**First Order Predicate Logic:**

* Predicate symbols are operators that return either true or false
  + 2 = 3 is false, therefore = is a predicate symbol
  + A symbol must be a predicate symbol if it is operated on by a propositional logic operator (¬ ∧ ∨ →) because it must evaluate to either true or false
* Function symbols are operators that return a value
  + 4 + 1 is 5, therefore +, 4, and 1 are function symbols
  + Numbers are function symbols with arity 0, as well
* Terms: variables and function symbols with only terms in them
  + Ex: f(x): x is term as well as f(x)
* A function is set as closed only if all the variables are bound to a quantifier

Universal quantifier Ɐx reads as “for all assignments of x”

Existential quantifier ∃x reads as “for some assignment of x” or “for at least one x”

**“And” vs. “if-then”:**

* In general, if a statement is prefaced by a universal quantifier, then implication is used
* In general, if a statement is prefaced by an existential quantifier, then conjunction is used

|  |
| --- |
| Character toolbox: ㄱ⋀⋁≡∀∃∈∉𝓐𝓑 |

**Proving the equivalence ( ≡ ) or non-equivalence ( ≢ ) of formulas:**

One strategy for proving F ≡ G without using statements in Theorem 3.4.1:

Left side:

𝓐(F) = 1

if … then …

If … then 𝓐(G) = 1

Right side:

𝓐(G) = 1

If … then …

If … then 𝓐(F) = 1

One strategy for proving F ≢ G:

First , define a structure that would hold for one formula but not the other. Then do the following for the left and right sides:

Left side:

𝓐(F) = 1

if … then …

If … then …

𝓐 is a model for F

Right side:

𝓐(G) = 1

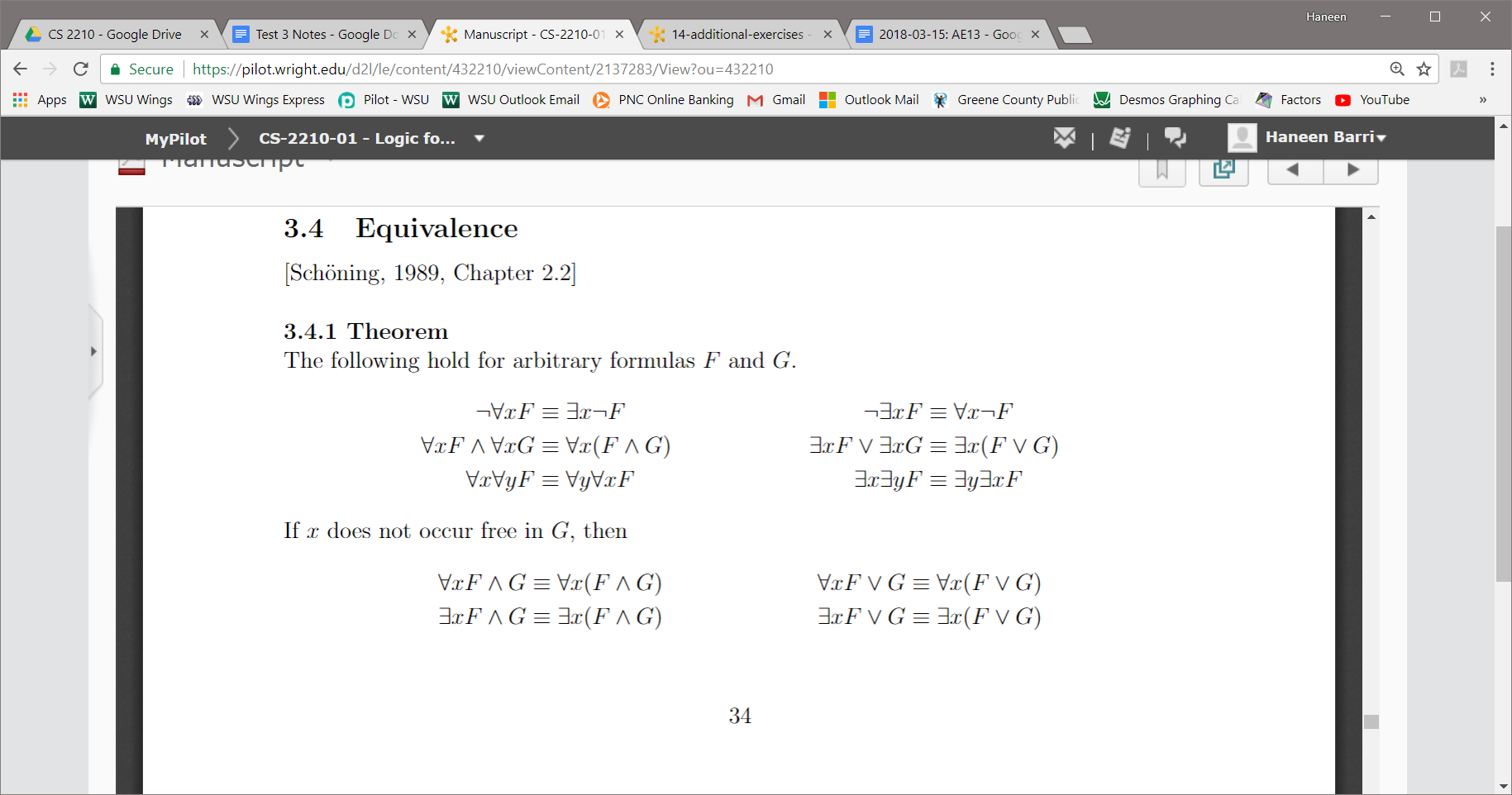
if … then …

If … then xyz

But since xyz does not hold, 𝓐 is not a model for G.

Since 𝓐 is a model for one of the formulas but not the other, they are not equivalent.

**Equivalences:**

****

**Tableaus:**

∃ : Whenever you resolve an existential you must add in a new variable in the tableaus.

In general, you should resolve existentials before universals, and conjunctions before disjunctions.

Questions for Q&A session Exam 3 | 4/19

* When creating structures

Example: P(x) ⋀ Q(x,y)

For a structure and not a model go with null sets its the easiest

For Predicate and Function Symbols

Predicate: you know that the result is true or false

Function: you give an input and get a different output as a result

* Numbers (e.g. 0) can be classified as function symbols with arity zero

Before we determine subformulas, should we convert the formula to NNF?

* Convert to NNF just in case

Problems when checking when the subformulas are closed and open:

The starting formula given at the beginning is closed right from the start

As you go through each subformula when you have removed all the ∀, ∃ the rest of the function is open.

Different in Term, function symbol:

Terms: are only variables, or a function as defined below:

Functions that are terms: With terms of only variables, or other functions holding only terms

Model Equivalence: