eval: for this function I was not sure how to look for varId in varAsgn so I used Map.lookup and if it exists it returns varId and if it does not then error. I was not able to get any error results so I think it will always be the first line. I kept the error there for some off case that I cannot think of.

findVarsIds: I wanted to do something along the lines of (_ p1 p2) so I can just do it in one line but it would give me errors. So I went with the repetition method of doing all the outcomes.

gemVarAsgms: I tried to use zip replicate map and sequence to do it. I really didn't understand the logic behind it but someone in the discord server mentioned I could do it that way to make it one line. I ended up trying every combination of those 4 and got it working. Didn't understand how it worked so I got rid of it and stuck with what I have because it makes sense to me. Would be nice to get a full clarification on how this method works.

```haskell
genVarAsgns :: [VarId] -> [VarAsgn]
genVarAsgns vars = map (Map.fromList . zip vars) (sequence (replicate (length vars) [True, False]))
```

Overall I tried to keep the program short as possible. I think when the program is concise and straight to the point it is easier for me to understand. I implemented the features that were mentioned and I could not think of anymore features to add.

test cases I used:
Const True

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

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

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

Var "x"

Const False

Not (Var "x")

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

Imply (Var "p") (Var "q")

Iff (Var "p") (Var "q")

And (Or (Var "p") (Var "q")) (Not (Var "r"))

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

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

And (Or (Var "x1") (Var "x2")) (Or (Var "x3") (Var "x4"))

Const True

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

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

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

Var "x"

Const False

Not (Var "x")

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

Imply (Var "p") (Var "q")

Iff (Var "p") (Var "q")

And (Or (Var "p") (Var "q")) (Not (Var "r"))

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

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

And (Or (Var "x1") (Var "x2")) (Or (Var "x3") (Var "x4"))