Formal Logic

A Pragmatic Introduction
(Based on [Gallier 1986] and [Huth and Ryan 2004])

Yih-Kuen Tsay

Department of Information Management
National Taiwan University

Prelude: Ambiguity in Natural Language

- Two signs at the foot of a public escalator:
  - Shoes must be worn
  - Dogs must be carried
- What do they mean?

Prelude: Ambiguity in Natural Language

- Two signs at the foot of a public escalator:
  - Shoes must be worn
  - Dogs must be carried
- What do they mean?
- In formal logic, "Shoes must be worn" may be expressed as:
  $\forall x(OnEscalator(x) \rightarrow \exists y(PairOfShoes(y) \wedge IsWearing(x, y)))$.

Prelude: Ambiguity in Natural Language

- Two signs at the foot of a public escalator:
  - Shoes must be worn
  - Dogs must be carried
- What do they mean?
- In formal logic, "Shoes must be worn" may be expressed as:
  $\forall x(\mathit{OnEscalator}(x) \rightarrow \exists y(\mathit{PairOfShoes}(y) \wedge \mathit{IsWearing}(x, y)))$.
- And, "Dogs must be carried" as:
  $\forall x(\mathit{OnEscalator}(x) \wedge \mathit{IsDog}(x) \rightarrow \mathit{IsCarried}(x))$.

Prelude: Ambiguity in Natural Language

- Two signs at the foot of a public escalator:
  - Shoes must be worn
  - Dogs must be carried
- What do they mean?
- In formal logic, “Shoes must be worn” may be expressed as:
  $\forall x(\mathit{OnEscalator}(x) \rightarrow \exists y(\mathit{PairOfShoes}(y) \wedge \mathit{IsWearing}(x, y)))$.
- And, “Dogs must be carried” as:
  $\forall x(\mathit{OnEscalator}(x) \wedge \mathit{IsDog}(x) \rightarrow \mathit{IsCarried}(x))$.
- Note that the logic used above is “untyped” classical first-order logic; types, if needed, are enforced by predicates.

Prelude: Ambiguity in Natural Language

- Two signs at the foot of a public escalator:
  - Shoes must be worn
  - Dogs must be carried
- What do they mean?
- In formal logic, "Shoes must be worn" may be expressed as:
  $\forall x(\mathit{OnEscalator}(x) \rightarrow \exists y(\mathit{PairOfShoes}(y) \wedge \mathit{IsWearing}(x, y)))$.
- And, "Dogs must be carried" as:
  $\forall x(\mathit{OnEscalator}(x) \wedge \mathit{IsDog}(x) \rightarrow \mathit{IsCarried}(x))$.
- Note that the logic used above is "untyped" classical first-order logic; types, if needed, are enforced by predicates.

Source: the example is due to M. Jackson [Jackson 1995].

What Formal Logic Is

- Logic concerns two concepts:
  - truth (in a specific or general context/model)
  - provability (of truth from assumed truth)
- *Formal (symbolic) logic* approaches logic by rules for manipulating symbols:
  - syntax rules: for writing statements or formulae.
    (There are also semantic rules determining whether a statement is true or false in a context or mathematical structure.)
  - inference rules: for obtaining true statements from other true statements.
    (It is also possible to confirm true statements by considering all possible contexts.)
- Two main branches of formal logic:
  - *propositional logic* (sentential logic; cf. Boolean algebra)
  - *first-order logic* (predicate logic/calculus)

Why We Need It in Software Development

- Correctness of software hinges on a precise statement of its requirements.
- Logical formulae give the most precise kind of statements about software requirements.
- The fact that "a software program satisfies a requirement (property)" is very much the same as "a mathematical structure satisfies a logical formula (property)":

$$\mathit{prog} \models \mathit{req} \quad \text{vs.} \quad M \models \varphi$$

- To prove (formally verify) that a software program is correct, one may utilize the kind of inferences seen in formal logic.
- The verification may be done manually, semi-automatically, or fully automatically.

A Bit More About Program Correctness

- For a sequential program (or code segment), its correctness requirement (property) may be specified by a pair of conditions, conventionally in the form of $\{P\}\ S\ \{Q\}$ (cf. $S \models [P, Q]$).
  - Pre-condition (P): what Program S requires/assumes
  - Post-condition (Q): what Program S ensures/guarantees
- These conditions are best expressed using formal logic formulae.
- For instance,

$$\{\exists i(0 \leq i < n \wedge A[i] = x)\}\ S\ \{0 \leq m < n \wedge A[m] = x\}$$

  says that, assuming the value x is in the array A, Program S finds an element in A, indexed by the value of m, that equals to x.

More about this when we introduce Hoare Logic in a subsequent lecture...

Propositions

IM NTU

- A *proposition* is a statement that is either *true* or *false* such as the following:
  - Leslie is a teacher.
  - Leslie is rich.
  - Leslie is a pop singer.
- Simplest (atomic) propositions may be combined to form compound propositions:
  - Leslie is *not* a teacher.
  - *Either* Leslie is not a teacher *or* Leslie is not rich.
  - *If* Leslie is a pop singer, *then* Leslie is rich.

Inferences

IM NTU

- We are given the following assumptions:
  - Leslie is a teacher.
  - Either Leslie is not a teacher or Leslie is not rich.
  - If Leslie is a pop singer, then Leslie is rich.
- We wish to conclude the following:
  - Leslie is not a pop singer.
- The above process is an example of *inference* (deduction). Is it correct?

Symbolic Propositions

IM NTU

- Propositions are represented by *symbols*, when only their truth values are of concern.
  - P: Leslie is a teacher.
  - Q: Leslie is rich.
  - R: Leslie is a pop singer.
- Compound propositions can then be more succinctly written.
  - *not* P: Leslie is not a teacher.
  - *not* P *or* *not* Q: Either Leslie is not a teacher or Leslie is not rich.
  - R *implies* Q: If Leslie is a pop singer, then Leslie is rich.

Symbolic Inferences

- We are given the following assumptions:
  - P (Leslie is a teacher.)
  - *not* P *or not* Q (Either Leslie is not a teacher or Leslie is not rich.)
  - R *implies* Q (If Leslie is a pop singer, then Leslie is rich.)
- We wish to conclude the following:
  - *not* R (Leslie is not a pop singer.)
- Correctness of the inference may be checked by asking:
  - Is (P *and* (*not* P *or not* Q) *and* (R *implies* Q)) *implies* (*not* R) a tautology (valid formula)?
  - Or, is $P \wedge (\neg P \vee \neg Q) \wedge (R \rightarrow Q) \rightarrow \neg R$ valid?

Boolean Expressions and Propositions

- *Boolean expressions* are essentially propositional formulae, though they may allow more things (e.g., $x \geq 0$) as atomic formulae.
- Boolean expressions following variant syntactical conventions:
  - $(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) \wedge x$
  - $(x + y + \overline{z}) \cdot (\overline{x} + \overline{y}) \cdot x$
  - $(a \vee b \vee \overline{c}) \wedge (\overline{a} \vee \overline{b}) \wedge a$
  - etc.
- Propositional formula: $(P \vee Q \vee \neg R) \wedge (\neg P \vee \neg Q) \wedge P$

Normal Forms

- A *literal* is an atomic proposition or its negation.
- A propositional formula is in Conjunctive Normal Form (CNF) if it is a conjunction of disjunctions of literals.
  - $(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q) \land P$
  - $(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q \lor R) \land (P \lor \neg Q \lor \neg R)$
- A propositional formula is in Disjunctive Normal Form (DNF) if it is a disjunction of conjunctions of literals.
  - $(P \land Q \land \neg R) \lor (\neg P \land \neg Q) \lor P$
  - $(\neg P \land \neg Q \land R) \lor (P \land Q \land \neg R) \lor (\neg P \land Q \land R)$
- A propositional formula is in Negation Normal Form (NNF) if negations occur only in literals.
  - CNF or DNF is also NNF (but not vice versa).
  - $(P \land \neg Q) \land (P \lor (Q \land \neg R))$ in NNF, but not CNF or DNF.
- Every propositional formula has an equivalent formula in each of these normal forms.

Models, Satisfiability, and Validity

- *Models* provide the (semantic) context in which a logic formula is judged to be true or false.
- Models are formally represented as mathematical structures.
- A formula can be true in one model, but false in another.
- A model *satisfies* a formula if the formula is true in the model (notation: $M \models \varphi$).
  - $v(P) = F, v(Q) = T \models (P \vee Q) \wedge (\neg P \vee \neg Q)$
- A formula is *satisfiable* if there is a model that satisfies the formula.
- A formula is *valid* if it is true in every model (notation: $\models \varphi$).
  - $\models A \vee \neg A$
  - $\models (A \wedge B) \rightarrow (A \vee B)$

Semantic Entailment

- Let Γ be a set of formulae.
- A model satisfies Γ if the model satisfies every formula in Γ.
- We say that Γ *semantically entails* C if every model that satisfies Γ also satisfies C, written as $\Gamma \models C$.
  - $A, A \rightarrow B \models B$
  - $A \rightarrow B, \neg B \models \neg A$
- A main ingredient of a logic is a systematic way to draw conclusions of the above form, namely $\Gamma \models C$.

Sequents

- We write "$A_1, A_2, \cdots, A_m \vdash C$" to mean that the truth of formula C follows from the truth of formulae $A_1, A_2, \cdots, A_m$.
- "$A_1, A_2, \cdots, A_m \vdash C$" is called a *sequent*.
- In the sequent, $A_1, A_2, \cdots, A_m$ collectively are called the *antecedent* (also *context*) and C the *consequent*.

Note: Many authors prefer to write a sequent as $\Gamma \longrightarrow C$ or $\Gamma \Longrightarrow C$, while reserving the symbol $\vdash$ for provability (deducibility) in the proof (deduction) system under consideration.

Inference Rules

IM NTU

- Inference rules allow one to obtain true statements from other true statements.
- Below is an inference rule for conjunction.

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\ (\wedge I)$$

- In an inference rule, the upper sequents (above the horizontal line) are called the *premises* and the lower sequent is called the *conclusion*.

Proofs

IM NTU

- A deduction tree is a tree where each node is labeled with a sequent such that, for every internal (non-leaf) node,
  - the label of the node corresponds to the conclusion and
  - the labels of its children correspond to the premises

  of an instance of an inference rule.
- A proof tree is a deduction tree, each of whose leaves is labeled with an axiom.
- The root of a deduction or proof tree is called the conclusion.
- A sequent is provable if there exists a proof tree of which it is the conclusion.

Natural Deduction in the Sequent Form

$$\frac{}{\Gamma, A \vdash A} \ (Ax)$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \ (\wedge I)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \ (\wedge E_1)$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \ (\wedge E_2)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \ (\vee I_1)$$

$$\frac{\Gamma \vdash B}{\Gamma \vdash A \vee B} \ (\vee I_2)$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C} \ (\vee E)$$

Natural Deduction (cont.)

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ (\to I) \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ (\to E)$$

$$\frac{\Gamma, A \vdash B \wedge \neg B}{\Gamma \vdash \neg A}\ (\neg I) \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash B}\ (\neg E)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \neg\neg A}\ (\neg\neg I) \qquad \frac{\Gamma \vdash \neg\neg A}{\Gamma \vdash A}\ (\neg\neg E)$$

Note: these inference rules collectively are called System *ND*.

A Proof in Propositional *ND*

Below is a partial proof of the validity of $P \wedge (\neg P \vee \neg Q) \wedge (R \rightarrow Q) \rightarrow \neg R$ in *ND*, where γ denotes $P \wedge (\neg P \vee \neg Q) \wedge (R \rightarrow Q)$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\gamma, R \vdash R \rightarrow Q} \qquad \dfrac{}{\gamma, R \vdash R}\,(Ax)}{\gamma, R \vdash Q}\,(\rightarrow E) \qquad \dfrac{\dfrac{\vdots}{\gamma, R, Q \vdash P \wedge \neg P}}{\gamma, R \vdash \neg Q}\,(\neg I)}{\gamma, R \vdash Q \wedge \neg Q}\,(\wedge I)}{P \wedge (\neg P \vee \neg Q) \wedge (R \rightarrow Q) \vdash \neg R}\,(\neg I)}{\vdash P \wedge (\neg P \vee \neg Q) \wedge (R \rightarrow Q) \rightarrow \neg R}\,(\rightarrow I)
$$

Soundness and Completeness

- A deduction (proof) system is *sound* if it produces only semantically valid results, and it is *complete* if every semantically valid result can be produced.
- More formally, a system is sound if, whenever $\Gamma \vdash C$ is provable in the system, then $\Gamma \models C$.
- A system is complete if, whenever $\Gamma \models C$, then $\Gamma \vdash C$ is provable in the system.
- Soundness allows us to draw semantically valid conclusions from purely syntactical inferences and completeness guarantees that this is always achievable.

Predicates

- A *predicate* is a "parameterized" statement that, when supplied with actual arguments, is either *true* or *false* such as the following:
  - Leslie is a teacher.
  - Chris is a teacher.
  - Leslie is a pop singer.
  - Chris is a pop singer.
- Like propositions, simplest (atomic) predicates may be combined to form compound predicates.

Inferences

- We are given the following assumptions:
  - *For any* person, *either* the person is not a teacher *or* the person is not rich.
  - *For any* person, *if* the person is a pop singer, *then* the person is rich.
- We wish to conclude the following:
  - *For any* person, *if* the person is a teacher, *then* the person is not a pop singer.

Symbolic Predicates

- Like propositions, predicates are represented by *symbols*.
  - $p(x)$: x is a teacher.
  - $q(x)$: x is rich.
  - $r(y)$: y is a pop singer.
- Compound predicates can be expressed:
  - For all x, $r(x) \to q(x)$: *For any* person, *if* the person is a pop singer, *then* the person is rich.
  - For all y, $p(y) \to \neg r(y)$: *For any* person, *if* the person is a teacher, *then* the person is *not* a pop singer.

Symbolic Inferences

IM NTU

- We are given the following assumptions:
  - For all $x, \neg p(x) \vee \neg q(x)$.
  - For all $x, r(x) \to q(x)$.
- We wish to conclude the following:
  - For all $x, p(x) \to \neg r(x)$.
- To check the correctness of the inference above, we ask:
  - is $((\text{for all } x, \neg p(x) \vee \neg q(x)) \wedge (\text{for all } x, r(x) \to q(x))) \to (\text{for all } x, p(x) \to \neg r(x))$ valid?
  - or, is $\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \to q(x)) \to \forall x(p(x) \to \neg r(x))$ valid?

Syntax and Semantics by Examples

- A first-order formula is written using logical and non-logical symbols.
  - logical symbols: variables, boolean connectives, and quantifiers (which are standard)
  - non-logical symbols: predicates, functions, and constants (which vary, depending on the purpose)
- Below are some terms and formulae in the simple language with predicate $=$, function $\cdot$, and constant e:
  - terms: e, x, $x \cdot y$, $x \cdot (y \cdot z)$, etc..
  - formulae: $\forall x((x \cdot e = e \cdot x) \wedge (e \cdot x = x))$ or
    $\forall x(x \cdot e = e \cdot x = x)$,
    $\forall x(\forall y(\forall z(x \cdot (y \cdot z) = (x \cdot y) \cdot z)))$ or
    $\forall x, y, z(x \cdot (y \cdot z) = (x \cdot y) \cdot z)$, etc.
- What do the formulae mean?
  - $(Z, \{+, 0\}) \models \forall x(x \cdot e = e \cdot x = x)$
  - $(Q \setminus \{0\}, \{\times, 1\}) \models \forall x, y, z(x \cdot (y \cdot z) = (x \cdot y) \cdot z)$

What about Types

- Ordinary first-order formulae are interpreted over a single domain of discourse (the universe).
- A variant of first-order logic, called many-sorted (or typed) first-order logic, allows variables of different sorts (which correspond to partitions of the universe).
- When the number of sorts is finite, one can emulate sorts by introducing additional unary predicates in the ordinary first-order logic.
  - Suppose there are two sorts.
  - We introduce two new unary predicates P_1 and P_2.
  - We then stipulate that $\forall x(P_1(x) \vee P_2(x)) \wedge \neg(\exists x(P_1(x) \wedge P_2(x)))$.
  - For example, $\exists x(P_1(x) \wedge \varphi(x))$ means that there is an element of the first sort satisfying φ; $\forall x(P_1(x) \to \psi(x))$ means that every element of the first sort satisfies ψ.

Free and Bound Variables

- In a formula $\forall x A$ (or $\exists x A$), the variable x is *bound* by the quantifier $\forall$ (or $\exists$).
- A *free* variable is one that is not bound.
- The same variable may have both a free and a bound occurrence.
- For example, consider
  $(\forall x(R(x, \underline{y}) \rightarrow P(x)) \wedge \forall y(\neg R(\underline{x}, y) \wedge \forall x P(x)))$.
  The underlined occurrences of x and y are free, while others are bound.
- A formula is *closed*, also called a *sentence*, if it does not contain a free variable.

Substitutions

IM NTU

- Let t be a term (such as x, $g(x,y)$, etc.) and A a formula.
- The result of substituting t for a free variable x in A is denoted by $A[t/x]$.
- Consider $A = \forall x(P(x) \to Q(x, f(y)))$.
  - When $t = g(y)$, $A[t/y] = \forall x(P(x) \to Q(x, f(g(y))))$.
  - For any t, $A[t/x] = \forall x(P(x) \to Q(x, f(y))) = A$, since there is no free occurrence of x in A.
- A substitution is *admissible* if no free variable of t would become bound (be captured by a quantifier) after the substitution.
- For example, when $t = g(x,y)$, $A[t/y]$ is not admissible, as the free variable x of t would become bound.

Quantifier Rules of Natural Deduction

$$\frac{\Gamma \vdash A[y/x]}{\Gamma \vdash \forall x A} (\forall I) \qquad \frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[t/x]} (\forall E)$$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x A} (\exists I) \qquad \frac{\Gamma \vdash \exists x A \qquad \Gamma, A[y/x] \vdash B}{\Gamma \vdash B} (\exists E)$$

In the rules above, we assume that all substitutions are admissible and y does not occur free in Γ or A.

A Proof in First-Order *ND*

Below is a partial proof of the validity of $\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)) \rightarrow \forall x(p(x) \rightarrow \neg r(x))$ in *ND*, where γ denotes $\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x))$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\gamma, p(y), r(y) \vdash r(y) \rightarrow q(y)} \quad \dfrac{}{\gamma, p(y), r(y) \vdash r(y)}\,(Ax)}{\gamma, p(y), r(y) \vdash q(y)}\,(\rightarrow E) \quad \vdots}{\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)), p(y), r(y) \vdash q(y) \wedge \neg q(y)}\,(\wedge I)}{\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)), p(y) \vdash \neg r(y)}\,(\neg I)}{\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)) \vdash p(y) \rightarrow \neg r(y)}\,(\rightarrow I)}{\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)) \vdash \forall x(p(x) \rightarrow \neg r(x))}\,(\forall I)}{\vdash \forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \rightarrow q(x)) \rightarrow \forall x(p(x) \rightarrow \neg r(x))}\,(\rightarrow I)
$$

Equality Rules of Natural Deduction

Let t, t_1, t_2 be arbitrary terms; again, assume all substitutions are admissible.

$$\frac{}{\Gamma \vdash t = t}\,(=I) \qquad\qquad \frac{\Gamma \vdash t_1 = t_2 \qquad \Gamma \vdash A[t_1/x]}{\Gamma \vdash A[t_2/x]}\,(=E)$$

Note: The $=$ sign is part of the object language, not a meta symbol.

Theory

- Assume a fixed first-order language.
- A set S of sentences is closed under provability if

$$S = \{A \mid A \text{ is a sentence and } S \vdash A \text{ is provable}\}.$$

- A set of sentences is called a *theory* if it is closed under provability.
- A theory is typically represented by a smaller set of sentences, called its *axioms*.

Note: a sentence is a formula without free variables. For example, $\forall x(x \geq 0)$ is a sentence, but $x \geq 0$ is not.

Group as a First-Order Theory

- The set of non-logical symbols is $\{\cdot, e\}$, where $\cdot$ is a binary function (operation) and e is a constant (the identity).
- Axioms:
  - $\forall a, b, c(a \cdot (b \cdot c) = (a \cdot b) \cdot c)$ (Associativity)
  - $\forall a(a \cdot e = e \cdot a = a)$ (Identity)
  - $\forall a(\exists b(a \cdot b = b \cdot a = e))$ (Inverse)
- $(Z, \{+, 0\})$ is a model of the theory.
- So is $(Q \setminus \{0\}, \{\times, 1\})$.
- Additional axiom for Abelian groups:
  - $\forall a, b(a \cdot b = b \cdot a)$ (Commutativity)

Theorems

- A *theorem* is just a statement (sentence) in a theory (a set of sentences).
- For example, the following are theorems in Group theory:
  - $\forall a \forall b \forall c((a \cdot b = a \cdot c) \rightarrow b = c)$.
  - $\forall a \forall b \forall c(((a \cdot b = e) \wedge (b \cdot a = e) \wedge (a \cdot c = e) \wedge (c \cdot a = e)) \rightarrow b = c)$, which says that every element has a unique inverse.