

I went by order or requirements to create the program. With the initial tasks `findVarIds` solved AFTER realising I needed to have parentheses around the parameters (I didn't think about function type at the time). Help for `genVarAsgns` with hints saying to use `Map.insert` and `Map.empty` was nice and was able to be implemented relatively easier afterwards.

I found most difficult was the type that was required for `eval`:

I was stuck for the longest time on how to implement because I didn't realise it was a recursive function, for example I wrote:

`eval (And p q) asgn = p && q` - - silly I know...

Instead of what I have now:

`eval (And p q) asgn = (eval p asgn) && (eval q asgn)`

`sat :: Prop -> Bool` had me finally piece together how I was going to use all previous functions together.

I felt as though 5 and 6 could just combined as a single requirement,
doing

`checkFormula s = if sat (read s) then "SAT" else "UNSAT"`

Would do the same thing but I used `readFormula` instead in case i lose marks for not making the function

For main I had to google how to pass `formulas.txt` as an argument for the program:

<https://stackoverflow.com/questions/25537861/how-to-pass-arguments-to-functions-in-haskell>

***PERSONAL THOUGHTS/REFLECTIONS**

In the future I feel as though implementing it from main and going backwards might have made things more simple to implement, I didn't initially have a clear direction of how to use the what i made in section 1 in section 2, but it wasn't until section 5 and onwards where I could see how it all came together. I can see how going step by step is easier per function, but big picture wise It serves to have a better understanding if the order was reversed.

***TESTS USED:**

Base tests from formulas.txt

Const True

Iff (Var "x") (Var "y")

And (Var "x1") (Not (Var "x2"))

Not (Imply (Var "p") (Imply (Var "q") (Var "p")))

In addition to:

Const False

Var "z"

Not (Var "z")

Or (Var "y") (Not (Var "x"))

And (Var "x") (Not (Var "x"))

Imply (Var "x") (Var "x")

Iff (Var "x") (Var "x")

And (Or (Var "x") (Var "y")) (Not (Var "x"))

Or (And (Var "x") (Var "y")) (And (Not (Var "x")) (Not (Var "y"))))

Imply (And (Var "x") (Var "y")) (Var "x")

Not (Iff (Var "x") (Var "y"))

Imply (Var "x") (Or (Var "x") (Var "y"))

^ These additional tests were added throughout working on the project. Although a lot of the other tests I've done were a one time thing and weren't included in the edited formulas.txt file and tried different tests so my formulas.txt doesn't have a final file that contains ALL of the tests