

HW2 Code Generation for Propositional Logic

Deadline: March 18, 9 AM

This homework is based on the homework 1.

Given a Boolean formula F , the Boolean satisfiability problem asks if there are values for the variables in F that makes the F to be True. For example, in $G = P \wedge (Q \vee !P)$, G evaluates to True for $P = \text{True}$ and $Q = \text{True}$. We call G satisfiable and $(P = \text{True}, Q = \text{True})$ a satisfying assignment for G .

In this homework, you are asked to use the parser in HW1 and pysmt (<https://github.com/pysmt/pysmt>) to (1) generate python code that represent the constraints represented by propositions to an equivalents logical formula using pysmt APIs, (2) run the python code in the previous step to check the satisfiability of the formula.

Your program will take a file containing a set of propositions and prints the result of the satisfiability of conjunction of the propositions.

Example 1:

Input:

$P \wedge !Q$

Corresponding python code:

```
from pysmt.shortcuts import Symbol, And, Not, is_sat

P = Symbol("P") # Default type is Boolean
Q = Symbol("Q")
f = And(P, Not(Q))
print is_sat(f)
```

\$ python main.py constraints.txt

True

Example 2:

Input:

$P \wedge !Q, !P \Leftrightarrow !Q$

python code:

```
from pysmt.shortcuts import Symbol, And, Iff, Not, is_sat

P = Symbol("P") # Default type is Boolean
Q = Symbol("Q")
prop1 = And(P, Not(Q))
prop2 = Iff(Not(P), Not(Q))
```

```
f = And(prop1, prop2)
print is_sat(f)
```

```
$ python main.py constraints.txt
True
```