Formal Proof of Truth

- Given a set of proof rules, two propositions are provably equivalent if each follows from the other according to those rules for logical equivalence.
- Here is a step-by-step reasoning, it is an example of formal proof of truth.

1. Show that $(p \to q) \wedge p \to q \Leftrightarrow T$

$(p \to q) \wedge p \to q$	$\Leftrightarrow (\neg p \vee q) \wedge p \to q$	Definition of →
	$\Leftrightarrow (\neg p \wedge p) \vee (q \wedge p) \to q$	Distributivity of ∧
	$\Leftrightarrow F \vee (q \wedge p) \to q$	Contradiction
	$\Leftrightarrow (q \wedge p) \to q$	Identity
	$\Leftrightarrow \neg(q \wedge p) \vee q$	Definition of →
	$\Leftrightarrow (\neg q \vee \neg p) \vee q$	DeMorgan's law
	$\Leftrightarrow \neg p \vee (\neg q \vee q)$	Commutativity and Associativity of ∨
	$\Leftrightarrow \neg p \vee T$	Excluded middle
	$\Leftrightarrow T$	Domination

- Some comments of the above proof:
  - In this proof, we start with the lefthand side (**lhs**) of the logic equality then go all the way to the righthand side (**rhs**). This is one way to prove. People can also prove equality from the rhs to the lhs; or prove equality by starting from both sides then reaching the same expression.
  - We should try to avoid "unpleasant" steps, for example: technically, the excluded middle rule says "$p \vee \neg p \Leftrightarrow T$", but we used "$\neg q \vee q \Leftrightarrow T$" in the proof directly.

- We will see the proof of logic implication a lot as well, because we don't always have or need a logic equality in reasoning. While proving $p \Rightarrow q$, we can only start from the lhs then try to reach the rhs.

2. Show that $(p \vee q) \wedge (p \to r) \wedge (q \to r) \Rightarrow r$ without using or-elimination immediately.

$(p \vee q) \wedge (p \to r) \wedge (q \to r)$	
$\Leftrightarrow (p \vee q) \wedge \big((p \to r) \wedge (q \to r)\big)$	Associativity of ∧
$\Leftrightarrow \Big(p \wedge \big((p \to r) \wedge (q \to r)\big)\Big) \vee \Big(q \wedge \big((p \to r) \wedge (q \to r)\big)\Big)$	Distributivity of ∧
$\Leftrightarrow \Big(\big(p \wedge (p \to r)\big) \wedge (q \to r)\Big) \vee \Big(q \wedge \big((p \to r) \wedge (q \to r)\big)\Big)$	Associativity of ∧
$\Rightarrow \big(r \wedge (q \to r)\big) \vee \Big(q \wedge \big((p \to r) \wedge (q \to r)\big)\Big)$	Modus Ponens
$\Leftrightarrow \big(r \wedge (q \to r)\big) \vee \Big(\big(q \wedge (q \to r)\big) \wedge (p \to r)\Big)$	Associativity & Commutativity of ∧
$\Rightarrow \big(r \wedge (q \to r)\big) \vee \big(r \wedge (p \to r)\big)$	Modus Ponens
$\Rightarrow r \vee r$	or-introduction
$\Leftrightarrow r$	Idempotency

Review of Predicate Logic

- A **predicate** is a logical assertion (it can be T or F) that describes some property of values, such as $8 < 2.45$, $x = 2$ or $x^2 \geq x$ where $x \in \mathbb{R}$.
- In predicate logic, we assert truths about values from one or more "**domains of discourse**" (sets of values) like integers, real numbers …
  - For example, in the predicate $8 < 2.45$, we can consider we have values from two different domains: integers and real numbers. In predicate $x = 2$, we can consider that the domain of x is the set of all integers.
- Note that the domain of variables can affect the truth value of a predicate.
  - The predicate $x^2 \geq x$ where $x \in \mathbb{R}$ involves the domain "real numbers", and it is not always $True$; if we have the same expression but we use a different domain "$x \in \mathbb{Z}$", then predicate is always $True$.

In this class, the default domain of a domain variable is considered as $\mathbb{Z}$.

- In predicate logic, we can have operators that are used to calculate values like: $+,\ -,\ *,\ /\ (\div),\ \%$ … and operators that describe relations like: $>,\ \geq,\ =,\ \leq$, and $<$. Use your math knowledge to evaluate a predicate.
- **Hierarchy:** When there are logical operators involved, we treat each "math" part as if they're wrapped in parentheses.

3. True or False:
   a. $8 > 2 \wedge 8 \leq 10$ $True$
   b. $x > 1 \rightarrow x^2 > x$ $True$
   c. $x = 1 \vee x = -1 \rightarrow x^2 = 1$ $True$

- Same as propositional logic, a **state in predicate logic** is also a collection of bindings, but now in each binding, a domain variable can bind with a domain value.

- We have seen **well-formedness** and **properness** of states in propositional expressions. States for predicates follow the same concepts, the only difference being the kinds of values that variables map to: In addition to Boolean values, there are also values from whatever domain our predicates range over (such as $\mathbb{Z}$).

4. True or False:
   a. $\{x = y, y = 0\}$, $\{x = 5, x = 6, y = 2\}$ are well-formed states. $False$
   b. State $\sigma = \{x = 8, y = 2\}$ is proper for $x + y + z > 0$. $False$
   c. State $\sigma = \{x = 8, y = 0\}$ is proper for $\frac{x}{y} > 0$.
      $True$, but it will give a runtime error while evaluating and we will discuss runtime error in future classes.
   d. State $\sigma = \{b = 2\}$ is proper for $b[3] = 4$.
      $False$, because in the predicate b is an array, but in the state b is an integer.

- A state σ **satisfies** a predicate p if $\sigma(p) = T$, which is denoted as $\sigma \vDash p$. Similarly, if $\sigma(p) = F$, then $\sigma \nvDash p$ (σ doesn't satisfy p).

5. True or False:
   a. Let $\sigma = \{x = 1,\ y = 3\}$, then $\sigma \vDash x + y \geq 4$. $True$
   b. Let $\sigma = \{b = (3, 5, 9),\ x = 2\}$, then $\sigma \vDash b[x] = 9$. $True$
   c. $\{x = 1,\ y = 2\} \nvDash (x^2 + 2y)\ \%\ 5 = 1$ $True$

Quantified Predicate

- When a predicate includes a variable (such as $x^2 \geq x$ for the domain $x \in \mathbb{R}$), we can use **quantifiers** to specify the values for variables we consider (so a quantifier usually can also help us specify the domain of a variable).

- Types of quantifiers:
  - A **universal quantifier** has a form "$\forall x \in S$" where S is a set of values. A **universally quantified predicate** (or just "**universal**" for short) has the form $\forall x \in S.p$ where S is a set and p (the **body** of the universal) is a predicate involving x.
    - For example: $\forall x \in \mathbb{Z}. x > 1 \rightarrow x < x^2$ means "for all integer x, if it is greater than 1 then it is smaller than its square". Here $\forall x \in \mathbb{Z}$ is the quantifier, and $x > 1 \rightarrow x < x^2$ is the body.
    - We usually leave out the set if it is understood. For example, when we are clear from the context that $x \in \mathbb{Z}$ then we can simplify the above predicate to $\forall x. x > 1 \rightarrow x < x^2$.

  - An **existential quantifier** has a form "$\exists x \in S$" where S is a set of values. An **existentially quantified predicate** (or just "**existential**" for short) has the form $\exists\, x \in S.p$ where S is a set and p (the body of the existential) is a predicate involving x.
    - For example, $\exists x \in \mathbb{Z}. x \neq 0 \wedge x = x^2$ means "there exists integer x such that it is not equal to 0 but it is equal to its square".

- We can use **bounded quantifier** to abbreviate a quantifier with a condition in the body by moving the condition to the quantifier. More specifically, $\forall p. q$ means $\forall x. p \rightarrow q$ where x appears in p and x is understood to be the variable we are quantifying over; $\exists p. q$ means $\exists x. p \wedge q$ where x appears in p and x is understood to be the variable we are quantifying over.
  - For example: $(\forall x > 1. x < x^2) \equiv (\forall x. x > 1 \rightarrow x < x^2)$
  - For example: $(\exists x > 0. x = x^2) \equiv (\exists x. x > 0 \wedge x = x^2\ )$

- The body of a quantified predicate is always as long as possible.

6. Show full parenthesization for the following quantified predicates.
   a. $\forall x \in \mathbb{Z}. x > 1 \rightarrow x < x^2 \equiv \Big(\forall x \in \mathbb{Z}. \big((x > 1) \rightarrow (x < x^2)\big)\Big)$
   b. $\forall x \in \mathbb{Z}. \exists\, y \in \mathbb{Z}. y \leq x^2 \equiv \Big(\forall x \in \mathbb{Z}. \big(\exists\, y \in \mathbb{Z}. (y \leq x^2)\big)\Big)$
   c. $(\exists\, y \in \mathbb{Z}. y > 0 \wedge b[x+1] > y) \rightarrow x \geq 1 \equiv \Big(\Big(\exists\, y \in \mathbb{Z}. \big((y > 0) \wedge (b[x+1] > y)\big)\Big) \rightarrow (x \geq 1)\Big)$

   - Note that, for full parenthesization, we add parentheses around basic tests, but we still omit them around single variables and single constants, and we also omit those around array indexes.

- We also have proof of truth in quantified predicate logic. Most of the proof rules are like in propositional logic. We will look at something only in predicate logic. First, let's look at the DeMorgan's law in predicate logic:
  - **DeMorgan's Law:** $(\neg \forall x\ .p) \Leftrightarrow (\exists\, x. \neg\ p)$ "Not all apples are red" means "There exist non-red apples"
    $(\neg \exists\ x\ .p) \Leftrightarrow (\forall x\ . \neg p)$ "There exist no white crows" means "All crows are not white."

7. Simplify the following predicate to remove negation operators.

$\neg\exists x\,.\,x \leq 0 \wedge x > 0 \Leftrightarrow \forall x\,.\,\neg(x \leq 0 \wedge x > 0)$ DeMorgan's Law in predicate logic

$\Leftrightarrow \forall x\,.\,\neg(x \leq 0) \vee \neg(x > 0)$ DeMorgan's Law in propositional logic

$\Leftrightarrow \forall x\,.\,x > 0 \vee x \leq 0$

- In general, it is easy to prove "(It is true that) $\exists x.p$", what we need to do is to find a **witness value** for x that satisfies p.

8. Prove $\exists x \in \mathbb{Z}.x \neq 0 \wedge x \leq x^2$.
   - When $x = 1$ we have $x \neq 0 \wedge x \leq x^2$, so $\exists x \in \mathbb{Z}.x \neq 0 \wedge x \leq x^2$.

- In general, it is hard to prove "(It is true that) $\forall x.p$", it usually needs knowledge in other areas.

9. How to prove $\forall x \in \mathbb{Z}.x \neq 0 \rightarrow x \leq x^2$?
   - You can prove by cases, and you need to show the statement holds when $x < 0, x = 1, x > 1$, respectively.
   - Or you can prove by contradiction, you start with assuming the statement being $False$, then you will have its negation being true ($\exists x \in \mathbb{Z}.x \neq 0 \wedge x > x^2$), then you try to come up with a contradiction from there.

Predicate functions

- We like to give names to predicates and parameterize them, in other words, we like to write predicates into functions from a set of variables to a Boolean value. These functions are called predicate functions.
  - One reason for using predicate functions is that it can simplify our program, so it is easier to read.

  - Another reason is that predicate functions help us to handle expressions with expanded domains of variables.
    - For example, if we define $L(x, y) \equiv$ "x loves y", where the domain of x and y are all people, then we can express "Everyone loves someone other than themselves" as $\forall x.\exists y.L(x, y) \wedge x \neq y$.

10. Define predicate functions as required.
    a. Define function $isSorted(b, m, n)$ so that it returns $Ture$ if and only if subarray b between index m to n (including m and n) is sorted. You may assume that m and n are both valid index in array b and $n > m$. For example, $isSorted\big((1, 3, 5, 4), 0, 2\big) = True$.

$$isSorted(b, m, n) \equiv \forall i.m \leq i \leq n - 1 \rightarrow b[i] \leq b[i + 1]$$

    b. Define function $isZero(b, n)$ so that it returns $True$ if and only if the first n numbers in array b are all 0. You may assume that n is no larger than the size of the array b.
       - One attempt is to write the function as $isZero(b, n) \equiv b[0] = 0 \wedge b[1] = 0 \wedge b[2] = 0 \wedge \ldots \wedge b[n - 1] = 0$. This is not correct, since "…" is not understandable. Instead, we should write:

$$isZero(b, n) \equiv \forall 0 \leq i < n.b[i] = 0$$