CS70 Note 1: Mathematical Foundations

Lucas Zheng

1 Propositions

A statement that's either true or false is called a **proposition**.
Ex: $\sqrt{3}$ is rational, $1 + 1 = 5$, CS70 is cool
Propositional variables P, Q, R represent arbitrary propositions.
Connectives oin propositions together to form more complex ones.

- **And** $P \land Q$
- **Or** $P \lor Q$
- **Not** $\neg P$

Ex: $\sqrt{3}$ is rational $\land$ $1 + 1 = 5$
Propositional formulas are created by combining propositional variables with connectives.
Ex: $P \land Q \lor \neg R$

2 Propositional Logic

Tautology is a propositional formula that is always true regardless of the truth values of the variables.
Contradiction always false regardless of the truth values of the variables (tautologically false).
Truth table is an algorithm to verify if a propositional formula is a tautology.

P	P	$\land$	$\neg P$
T	T	F	F
F	T	T	T

Implication $P \implies Q$ means "if P, then Q". Only false if P is T and Q is F. Equivalent to $\neg P \lor Q$.
2 propositional formulas are **tautologically equivalent** if they have the

same truth table, written as $P \equiv Q$
Material equivalence $P \iff Q$ means "P if and only if Q". Only true if P and Q are the same.
Tautological equivalence allows us to transform difficult propositional formulas into equivalent easier ones.
Contrapositive $\neg Q \implies \neg P$ (tautologically equivalent to $P \implies Q$)
Converse $Q \implies P$
De Morgan's Laws $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ and $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$

3 First Order Logic

Predicate is a function which takes as input some element from a domain and outputs a proposition.
Ex: The predicate statement $x^2 + 3x = 0$ truth depends on the value of x.
First-order sentence is a proposition which uses quantifiers.
Quantifier quantifies the variable predicate statements use, to turn them into valid propositions.

- **Existence** $\exists x P(x) \sim \bigwedge_{i=1}^{\infty} P(a_i)$
- **Universal** $\forall x P(x) \sim \bigvee_{i=1}^{\infty} P(a_i)$

First-order formulas are created by combining predicate variables and quantifiers. All variables must be bound by a single quantifier to be valid.
Predicate variables can represent any predicate, so determining its truth value is hard. We have to consider all possible predicates (models).
Model there are infinitely many per predicate, consists of

- **Domain** a non-empty set of objects and rules that govern them.
- An interpretation for all predicates in the first-order formulas.

Logically true if every model makes P true.
Logical implication $P \implies Q$ if any model that makes P true makes Q true.
Logical equivalence $P \iff Q$ if every model that makes one true makes the other true.
Quantifier rules allow manipulation.

- Universal Instantiation (UI): $\forall x P(x) \implies P(c)$ over any c.
- Existential Generalization (EG): $P(b) \implies \exists x P(x)$.
- Universal Generalization (UG): $P(c)$ for c arbitrary $\implies \forall x P(x)$.
- Existential Instantiation (EI): $\exists x P(x) \implies P(b)$ for some b.

Restrict quantifiers to a certain set of models.
Ex: $\exists x(x \in \mathbb{R} \wedge x^2 + 3x = 0)$. We short hand this as: $(\exists x \in \mathbb{R})(x^2 + 3x = 0)$.
Negate quantifiers with De Morgan's law.

$$\neg \forall x P(x) \iff \exists x \neg P(x) \text{ and } \neg \exists x P(x) \iff \forall x \neg P(x)$$

Numerical quantification to get a number between "exists" and "for all".
Ex: "At least 2 objects satisfying P"

$$\exists x \exists y (x \neq y \wedge P(x) \wedge P(y))$$