

Checking syntatic correctness and returning false truth values of a formula - Python

Logic and set theory

Mikolaj Aleksandrak, Michal Harenda, Angelika Lach, Konrad Skrzypczak

21.06.2018

Program that checks if the formula in classical logic is syntatically correct, checks the truth of any given correct formula and returns the possible evaluation of variables for which formula is false.

Program that checks if the formula in classical logic is syntatically correct, checks the truth of any given correct formula and returns the possible evaluation of variables for which formula is false.

Program that checks if the formula in classical logic is syntatically correct, checks the truth of any given correct formula and returns the possible evaluation of variables for which formula is false.

- Truth values in classical logic

Program that checks if the formula in classical logic is syntatically correct, checks the truth of any given correct formula and returns the possible evaluation of variables for which formula is false.

- Truth values in classical logic
- Syntatic correctness

Definitions

- Truth values in classical logic
 - each statement has a truth value of either “true” or “false”, but not both

Definitions

- Truth values in classical logic
 - each statement has a truth value of either “true” or “false”, but not both
- Syntactic correctness
 - Polish notation
 - Negation: N
 - Alternative: A
 - Conjunction: K
 - Implication: C
 - Equivalence: E
 - **Each operator with exception of N is followed by two variables; N precedes only one.**
 - Variables are represented by p^* , where $*$ is any positive integer starting from 1

Pseudocode

Get input and pass it to Syntax()

Syntax():

 If no forbidden symbols:

 If formula is a single variable:

 Print results for single variable

 Else if formula is valid:

 Pass formula to Manipulator()

Manipulator():

 Extract unique variables from formula

 For 2 to the power of amount of variables:

 Count in binary for variable values

 Pass values to Solver()

 Pass all solved to Printer()

Pseudocode

Solver():

 Replace variables by assigned values

 Until formula is not solved:

 Replace subformulas with their values

 Return formula

Printer():

 If there is no false values in formula:

 Print "No occurrences where formula is false"

 Else:

 Print variable values where formula is false

Program examples - non-valid inputs

```
## Ep1
```

```
## Formula not valid
```

```
## -----
```

```
## p1p2
```

```
## Formula not valid
```

```
## -----
```

```
## q1
```

```
## Non-valid characters used
```

```
## -----
```

Program examples - valid inputs

```
## p1
## False for:
## p1
## 0
## -----
## Ep1p1
## No occurrences where formula is false
## -----
## Ep1p2
## False for:
## p1 p2
## 1  0
## 0  1
## -----
```