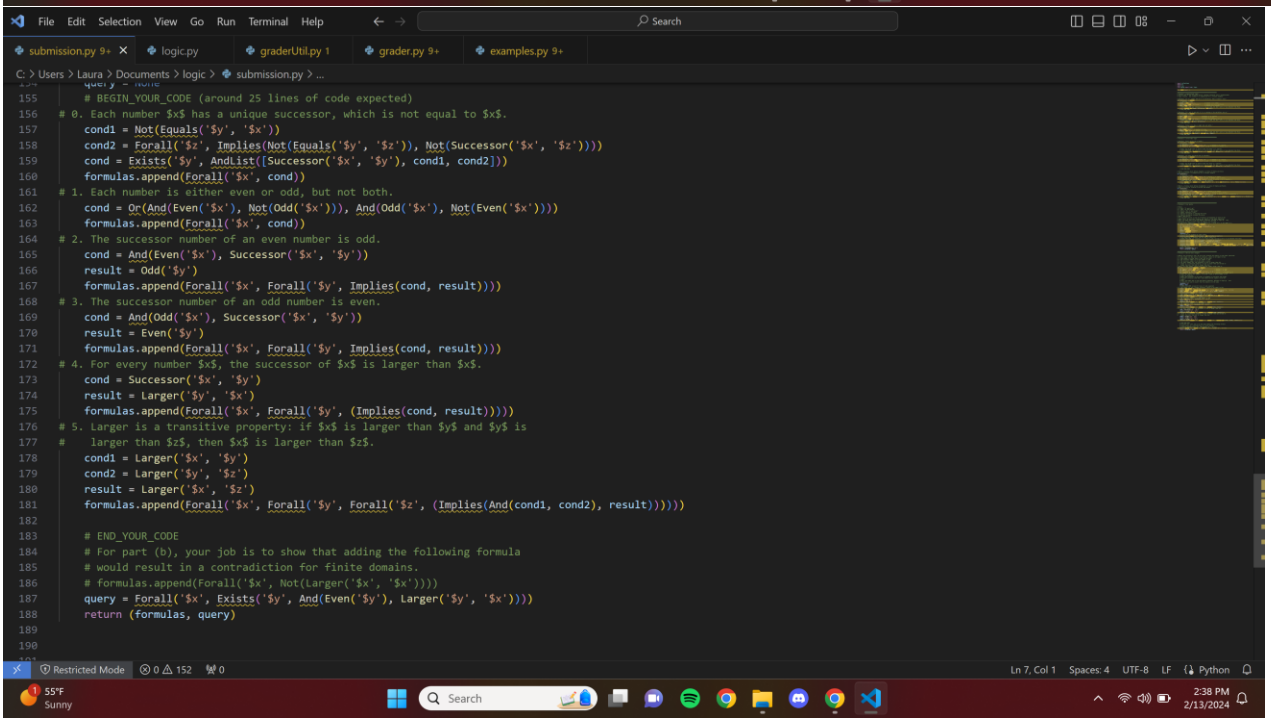

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > submission.py > ...
from logic import *
##
# Problem 1: propositional logic
# Convert each of the following natural language sentences into a propositional
# logic formula. See rainWet() in examples.py for a relevant example.
# Sentence: "If it's summer and we're in California, then it doesn't rain."
def formula1a() -> Formula:
# Predicates to use:
Summer = Atom('Summer') # whether it's summer
California = Atom('California') # whether we're in California
Rain = Atom('Rain') # whether it's raining
# BEGIN_YOUR_CODE (our solution is 1 line of code, but don't worry if you deviate from this)
return Implies(And(Summer, California), Not(Rain))
# END_YOUR_CODE
# Sentence: "It's wet if and only if it is raining or the sprinklers are on."
def formula1b() -> Formula:
# Predicates to use:
Rain = Atom('Rain') # whether it is raining
Wet = Atom('Wet') # whether it it wet
Sprinklers = Atom('Sprinklers') # whether the sprinklers are on
# BEGIN_YOUR_CODE (our solution is 1 line of code, but don't worry if you deviate from this)
return Equiv(Wet, Or(Rain, Sprinklers))
# END_YOUR_CODE
# Sentence: "Either it's day or night (but not both)."
def formula1c() -> Formula:
# Predicates to use:
Day = Atom('Day') # whether it's day
Night = Atom('Night') # whether it's night
# BEGIN_YOUR_CODE (our solution is 1 line of code, but don't worry if you deviate from this)
return Or(And(Day, Not(Night)), And(Not(Day), Night))
END_YOUR_CODE
##
Restricted Mode
Ln 7, Col 1
Spaces: 4
UTF-8
LF
Python
55°F
Sunny
Search
2:37 PM
2/13/2024

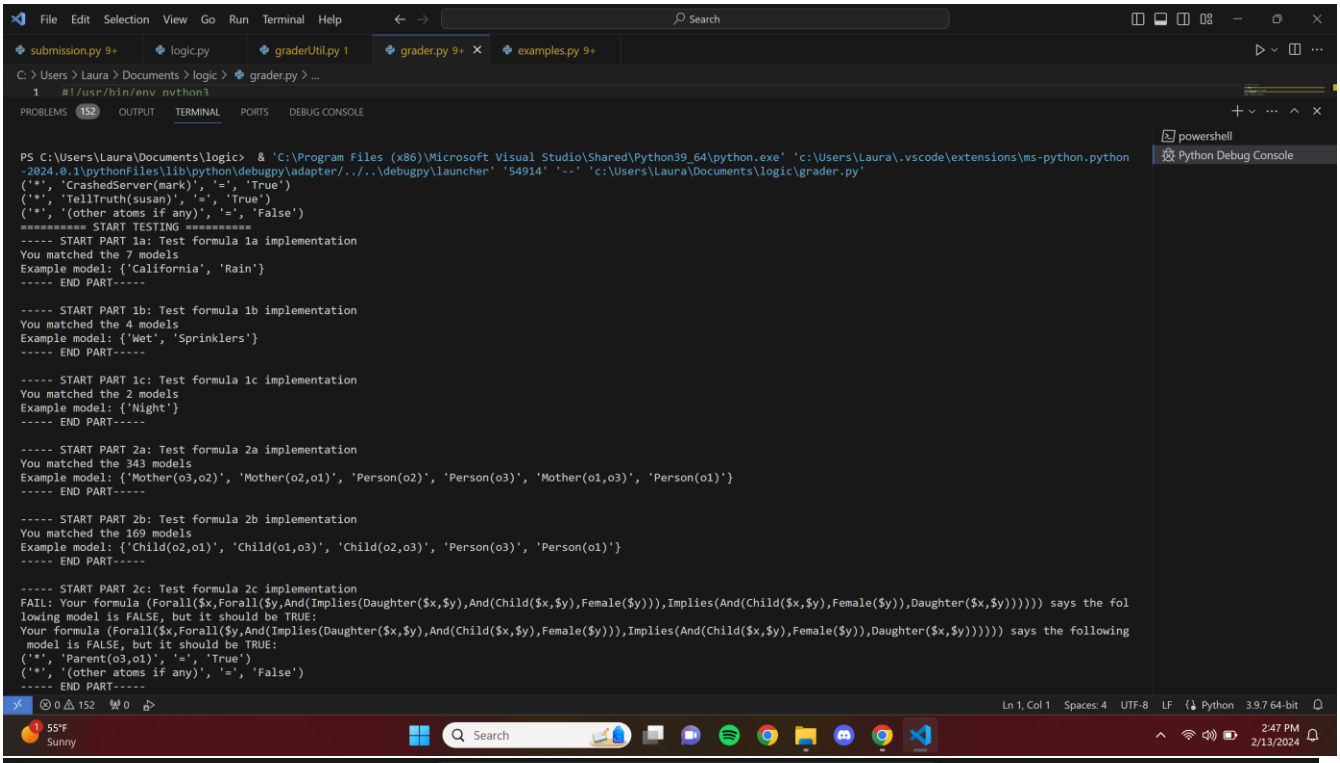

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > submission.py > ...
# Sentence: "At least one person has no children."
def formula2b() -> Formula:
# Predicates to use:
def Person(x): return Atom('Person', x) # whether x is a person
def Child(x, y): return Atom('Child', x, y) # whether x has a child y
Note: You do NOT have to enforce that the child is a "person"
BEGIN_YOUR_CODE (our solution is 1 line of code, but don't worry if you deviate from this)
return Exists('$x', And(Person('$x'), Not(Exists('$y', Child('$x', '$y')))))
# END_YOUR_CODE
# Return a formula which defines Daughter in terms of Female and Child.
# See parentChild() in examples.py for a relevant example.
def formula2c():
# Predicates to use:
def Female(x): return Atom('Female', x) # whether x is female
def Child(x, y): return Atom('Child', x, y) # whether x has a child y
def Daughter(x, y): return Atom('Daughter', x, y) # whether x has a daughter y
BEGIN_YOUR_CODE (around 5 lines of code expected)
return Forall('$x', Forall('$y', Equiv(Daughter('$x', '$y'), And(Child('$x', '$y'), Female('$y')))))
# END_YOUR_CODE
# Return a formula which defines Grandmother in terms of Female and Parent.
# Note: It is ok for a person to be her own parent
def formula2d():
# Predicates to use:
def Female(x): return Atom('Female', x) # whether x is female
def Parent(x, y): return Atom('Parent', x, y) # whether x has a parent y
def Grandmother(x, y): return Atom('Grandmother', x, y) # whether x has a grandmother y
BEGIN_YOUR_CODE (around 5 lines of code expected)
return Forall('$x', Forall('$z', Equiv(Grandmother('$x', '$z'), Exists('$y', AndList([Female('$z'), Parent('$x', '$y'), Parent('$y', '$z')])))))
END_YOUR_CODE
Restricted Mode
Ln 7, Col 1
Spaces: 4
UTF-8
LF
Python
55°F
Sunny
Search
2:37 PM
2/13/2024

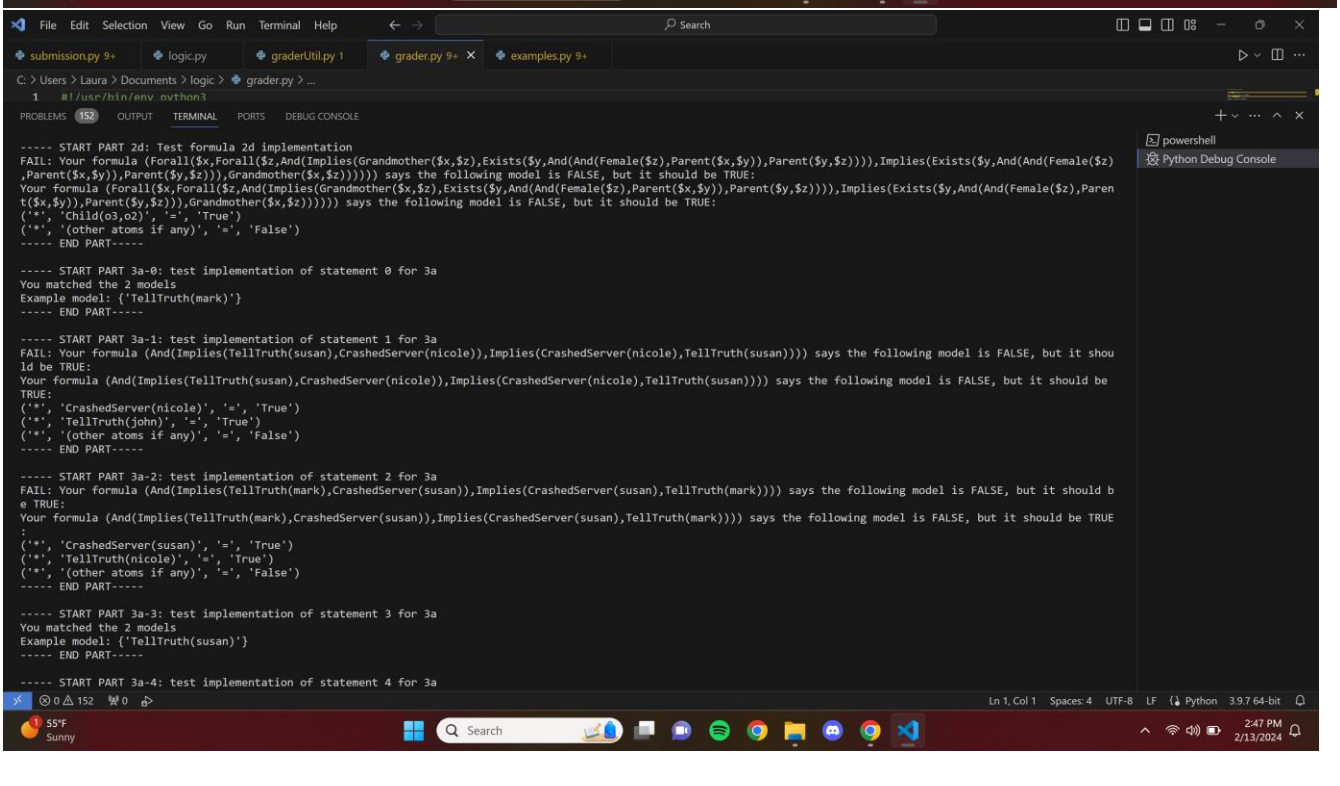


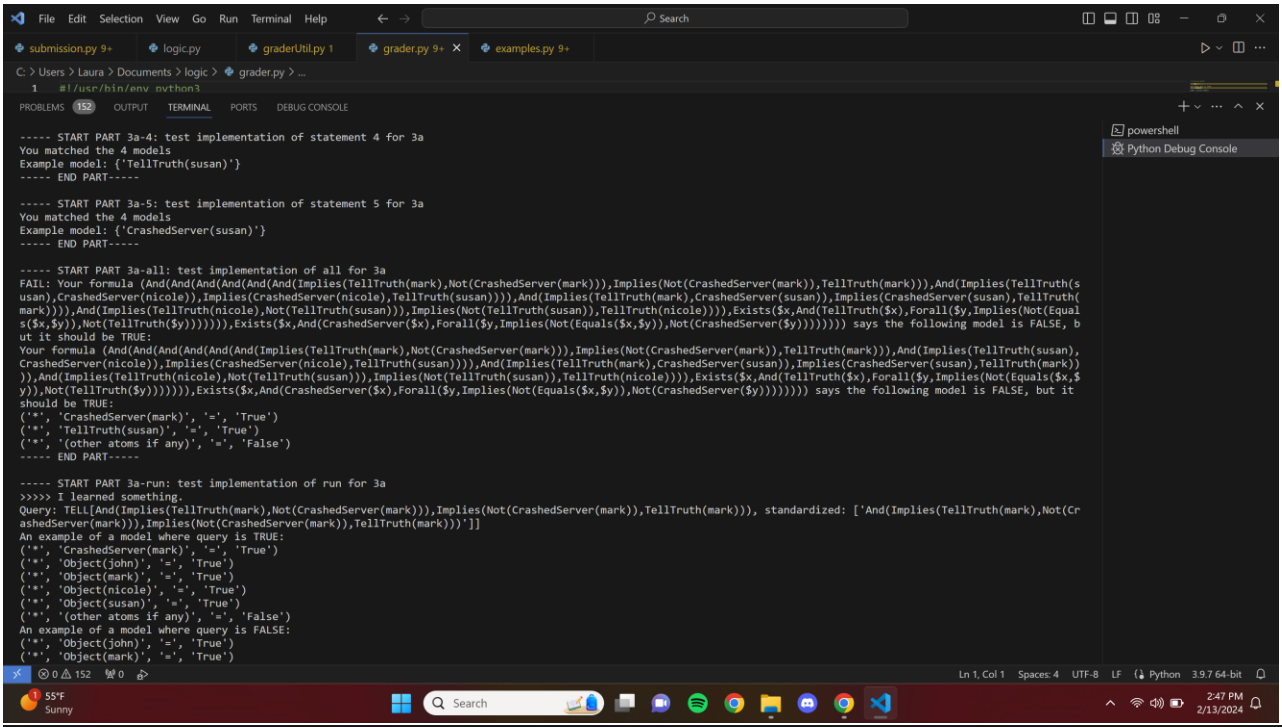
```
# Problem 3: Liar puzzle

# Facts:
# 0. Mark: "It wasn't me!"
# 1. John: "It was Nicole!"
# 2. Nicole: "No, it was Susan!"
# 3. Susan: "Nicole's a liar."
# 4. Exactly one person is telling the truth.
# 5. Exactly one person crashed the server.
# Query: Who did it?
# This function returns a list of 6 formulas corresponding to each of the
# above facts. Be sure your formulas are exactly in the order specified.
# Hint: You might want to use the Equals predicate, defined in logic.py.  This
# predicate is used to assert that two objects are the same.
# In particular, Equals(x,x) = True and Equals(x,y) = False iff x is not equal to y.
def liar() -> Tuple[List[Formula], Formula]:
    def TellTruth(x): return Atom('TellTruth', x)
    def CrashedServer(x): return Atom('CrashedServer', x)
    mark = Constant('mark')
    john = Constant('john')
    nicole = Constant('nicole')
    susan = Constant('susan')
    formulas = []
    # We provide the formula for fact 0 here.
    formulas.append(Equiv(TellTruth(mark), Not(CrashedServer(mark))))
    # You should add 5 formulas, one for each of facts 1-5.
    # BEGIN_YOUR_CODE (our solution is 11 lines of code, but don't worry if you deviate from this)
    formulas.append(Equiv(TellTruth(susan), CrashedServer(nicole)))
    formulas.append(Equiv(TellTruth(mark), CrashedServer(susan)))
    formulas.append(Equiv(TellTruth(nicole), Not(TellTruth(susan))))
    formulas.append(Exists('$x', And(TellTruth('$x'), Forall('$y', Implies(Not(Equals('$x', '$y')) , Not(TellTruth('$y')))))))
    formulas.append(Exists('$x', And(CrashedServer('$x'), Forall('$y', Implies(Not(Equals('$x', '$y')) , Not(CrashedServer('$y')))))))
    # END_YOUR_CODE
    query = CrashedServer('$x')
    return (formulas, query)

############################################################
```

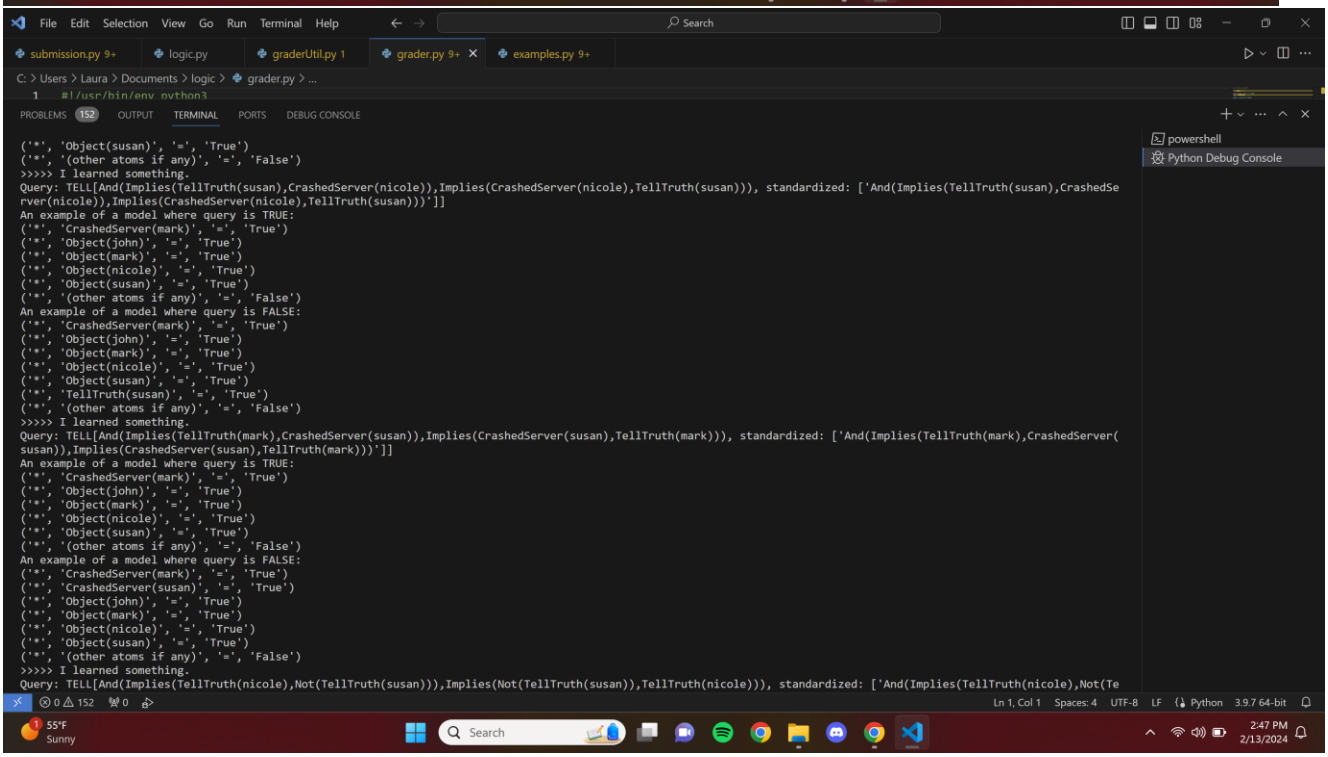


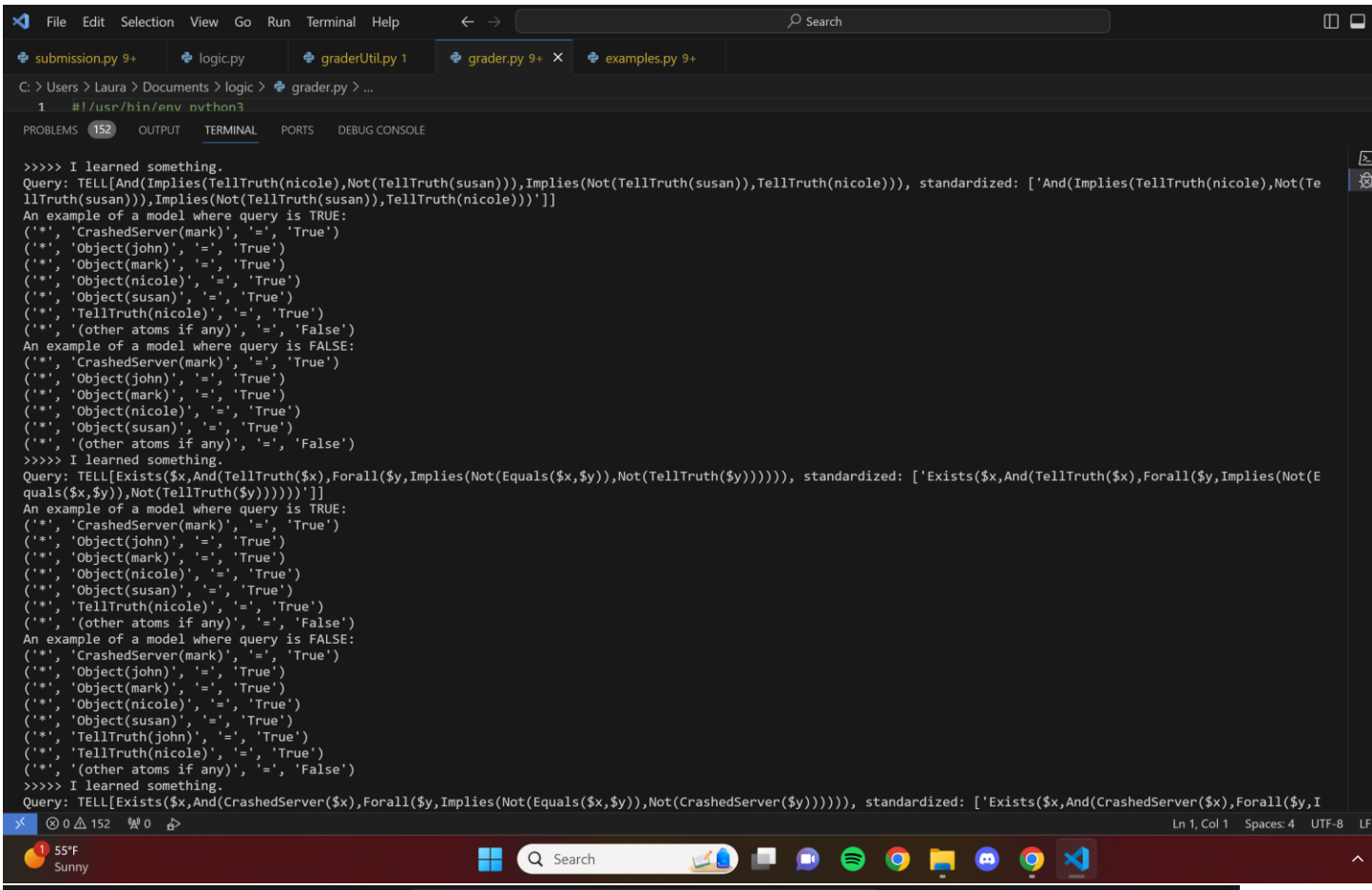
```
    # BEGIN_YOUR_CODE (around 25 lines of code expected)
# 0. Each number $x$ has a unique successor, which is not equal to $x$.
    cond1 = Not(Equals('$y', '$x'))
    cond2 = Forall('$z', Implies(Not(Equals('$y', '$z')), Not(Successor('$x', '$z'))))
    cond = Exists('$y', AndList([Successor('$x', '$y'), cond1, cond2]))
    formulas.append(Forall('$x', cond))
# 1. Each number is either even or odd, but not both.
    cond = Or(And(Even('$x'), Not(Odd('$x'))), And(Odd('$x'), Not(Even('$x'))))
    formulas.append(Forall('$x', cond))
# 2. The successor number of an even number is odd.
    cond = And(Even('$x'), Successor('$x', '$y'))
    result = Odd('$y')
    formulas.append(Forall('$x', Forall('$y', Implies(cond, result))))
# 3. The successor number of an odd number is even.
    cond = And(Odd('$x'), Successor('$x', '$y'))
    result = Even('$y')
    formulas.append(Forall('$x', Forall('$y', Implies(cond, result))))
# 4. For every number $x$, the successor of $x$ is larger than $x$.
    cond = Successor('$x', '$y')
    result = Larger('$y', '$x')
    formulas.append(Forall('$x', Forall('$y', (Implies(cond, result)))))
# 5. Larger is a transitive property: if $x$ is larger than $y$ and $y$ is
#    larger than $z$, then $x$ is larger than $z$.
    cond1 = Larger('$x', '$y')
    cond2 = Larger('$y', '$z')
    result = Larger('$x', '$z')
    formulas.append(Forall('$x', Forall('$y', Forall('$z', (Implies(And(cond1, cond2), result))))))

    # END_YOUR_CODE
    # For part (b), your job is to show that adding the following formula
    # would result in a contradiction for finite domains.
    # formulas.append(Forall('$x', Not(Larger('$x', '$x'))))
    query = Forall('$x', Exists('$y', And(Even('$y'), Larger('$y', '$x'))))
    return (formulas, query)
```

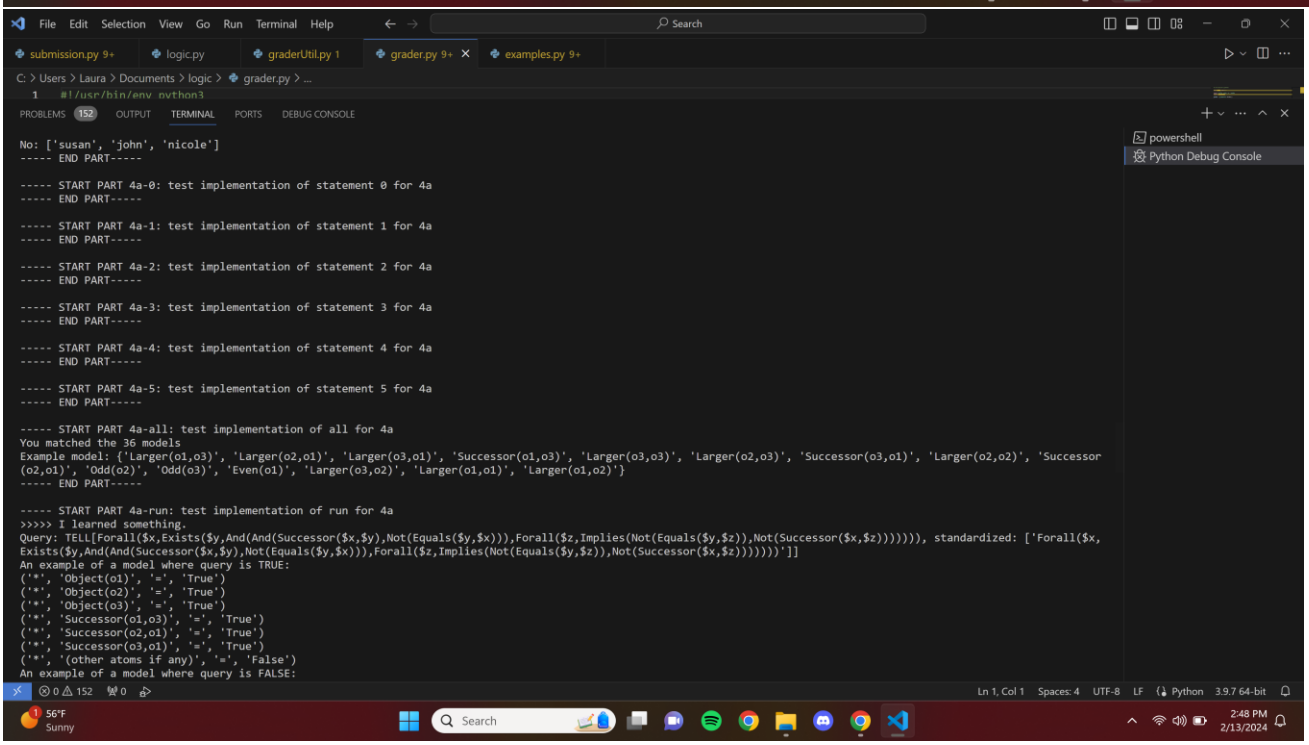

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
PS C:\Users\Laura\Documents\logic> & 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe' 'c:\Users\Laura\.vscode\extensions\ms-python.python-2024.0.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '54914' '--' 'c:\Users\Laura\Documents\logic\grader.py'
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'TellTruth(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
========== START TESTING ==========
----- START PART 1a: Test formula 1a implementation
You matched the 7 models
Example model: {'California', 'Rain'}
----- END PART-----
----- START PART 1b: Test formula 1b implementation
You matched the 4 models
Example model: {'Wet', 'Sprinklers'}
----- END PART-----
----- START PART 1c: Test formula 1c implementation
You matched the 2 models
Example model: {'Night'}
----- END PART-----
----- START PART 2a: Test formula 2a implementation
You matched the 343 models
Example model: {'Mother(o3,o2)', 'Mother(o2,o1)', 'Person(o2)', 'Person(o3)', 'Mother(o1,o3)', 'Person(o1)'}
----- END PART-----
----- START PART 2b: Test formula 2b implementation
You matched the 169 models
Example model: {'Child(o2,o1)', 'Child(o1,o3)', 'Child(o2,o3)', 'Person(o3)', 'Person(o1)'}
----- END PART-----
----- START PART 2c: Test formula 2c implementation
FAIL: Your formula (Forall($x,Forall($y,And(Implies(Daughter($x,$y),And(Child($x,$y),Female($y))),Implies(And(Child($x,$y),Female($y)),Daughter($x,$y)))))) says the following model is FALSE, but it should be TRUE:
Your formula (Forall($x,Forall($y,And(Implies(Daughter($x,$y),And(Child($x,$y),Female($y))),Implies(And(Child($x,$y),Female($y)),Daughter($x,$y)))))) says the following model is FALSE, but it should be TRUE:
('*', 'Parent(o3,o1)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
----- END PART-----
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
55°F Sunny
Search
2:47 PM 2/13/2024

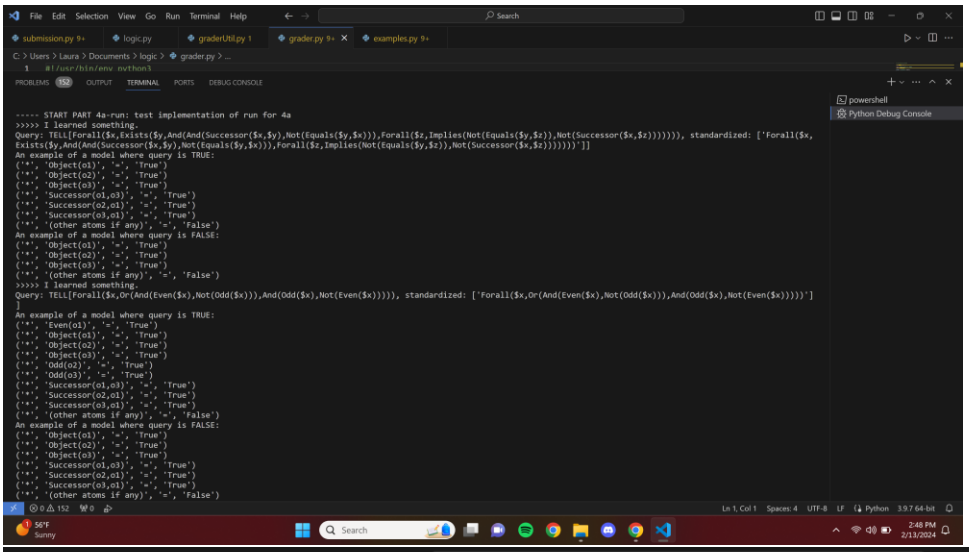

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
----- START PART 2d: Test formula 2d implementation
FAIL: Your formula (Forall($x,Forall($z,And(Implies(Grandmother($x,$z),Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z)))),Implies(Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z))),Grandmother($x,$z)))))) says the following model is FALSE, but it should be TRUE:
Your formula (Forall($x,Forall($z,And(Implies(Grandmother($x,$z),Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z)))),Implies(Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z))),Grandmother($x,$z)))))) says the following model is FALSE, but it should be TRUE:
('*', 'Child(o3,o2)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
----- END PART-----
----- START PART 3a-0: test implementation of statement 0 for 3a
You matched the 2 models
Example model: {'TellTruth(mark)'}
----- END PART-----
----- START PART 3a-1: test implementation of statement 1 for 3a
FAIL: Your formula (And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))) says the following model is FALSE, but it should be TRUE:
Your formula (And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))) says the following model is FALSE, but it should be TRUE:
('*', 'CrashedServer(nicole)', '=', 'True')
('*', 'TellTruth(john)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
----- END PART-----
----- START PART 3a-2: test implementation of statement 2 for 3a
FAIL: Your formula (And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))) says the following model is FALSE, but it should be TRUE:
Your formula (And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))) says the following model is FALSE, but it should be TRUE:
('*', 'CrashedServer(susan)', '=', 'True')
('*', 'TellTruth(nicole)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
----- END PART-----
----- START PART 3a-3: test implementation of statement 3 for 3a
You matched the 2 models
Example model: {'TellTruth(susan)'}
----- END PART-----
----- START PART 3a-4: test implementation of statement 4 for 3a
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
55°F Sunny
Search
2:47 PM 2/13/2024


File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
----- START PART 3a-4: test implementation of statement 4 for 3a
You matched the 4 models
Example model: {'TellTruth(susan)'}
----- END PART-----
----- START PART 3a-5: test implementation of statement 5 for 3a
You matched the 4 models
Example model: {'CrashedServer(susan)'}
----- END PART-----
----- START PART 3a-all: test implementation of all for 3a
FAIL: Your formula (And(And(And(And(And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTruth(mark))),And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))),And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))),And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole)))),Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y)))))))),Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y))))))) says the following model is FALSE, but it should be TRUE:
Your formula (And(And(And(And(And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTruth(mark))),And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))),And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))),And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole)))),Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y)))))))),Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y))))))) says the following model is FALSE, but it should be TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'TellTruth(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
----- END PART-----
----- START PART 3a-run: test implementation of run for 3a
>>>>> I learned something.
Query: TELL[And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTruth(mark))), standardized: ['And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTruth(mark)))']]
An example of a model where query is TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
55°F Sunny
Search
2:47 PM 2/13/2024

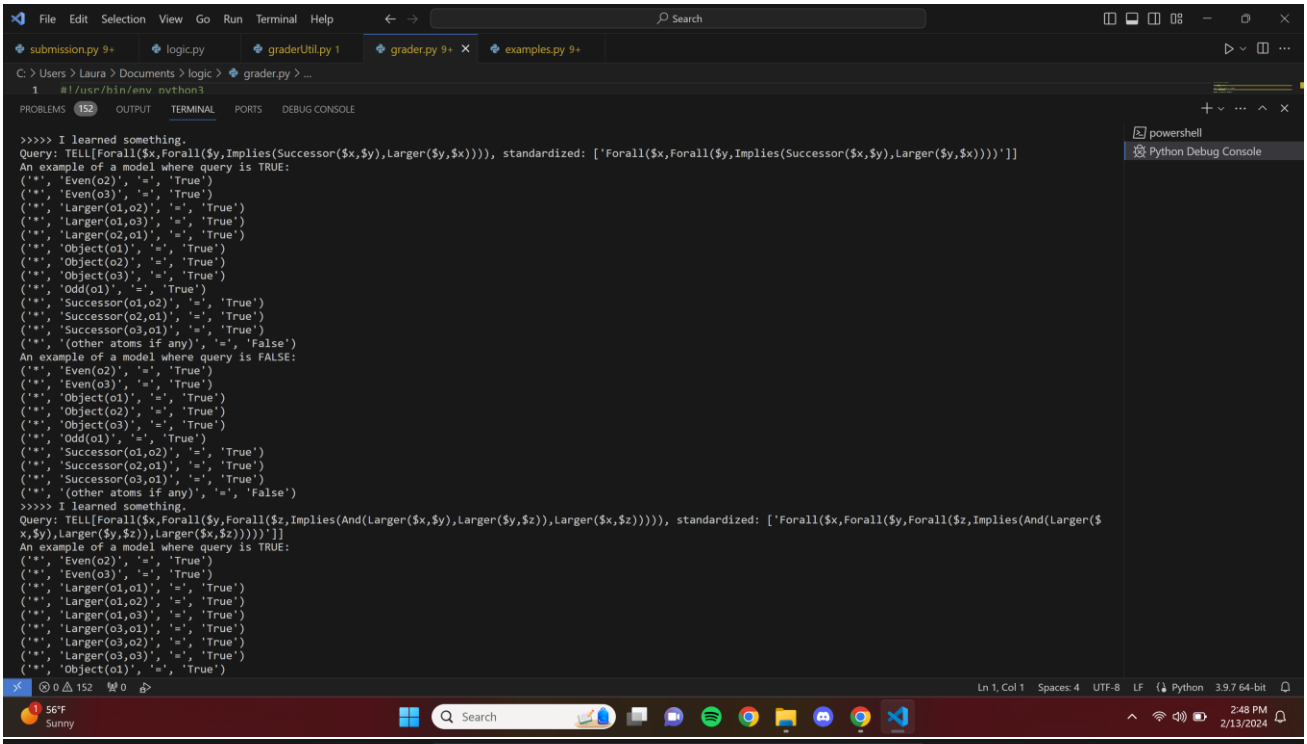

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan))), standardized: ['And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))']]
An example of a model where query is TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', 'TellTruth(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark))), standardized: ['And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))']]
An example of a model where query is TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'CrashedServer(susan)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole))), standardized: ['And(Implies(TellTruth(nicole),Not(Te
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
55°F Sunny
Search
2:47 PM 2/13/2024

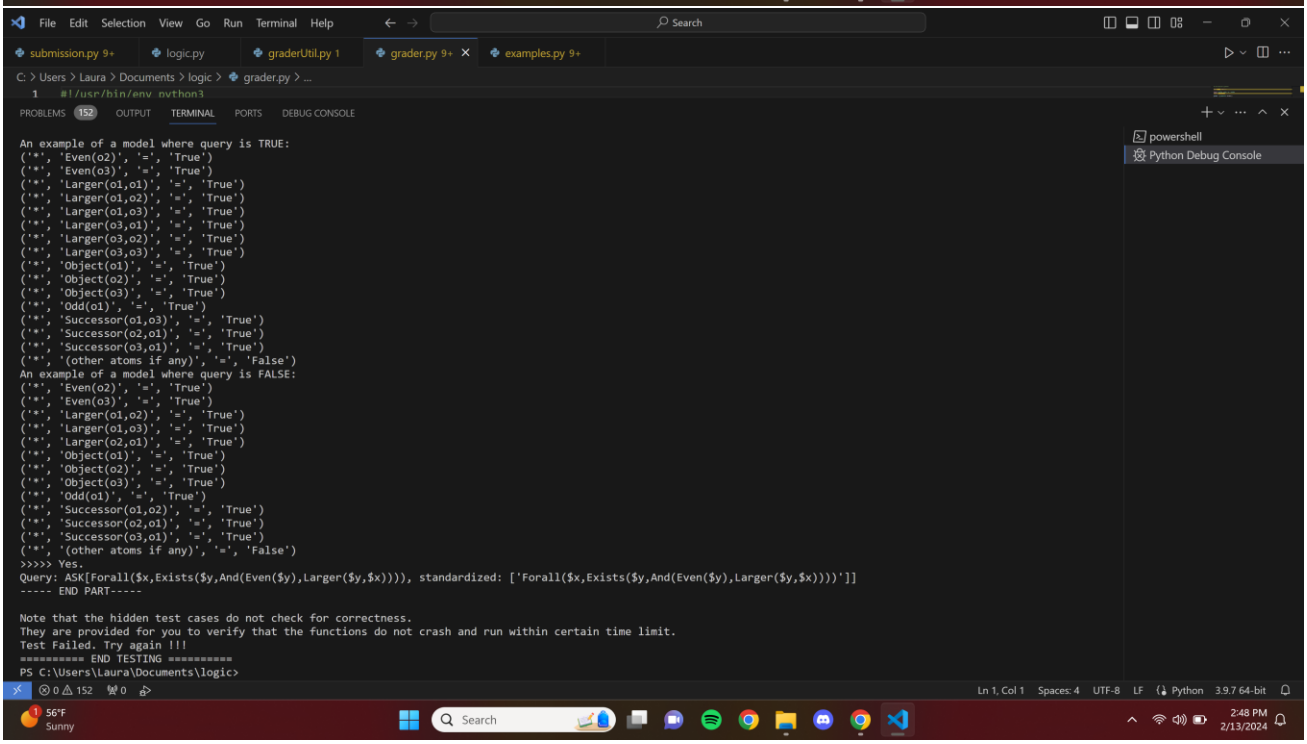

File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
>>>>> I learned something.
Query: TELL[And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole))), standardized: ['And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole)))']]
An example of a model where query is TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', 'TellTruth(nicole)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y)))))), standardized: ['Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y)))))']]
An example of a model where query is TRUE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', 'TellTruth(nicole)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'CrashedServer(mark)', '=', 'True')
('*', 'Object(john)', '=', 'True')
('*', 'Object(mark)', '=', 'True')
('*', 'Object(nicole)', '=', 'True')
('*', 'Object(susan)', '=', 'True')
('*', 'TellTruth(john)', '=', 'True')
('*', 'TellTruth(nicole)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y)))))), standardized: ['Exists($x,And(CrashedServer($x),Forall($y,I
Ln 1, Col 1 Spaces: 4 UTF-8 LF
55°F Sunny
Search


File Edit Selection View Go Run Terminal Help
Search
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
No: ['susan', 'john', 'nicole']
----- END PART-----
----- START PART 4a-0: test implementation of statement 0 for 4a
----- END PART-----
----- START PART 4a-1: test implementation of statement 1 for 4a
----- END PART-----
----- START PART 4a-2: test implementation of statement 2 for 4a
----- END PART-----
----- START PART 4a-3: test implementation of statement 3 for 4a
----- END PART-----
----- START PART 4a-4: test implementation of statement 4 for 4a
----- END PART-----
----- START PART 4a-5: test implementation of statement 5 for 4a
----- END PART-----
----- START PART 4a-all: test implementation of all for 4a
You matched the 36 models
Example model: {'Larger(o1,o3)', 'Larger(o2,o1)', 'Larger(o3,o1)', 'Successor(o1,o3)', 'Larger(o3,o3)', 'Larger(o2,o3)', 'Successor(o3,o1)', 'Larger(o2,o2)', 'Successor(o2,o1)', 'Odd(o2)', 'Odd(o3)', 'Even(o1)', 'Larger(o3,o2)', 'Larger(o1,o1)', 'Larger(o1,o2)'}
----- END PART-----
----- START PART 4a-run: test implementation of run for 4a
>>>>> I learned something.
Query: TELL[Forall($x,Exists($y,And(And(Successor($x,$y),Not(Equals($y,$x))),Forall($z,Implies(Not(Equals($y,$z)),Not(Successor($x,$z))))))), standardized: ['Forall($x,Exists($y,And(And(Successor($x,$y),Not(Equals($y,$x))),Forall($z,Implies(Not(Equals($y,$z)),Not(Successor($x,$z)))))))']]
An example of a model where query is TRUE:
('*', 'Object(o1)', '=', 'True')
('*', 'Object(o2)', '=', 'True')
('*', 'Object(o3)', '=', 'True')
('*', 'Successor(o1,o3)', '=', 'True')
('*', 'Successor(o2,o1)', '=', 'True')
('*', 'Successor(o3,o1)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
56°F Sunny
Search
2:48 PM 2/13/2024



File Edit Selection View Go Run Terminal Help
submission.py 9+
logic.py
graderUtil.py 1
grader.py 9+
examples.py 9+
C: > Users > Laura > Documents > logic > grader.py > ...
1 #!/usr/bin/env python3
PROBLEMS 152 OUTPUT TERMINAL PORTS DEBUG CONSOLE
powershell
Python Debug Console
An example of a model where query is TRUE:
('*', 'Even(o1)', '=', 'True')
('*', 'Even(o2)', '=', 'True')
('*', 'Object(o1)', '=', 'True')
('*', 'Object(o2)', '=', 'True')
('*', 'Object(o3)', '=', 'True')
('*', 'Odd(o3)', '=', 'True')
('*', 'Successor(o1,o3)', '=', 'True')
('*', 'Successor(o2,o3)', '=', 'True')
('*', 'Successor(o3,o1)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'Even(o1)', '=', 'True')
('*', 'Even(o2)', '=', 'True')
('*', 'Object(o1)', '=', 'True')
('*', 'Object(o2)', '=', 'True')
('*', 'Object(o3)', '=', 'True')
('*', 'Odd(o3)', '=', 'True')
('*', 'Successor(o1,o3)', '=', 'True')
('*', 'Successor(o2,o1)', '=', 'True')
('*', 'Successor(o3,o1)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
>>>>> I learned something.
Query: TELL[Forall($x,Forall($y,Implies(And(Odd($x),Successor($x,$y)),Even($y)))), standardized: ['Forall($x,Forall($y,Implies(And(Odd($x),Successor($x,$y)),Even($y))))
']]
An example of a model where query is TRUE:
('*', 'Even(o1)', '=', 'True')
('*', 'Even(o2)', '=', 'True')
('*', 'Object(o1)', '=', 'True')
('*', 'Object(o2)', '=', 'True')
('*', 'Object(o3)', '=', 'True')
('*', 'Odd(o3)', '=', 'True')
('*', 'Successor(o1,o3)', '=', 'True')
('*', 'Successor(o2,o3)', '=', 'True')
('*', 'Successor(o3,o1)', '=', 'True')
('*', '(other atoms if any)', '=', 'False')
An example of a model where query is FALSE:
('*', 'Even(o1)', '=', 'True')
('*', 'Object(o1)', '=', 'True')
('*', 'Object(o2)', '=', 'True')
Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.9.7 64-bit
56°F Sunny
Search
2:48 PM 2/13/2024





Here is the results if screenshots miss one:

PS C:\Users\Laura\Documents\logic> & 'C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python39_64\python.exe' 'c:\Users\Laura\.vscode\extensions\ms-python.python-

2024.0.1\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '54914' '--' 'c:\Users\Laura\Documents\logic\grader.py'

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'TellTruth(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

========== START TESTING ==========

----- START PART 1a: Test formula 1a implementation

You matched the 7 models

Example model: {'California', 'Rain'}

----- END PART-----

----- START PART 1b: Test formula 1b implementation

You matched the 4 models

Example model: {'Wet', 'Sprinklers'}

----- END PART-----

----- START PART 1c: Test formula 1c implementation

You matched the 2 models

Example model: {'Night'}

----- END PART-----

----- START PART 2a: Test formula 2a implementation

You matched the 343 models

Example model: {'Mother(o3,o2)', 'Mother(o2,o1)', 'Person(o2)', 'Person(o3)', 'Mother(o1,o3)', 'Person(o1)'}

----- END PART-----

----- START PART 2b: Test formula 2b implementation

You matched the 169 models

Example model: {'Child(o2,o1)', 'Child(o1,o3)', 'Child(o2,o3)', 'Person(o3)', 'Person(o1)'}

----- END PART-----

----- START PART 2c: Test formula 2c implementation

FAIL: Your formula
(Forall($x,Forall($y,And(Implies(Daughter($x,$y),And(Child($x,$y),Female($y))),Implies(And(Child($x,$y),
Female($y)),Daughter($x,$y)))))) says the following model is FALSE, but it should be TRUE:

Your formula
(Forall($x,Forall($y,And(Implies(Daughter($x,$y),And(Child($x,$y),Female($y))),Implies(And(Child($x,$y),
Female($y)),Daughter($x,$y)))))) says the following model is FALSE, but it should be TRUE:

('*', 'Parent(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

----- END PART-----

----- START PART 2d: Test formula 2d implementation

FAIL: Your formula
(Forall($x,Forall($z,And(Implies(Grandmother($x,$z),Exists($y,And(And(Female($z),Parent($x,$y)),Parent
($y,$z)))),Implies(Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z))),Grandmother($x,$z)))))
says the following model is FALSE, but it should be TRUE:

Your formula
(Forall($x,Forall($z,And(Implies(Grandmother($x,$z),Exists($y,And(And(Female($z),Parent($x,$y)),Parent
($y,$z)))),Implies(Exists($y,And(And(Female($z),Parent($x,$y)),Parent($y,$z))),Grandmother($x,$z)))))
says the following model is FALSE, but it should be TRUE:

('*', 'Child(o3,o2)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

----- END PART-----

----- START PART 3a-0: test implementation of statement 0 for 3a

You matched the 2 models

Example model: {'TellTruth(mark)'}

----- END PART-----

----- START PART 3a-1: test implementation of statement 1 for 3a

FAIL: Your formula (And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))) says the following model is FALSE, but it should be TRUE:

Your formula (And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))) says the following model is FALSE, but it should be TRUE:

('*', 'CrashedServer(nicole)', '=', 'True')

('*', 'TellTruth(john)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

----- END PART-----

----- START PART 3a-2: test implementation of statement 2 for 3a

FAIL: Your formula (And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))) says the following model is FALSE, but it should be TRUE:

Your formula (And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))) says the following model is FALSE, but it should be TRUE:

('*', 'CrashedServer(susan)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

----- END PART-----

----- START PART 3a-3: test implementation of statement 3 for 3a

You matched the 2 models

Example model: {'TellTruth(susan)'}

----- END PART-----

----- START PART 3a-4: test implementation of statement 4 for 3a

You matched the 4 models

Example model: {'TellTruth(susan)'}

----- END PART-----

----- START PART 3a-5: test implementation of statement 5 for 3a

You matched the 4 models

Example model: {'CrashedServer(susan)'}

----- END PART-----

----- START PART 3a-all: test implementation of all for 3a

FAIL: Your formula
(And(And(And(And(And(And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedSe
rver(mark)),TellTruth(mark))),And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServe
r(nicole),TellTruth(susan)))),And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(s
usan),TellTruth(mark)))),And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susa
n)),TellTruth(nicole)))),Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y))
)))))),Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y))))))))
says the following model is FALSE, but it should be TRUE:

Your formula
(And(And(And(And(And(And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedSe
rver(mark)),TellTruth(mark))),And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServe
r(nicole),TellTruth(susan)))),And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(s
usan),TellTruth(mark)))),And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susa
n)),TellTruth(nicole)))),Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y))
)))))),Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y))))))))
says the following model is FALSE, but it should be TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'TellTruth(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

----- END PART-----

----- START PART 3a-run: test implementation of run for 3a

>>>>> I learned something.

Query:
TELL[And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTrut
h(mark))), standardized:

['And(Implies(TellTruth(mark),Not(CrashedServer(mark))),Implies(Not(CrashedServer(mark)),TellTruth(mark)))']]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query:
TELL[And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan))), standardized:
['And(Implies(TellTruth(susan),CrashedServer(nicole)),Implies(CrashedServer(nicole),TellTruth(susan)))']]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query:
TELL[And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))
, standardized:
['And(Implies(TellTruth(mark),CrashedServer(susan)),Implies(CrashedServer(susan),TellTruth(mark)))']]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'CrashedServer(susan)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query:
TELL[And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole)
)), standardized:
['And(Implies(TellTruth(nicole),Not(TellTruth(susan))),Implies(Not(TellTruth(susan)),TellTruth(nicole)))']]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y))))),
standardized: ['Exists($x,And(TellTruth($x),Forall($y,Implies(Not(Equals($x,$y)),Not(TellTruth($y))))))']]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(john)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query:
TELL[Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y)))))),
standardized:
['Exists($x,And(CrashedServer($x),Forall($y,Implies(Not(Equals($x,$y)),Not(CrashedServer($y)))))')]

An example of a model where query is TRUE:

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'CrashedServer(john)', '=', 'True')

('*', 'CrashedServer(mark)', '=', 'True')

('*', 'Object(john)', '=', 'True')

('*', 'Object(mark)', '=', 'True')

('*', 'Object(nicole)', '=', 'True')

('*', 'Object(susan)', '=', 'True')

('*', 'TellTruth(nicole)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

Yes: ['mark']

Maybe: []

No: ['susan', 'john', 'nicole']

----- END PART-----

----- START PART 4a-0: test implementation of statement 0 for 4a

----- END PART-----

----- START PART 4a-1: test implementation of statement 1 for 4a

----- END PART-----

----- START PART 4a-2: test implementation of statement 2 for 4a

----- END PART-----

----- START PART 4a-3: test implementation of statement 3 for 4a

----- END PART-----

----- START PART 4a-4: test implementation of statement 4 for 4a

----- END PART-----

----- START PART 4a-5: test implementation of statement 5 for 4a

----- END PART-----

----- START PART 4a-all: test implementation of all for 4a

You matched the 36 models

Example model: {'Larger(o1,o3)', 'Larger(o2,o1)', 'Larger(o3,o1)', 'Successor(o1,o3)', 'Larger(o3,o3)', 'Larger(o2,o3)', 'Successor(o3,o1)', 'Larger(o2,o2)', 'Successor(o2,o1)', 'Odd(o2)', 'Odd(o3)', 'Even(o1)', 'Larger(o3,o2)', 'Larger(o1,o1)', 'Larger(o1,o2)'}

----- END PART-----

----- START PART 4a-run: test implementation of run for 4a

>>>>> I learned something.

Query:
TELL[Forall($x,Exists($y,And(And(Successor($x,$y),Not(Equals($y,$x))),Forall($z,Implies(Not(Equals($y,$z)),Not(Successor($x,$z)))))), standardized:
['Forall($x,Exists($y,And(And(Successor($x,$y),Not(Equals($y,$x))),Forall($z,Implies(Not(Equals($y,$z)),Not(Successor($x,$z)))))')]]

An example of a model where query is TRUE:

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Forall($x,Or(And(Even($x),Not(Odd($x))),And(Odd($x),Not(Even($x))))), standardized:
['Forall($x,Or(And(Even($x),Not(Odd($x))),And(Odd($x),Not(Even($x))))')]]

An example of a model where query is TRUE:

('*', 'Even(o1)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o2)', '=', 'True')

('*', 'Odd(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Forall($x,Forall($y,Implies(And(Even($x),Successor($x,$y)),Odd($y)))), standardized: ['Forall($x,Forall($y,Implies(And(Even($x),Successor($x,$y)),Odd($y))))']]

An example of a model where query is TRUE:

('*', 'Even(o1)', '=', 'True')

('*', 'Even(o2)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o3)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Even(o1)', '=', 'True')

('*', 'Even(o2)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Forall($x,Forall($y,Implies(And(Odd($x),Successor($x,$y)),Even($y)))), standardized: ['Forall($x,Forall($y,Implies(And(Odd($x),Successor($x,$y)),Even($y))))']]

An example of a model where query is TRUE:

('*', 'Even(o1)', '=', 'True')

('*', 'Even(o2)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o3)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Even(o1)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o2)', '=', 'True')

('*', 'Odd(o3)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o2)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Forall($x,Forall($y,Implies(Successor($x,$y),Larger($y,$x)))), standardized: ['Forall($x,Forall($y,Implies(Successor($x,$y),Larger($y,$x))))']]

An example of a model where query is TRUE:

('*', 'Even(o2)', '=', 'True')

('*', 'Even(o3)', '=', 'True')

('*', 'Larger(o1,o2)', '=', 'True')

('*', 'Larger(o1,o3)', '=', 'True')

('*', 'Larger(o2,o1)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o1)', '=', 'True')

('*', 'Successor(o1,o2)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Even(o2)', '=', 'True')

('*', 'Even(o3)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o1)', '=', 'True')

('*', 'Successor(o1,o2)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> I learned something.

Query: TELL[Forall($x,Forall($y,Forall($z,Implies(And(Larger($x,$y),Larger($y,$z)),Larger($x,$z))))), standardized: ['Forall($x,Forall($y,Forall($z,Implies(And(Larger($x,$y),Larger($y,$z)),Larger($x,$z)))))']]

An example of a model where query is TRUE:

('*', 'Even(o2)', '=', 'True')

('*', 'Even(o3)', '=', 'True')

('*', 'Larger(o1,o1)', '=', 'True')

('*', 'Larger(o1,o2)', '=', 'True')

('*', 'Larger(o1,o3)', '=', 'True')

('*', 'Larger(o3,o1)', '=', 'True')

('*', 'Larger(o3,o2)', '=', 'True')

('*', 'Larger(o3,o3)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o1)', '=', 'True')

('*', 'Successor(o1,o3)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

An example of a model where query is FALSE:

('*', 'Even(o2)', '=', 'True')

('*', 'Even(o3)', '=', 'True')

('*', 'Larger(o1,o2)', '=', 'True')

('*', 'Larger(o1,o3)', '=', 'True')

('*', 'Larger(o2,o1)', '=', 'True')

('*', 'Object(o1)', '=', 'True')

('*', 'Object(o2)', '=', 'True')

('*', 'Object(o3)', '=', 'True')

('*', 'Odd(o1)', '=', 'True')

('*', 'Successor(o1,o2)', '=', 'True')

('*', 'Successor(o2,o1)', '=', 'True')

('*', 'Successor(o3,o1)', '=', 'True')

('*', '(other atoms if any)', '=', 'False')

>>>>> Yes.

Query: ASK[Forall($x,Exists($y,And(Even($y),Larger($y,$x)))), standardized: ['Forall($x,Exists($y,And(Even($y),Larger($y,$x))))']]

----- END PART-----