Elementary Logic

(Based on [Gallier 1986], [Goubault-Larrecq and Mackie 1997], and [Huth and Ryan 2004])

Yih-Kuen Tsay

Department of Information Management
National Taiwan University

Outline

Propositions

- 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

- 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

- 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 (*A and* (*not A or not B*) *and* (*C implies B*)) *implies* (*not C*) a tautology (valid formula)?

Propositional Logic: Syntax

- Vocabulary:
  - A countable set $\mathcal{P}$ of *proposition symbols* (variables): $P, Q, R, \ldots$ (also called *atomic propositions*);
  - *Logical connectives* (operators): $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ and sometimes the constant $\perp$ (*false*);
  - Auxiliary symbols: "(", ")".
- How to read the logical connectives.
  - $\neg$ (negation): not
  - $\wedge$ (conjunction): and
  - $\vee$ (disjunction): or
  - $\rightarrow$ (implication): implies (or if . . . , then . . .)
  - $\leftrightarrow$ (equivalence): is equivalent to (or if and only if)
  - $\perp$ (*false* or bottom): false (or bottom)

Propositional Logic: Syntax (cont.)

- *Propositional Formulae*:
  - Any $A \in \mathcal{P}$ is a formula and so is $\bot$ (these are the "atomic" formula).
  - If A and B are formulae, then so are $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.
- A is called a *subformula* of $\neg A$, and A and B subformulae of $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.
- Precedence (for avoiding excessive parentheses):
  - $A \wedge B \rightarrow C$ means $((A \wedge B) \rightarrow C)$.
  - $A \rightarrow B \vee C$ means $(A \rightarrow (B \vee C))$.
  - $A \rightarrow B \rightarrow C$ means $(A \rightarrow (B \rightarrow C))$.
  - More about this later ...

About Boolean Expressions

- *Boolean expressions* are essentially propositional formulae, though they may allow more things as atomic formulae.
- Boolean expressions:
  - $(x \lor y \lor \overline{z}) \land (\overline{x} \lor \overline{y}) \land x$
  - $(x + y + \overline{z}) \cdot (\overline{x} + \overline{y}) \cdot x$
  - $(a \lor b \lor \overline{c}) \land (\overline{a} \lor \overline{b}) \land a$
  - etc.
- Propositional formula: $(P \lor Q \lor \neg R) \land (\neg P \lor \neg Q) \land P$

Propositional Logic: Semantics

- The meanings of propositional formulae may be conveniently summarized by the truth table:

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

  The meaning of $\perp$ is always F (false).
- There is an implicit inductive definition in the table. We shall try to make this precise.

Truth Assignment and Valuation

- The semantics of propositional logic assigns a truth function to each propositional formula.
- Let *BOOL* be the set of truth values $\{T, F\}$.
- A *truth assignment* (valuation) is a function from $\mathcal{P}$ (the set of proposition symbols) to *BOOL*.
- Let *PROPS* be the set of all propositional formulae.
- A truth assignment v may be extended to a *valuation* function $\hat{v}$ from *PROPS* to *BOOL* as follows:

Truth Assignment and Valuation (cont.)

$$
\begin{array}{rcl}
\hat{v}(\bot) & = & F \\
\hat{v}(P) & = & v(P) \ \ \text{for all } P \in \mathcal{P} \\
\hat{v}(P) & = & \text{as defined by the table below, otherwise}
\end{array}
$$

$\hat{v}(A)$	$\hat{v}(B)$	$\hat{v}(\neg A)$	$\hat{v}(A \wedge B)$	$\hat{v}(A \vee B)$	$\hat{v}(A \to B)$	$\hat{v}(A \leftrightarrow B)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

So, the truth value of a propositional formula is completely determined by the truth values of its subformulae.

Truth Assignment and Satisfaction

- We say $v \models A$ (v *satisfies* A) if $\hat{v}(A) = T$.
- So, the symbol $\models$ denotes a binary relation, called *satisfaction*, between truth assignments and propositional formulae.
- $v \not\models A$ (v *falsifies* A) if $\hat{v}(A) = F$.

Satisfaction

- Alternatively (in a more generally applicable format), the satisfaction relation $\models$ may be defined as follows:

$$
\begin{array}{lcl}
v \not\models \bot & & \\
v \models P & \iff & v(P) = T, \quad \text{for all } P \in \mathcal{P} \\
v \models \neg A & \iff & v \not\models A \text{ (it is } \textit{not} \text{ the case that } v \models A) \\
v \models A \wedge B & \iff & v \models A \textit{ and } v \models B \\
v \models A \vee B & \iff & v \models A \textit{ or } v \models B \\
v \models A \to B & \iff & v \not\models A \textit{ or } v \models B \\
v \models A \leftrightarrow B & \iff & (v \models A \textit{ and } v \models B) \\
 & & \textit{or } (v \not\models A \textit{ and } v \not\models B)
\end{array}
$$

Object vs. Meta Language

- The language that we study is referred to as the *object* language.
- The language that we use to study the object language is referred to as the *meta* language.
- For example, *not*, *and*, and *or* that we used to define the satisfaction relation $\models$ are part of the meta language.

Satisfiability

- A proposition A is *satisfiable* if there exists an assignment v such that $v \models A$.
  - $v(P) = F, v(Q) = T \models (P \vee Q) \wedge (\neg P \vee \neg Q)$
- A proposition is *unsatisfiable* if no assignment satisfies it.
  - $(\neg P \vee Q) \wedge (\neg P \vee \neg Q) \wedge P$ is unsatisfiable.
- The problem of determining whether a given proposition is satisfiable is called the *satisfiability problem*.

Tautology and Validity

- A proposition A is a *tautology* if every assignment satisfies A, written as $\models A$.
  - $\models A \vee \neg A$
  - $\models (A \wedge B) \rightarrow (A \vee B)$
- The problem of determining whether a given proposition is a tautology is called the *tautology problem*.
- A proposition is also said to be *valid* if it is a tautology.
- So, the problem of determining whether a given proposition is valid (a tautology) is also called the *validity problem*.

Note: the notion of a tautology is restricted to propositional logic. In first-order logic, we also speak of valid formulae.

Validity vs. Satisfiability

IM NTU

Theorem

A proposition A is valid (a tautology) if and only if $\neg A$ is unsatisfiable.

So, there are two ways of proving that a proposition A is a tautology:

- A is satisfied by every truth assignment (or A *cannot be falsified* by any truth assignment).
- $\neg A$ is unsatisfiable.

Relating the Logical Connectives

Lemma

$$\models (A \leftrightarrow B) \leftrightarrow ((A \rightarrow B) \wedge (B \rightarrow A))$$

$$\models (A \rightarrow B) \leftrightarrow (\neg A \vee B)$$

$$\models (A \vee B) \leftrightarrow \neg(\neg A \wedge \neg B)$$

$$\models \perp \leftrightarrow (A \wedge \neg A)$$

Note: these equivalences imply that some connectives could be dispensed with. We normally want a smaller set of connectives when analyzing properties of the logic and a larger set when actually using the logic.

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.

Semantic Entailment

- Consider two sets of propositions Γ and Δ.
- We say that $v \models \Gamma$ (v satisfies Γ) if $v \models B$ for every $B \in \Gamma$; analogously for Δ.
- We say that Δ is a *semantic consequence* of Γ if every assignment that satisfies Γ also satisfies Δ, written as $\Gamma \models \Delta$.
  - $A, A \to B \models A, B$
  - $A \to B, \neg B \models \neg A$
- We also say that Γ *semantically entails* Δ when $\Gamma \models \Delta$.

Sequents

- A (propositional) *sequent* is an expression of the form $\Gamma \vdash \Delta$, where $\Gamma = A_1, A_2, \cdots, A_m$ and $\Delta = B_1, B_2, \cdots, B_n$ are finite (possibly empty) sequences of (propositional) formulae.
- In a sequent $\Gamma \vdash \Delta$, Γ is called the *antecedent* (also *context*) and Δ the *consequent*.

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

Sequents (cont.)

- A sequent $A_1, A_2, \cdots, A_m \vdash B_1, B_2, \cdots, B_n$ is falsifiable if there exists a valuation v such that
  $v \models (A_1 \wedge A_2 \wedge \cdots \wedge A_m) \wedge (\neg B_1 \wedge \neg B_2 \wedge \cdots \wedge \neg B_n)$.
  - $A \vee B \vdash B$ is falsifiable, as
    $v(A) = T, v(B) = F \models (A \vee B) \wedge \neg B$.
- A sequent $A_1, A_2, \cdots, A_m \vdash B_1, B_2, \cdots, B_n$ is valid if, for every valuation v, $v \models A_1 \wedge A_2 \wedge \cdots \wedge A_m \rightarrow B_1 \vee B_2 \vee \cdots \vee B_n$.
  - $A \vdash A, B$ is valid.
  - $A, B \vdash A \wedge B$ is valid.
- A sequent is valid if and only if it is not falsifiable.
- In the following, we will use only sequents of this simpler form: $A_1, A_2, \cdots, A_m \vdash C$, where C is a formula.

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

- 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.

Detour: Another Style of Proofs

- Proofs may also be carried out in a calculational style (like in algebra); for example,

$$
\begin{array}{ll}
 & (A \vee B) \rightarrow C \\
\equiv & \{\ A \rightarrow B \ \equiv\ \neg A \vee B\ \} \\
 & \neg(A \vee B) \vee C \\
\equiv & \{\ \text{de Morgan's law}\ \} \\
 & (\neg A \wedge \neg B) \vee C \\
\equiv & \{\ \text{distributive law}\ \} \\
 & (\neg A \vee C) \wedge (\neg B \vee C) \\
\equiv & \{\ A \rightarrow B \ \equiv\ \neg A \vee B\ \} \\
 & (A \rightarrow C) \wedge (B \rightarrow C) \\
\Rightarrow & \{\ A \wedge B \ \Rightarrow\ A\ \} \\
 & (A \rightarrow C)
\end{array}
$$

- Here, $\Rightarrow$ corresponds to semantical entailment and $\equiv$ to mutual semantical entailment. Both are transitive.

Detour: Some Laws for Calculational Proofs

- Equivalence is commutative and associative
  - $A \leftrightarrow B \quad \equiv \quad B \leftrightarrow A$
  - $A \leftrightarrow (B \leftrightarrow C) \quad \equiv \quad (A \leftrightarrow B) \leftrightarrow C$
- $\bot \vee A \quad \equiv \quad A \vee \bot \quad \equiv \quad A$
- $\neg A \wedge A \quad \equiv \quad \bot$
- $A \rightarrow B \quad \equiv \quad \neg A \vee B$
- $A \rightarrow \bot \quad \equiv \quad \neg A$
- $(A \vee B) \rightarrow C \quad \equiv \quad (A \rightarrow C) \wedge (B \rightarrow C)$
- $A \rightarrow (B \rightarrow C) \quad \equiv \quad (A \wedge B) \rightarrow C$
- $A \rightarrow B \quad \equiv \quad A \leftrightarrow (A \wedge B)$
- $A \wedge B \quad \Rightarrow \quad A$

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.)

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

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

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

These inference rules collectively are called System *ND* (the propositional part).

A Proof in Propositional *ND*

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

$$
\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\gamma, R \vdash R \to Q} \quad \dfrac{}{\gamma, R \vdash R}(Ax)}{\gamma, R \vdash Q}(\to E) \quad \dfrac{\dfrac{\vdots}{\gamma, R, Q \vdash P \land \neg P}}{\gamma, R \vdash \neg Q}(\neg I)}{\dfrac{\gamma, R \vdash Q \land \neg Q}{P \land (\neg P \lor \neg Q) \land (R \to Q) \vdash \neg R}(\neg I)}(\land I)}{\vdash P \land (\neg P \lor \neg Q) \land (R \to Q) \to \neg R}(\to I)
$$

Soundness and Completeness

Theorem

System ND is sound, i.e., if a sequent $\Gamma \vdash C$ *is provable in ND, then* $\Gamma \vdash C$ *is valid.*

Theorem

System ND is complete, i.e., if a sequent $\Gamma \vdash C$ *is valid, then* $\Gamma \vdash C$ *is provable in ND.*

Compactness

A set Γ of propositions is satisfiable if some valuation satisfies every proposition in Γ. For example, $\{A \vee B, \neg B\}$ is satisfiable.

Theorem

For any (possibly infinite) set Γ of propositions, if every finite non-empty subset of Γ is satisfiable then Γ is satisfiable.

Proof hint: by contradiction and the completeness of *ND*.

Consistency

- A set Γ of propositions is *consistent* if there exists some proposition B such that the sequent $\Gamma \vdash B$ is not provable.
- Otherwise, Γ is *inconsistent*; e.g., $\{A, \neg(A \vee B)\}$ is inconsistent.

Lemma

For System ND, a set Γ of propositions is inconsistent if and only if there is some proposition A such that both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$ are provable.

Theorem

For System ND, a set Γ of propositions is satisfiable if and only if Γ is consistent.

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, $r(x) \rightarrow q(x)$: *For any* person, *if* the person is a pop singer, *then* the person is rich.
  - For all y, $p(y) \rightarrow \neg r(y)$: *For any* person, *if* the person is a teacher, *then* the person is *not* a pop singer.

Symbolic Inferences

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

First-Order Logic: Syntax

- Logical symbols:
  - A countable set V of *variables*: $x, y, z, \ldots$;
  - *Logical connectives* (operators): $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\perp$, $\forall$ (for all), $\exists$ (there exists);
  - Auxiliary symbols: "(", ")".
- Non-logical symbols:
  - A countable set of *function symbols* with associated ranks (arities);
  - A countable set of *constants* (which may be seen as functions with rank 0);
  - A countable set of *predicate symbols* with associated ranks (arities);
- We refer to a first-order language as *Language L*, where L is the set of non-logical symbols (e.g., $\{+, 0, 1, <\}$). The set L is usually referred to as the *signature* of the first-order language.

First-Order Logic: Syntax (cont.)

- Terms:
  - Every *constant* and every *variable* is a term.
  - If $t_1, t_2, \cdots, t_k$ are terms and f is a k-ary function symbol ($k > 0$), then $f(t_1, t_2, \cdots, t_k)$ is a term.
- Atomic formulae:
  - Every *predicate symbol* of 0-arity is an atomic formula and so is $\perp$.
  - If $t_1, t_2, \cdots, t_k$ are terms and p is a k-ary predicate symbol ($k > 0$), then $p(t_1, t_2, \cdots, t_k)$ is an atomic formula.
- For example, consider Language $\{+, 0, 1, <\}$.
  - 0, x, $x + 1$, $x + (x + 1)$, etc. are terms.
  - $0 < 1$, $x < (x + 1)$, etc. are atomic formulae.

First-Order Logic: Syntax (cont.)

- Formulae:
  - Every atomic formula is a formula.
  - If A and B are formulae, then so are $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$.
  - If x is a variable and A is a formula, then so are $\forall x A$ and $\exists x A$.
- First-order logic *with equality* includes equality ($=$) as an additional logical symbol, which behaves like a predicate symbol.
- Example formulae in Language $\{+, 0, 1, <\}$:
  - $(0 < x) \vee (x < 1)$
  - $\forall x(\exists y(x + y = 0))$

First-Order Logic: Syntax (cont.)

- We may give the logical connectives different binding powers, or precedences, to avoid excessive parentheses, usually in this order:

$$\neg, \{\forall, \exists\}, \{\wedge, \vee\}, \rightarrow, \leftrightarrow \ .$$

  For example, $(A \wedge B) \rightarrow C$ becomes $A \wedge B \rightarrow C$.
- Common abbreviations:
  - $x = y = z$ means $x = y \wedge y = z$.
  - $p \rightarrow q \rightarrow r$ means $p \rightarrow (q \rightarrow r)$. Implication associates to the right, so do other logical symbols.
  - $\forall x, y, zA$ means $\forall x(\forall y(\forall zA))$.

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.

Free Variables Formally Defined

For a term t, the set $FV(t)$ of free variables of t is defined inductively as follows:

- $FV(x) = \{x\}$, for a variable x;
- $FV(c) = \emptyset$, for a contant c;
- $FV(f(t_1, t_2, \cdots, t_n)) = FV(t_1) \cup FV(t_2) \cup \cdots \cup FV(t_n)$, for an n-ary function f applied to n terms $t_1, t_2, \cdots, t_n$.

Free Variables Formally Defined (cont.)

For a formula A, the set $FV(A)$ of free variables of A is defined inductively as follows:

- $FV(P(t_1, t_2, \cdots, t_n)) = FV(t_1) \cup FV(t_2) \cup \cdots \cup FV(t_n)$, for an n-ary predicate P applied to n terms $t_1, t_2, \cdots, t_n$;
- $FV(t_1 = t_2) = FV(t_1) \cup FV(t_2)$;
- $FV(\neg B) = FV(B)$;
- $FV(B * C) = FV(B) \cup FV(C)$, where $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$;
- $FV(\perp) = \emptyset$;
- $FV(\forall x B) = FV(B) - \{x\}$;
- $FV(\exists x B) = FV(B) - \{x\}$.

Bound Variables Formally Defined

For a formula A, the set $BV(A)$ of bound variables in A is defined inductively as follows:

- $BV(P(t_1, t_2, \cdots, t_n)) = \emptyset$, for an n-ary predicate P applied to n terms $t_1, t_2, \cdots, t_n$;
- $BV(t_1 = t_2) = \emptyset$;
- $BV(\neg B) = BV(B)$;
- $BV(B * C) = BV(B) \cup BV(C)$, where $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$;
- $BV(\perp) = \emptyset$;
- $BV(\forall x B) = BV(B) \cup \{x\}$;
- $BV(\exists x B) = BV(B) \cup \{x\}$.

Substitutions

- Let t be a term and A a formula.
- The result of substituting t for a free variable x in A is denoted by $A[t/x]$.
- Consider $A = \forall x(P(x) \rightarrow Q(x, f(y)))$.
  - When $t = g(y)$, $A[t/y] = \forall x(P(x) \rightarrow Q(x, f(g(y))))$.
  - For any t, $A[t/x] = \forall x(P(x) \rightarrow Q(x, f(y))) = A$, 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.

Substitutions (cont.)

- Suppose we change the bound variable x in A to z and obtain another formula $A' = \forall z(P(z) \to Q(z, f(y)))$.
- Intuitively, A' and A should be equivalent (under any reasonable semantics). (Technically, the two formulae A and A' are said to be α-*equivalent*.)
- We can avoid the capture in $A[g(x,y)/y]$ by renaming the bound variable x to z and the result of the substitution then becomes $A'[g(x,y)/y] = \forall z(P(z) \to Q(z, f(g(x,y))))$.
- So, in principle, we can make every substitution admissible while preserving the semantics.

Substitutions Formally Defined

Let s and t be terms. The result of substituting t in s for a variable x, denoted $s[t/x]$, is defined inductively as follows:

- $x[t/x] = t$;
- $y[t/x] = y$, for a variable y that is not x;
- $c[t/x] = c$, for a contant c;
- $f(t_1, t_2, \cdots, t_n)[t/x] = f(t_1[t/x], t_2[t/x], \cdots, t_n[t/x])$, for an n-ary function f applied to n terms $t_1, t_2, \cdots, t_n$.

Substitutions Formally Defined (cont.)

IM NTU

For a formula A, $A[t/x]$ is defined inductively as follows:

- $P(t_1, t_2, \cdots, t_n)[t/x] = P(t_1[t/x], t_2[t/x], \cdots, t_n[t/x])$, for an n-ary predicate P applied to n terms $t_1, t_2, \cdots, t_n$;
- $(t_1 = t_2)[t/x] = (t_1[t/x] = t_2[t/x])$;
- $(\neg B)[t/x] = \neg B[t/x]$;
- $(B * C)[t/x] = (B[t/x] * C[t/x])$, where $* \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$;
- $\perp[t/x] = \perp$;
- $(\forall x B)[t/x] = (\forall x B)$;
- $(\forall y B)[t/x] = (\forall y B[t/x])$, if variable y is not x;
- $(\exists x B)[t/x] = (\exists x B)$;
- $(\exists y B)[t/x] = (\exists y B[t/x])$, if variable y is not x;

First-Order Structures

- A first-order structure $\mathcal{M}$ is a pair (M, I), where
  - M (a non-empty set) is the *domain* of the structure, and
  - I is the *interpretation function*, that assigns functions and predicates over M to the function and predicate symbols.
- An interpretation may be represented by simply listing the functions and predicates.
- For instance, $(Z, \{+_Z, 0_Z\})$ is a structure for the language $\{+, 0\}$. The subscripts are omitted, as $(Z, \{+, 0\})$, when no confusion may arise.

Semantics of First-Order Logic

- Since a formula may contain free variables, its truth value depends on the specific values that are assigned to these variables.
- Given a first-order language and a structure $\mathcal{M} = (M, I)$, an *assignment* is a function from the set of variables to M.
- The structure $\mathcal{M}$ along with an assignment s determines the truth value of a formula A, denoted as $A_{\mathcal{M}}[s]$.
- For example, $(x + 0 = x)_{(Z,\{+,0\})}[x := 1]$ evaluates to T.

Semantics of First-Order Logic (cont.)

- We say $\mathcal{M}, s \models A$ when $A_{\mathcal{M}}[s]$ is T (true) and $\mathcal{M}, s \not\models A$ otherwise.
- Alternatively, $\models$ may be defined as follows (propositional part is as in propositional logic):

$$\begin{array}{lcll} \mathcal{M}, s \models \forall x A & \iff & \mathcal{M}, s[x := m] \models A & \text{for all } m \in M. \\ \mathcal{M}, s \models \exists x A & \iff & \mathcal{M}, s[x := m] \models A & \text{for some } m \in M. \end{array}$$

  where $s[x := m]$ denotes an updated assignment s' from s such that $s'(y) = s(y)$ for $y \neq x$ and $s'(x) = m$.
- For example, $(Z, \{+, 0\}), s \models \forall x (x + 0 = x)$ holds, since $(Z, \{+, 0\}), s[x := m] \models x + 0 = x$ for all $m \in Z$.

Satisfiability and Validity

IM NTU

- A formula A is *satisfiable in $\mathcal{M}$* if there is an assignment s such that $\mathcal{M}, s \models A$.
- A formula A is *valid in $\mathcal{M}$*, denoted $\mathcal{M} \models A$, if $\mathcal{M}, s \models A$ for every assignment s.
- For instance, $\forall x(x + 0 = x)$ is valid in $(Z, \{+, 0\})$.
- $\mathcal{M}$ is called a *model* of A if A is valid in $\mathcal{M}$.
- A formula A is *valid* if it is valid in every structure, denoted $\models A$.

Relating the Quantifiers

Lemma

$$\models \neg \forall x A \leftrightarrow \exists x \neg A$$

$$\models \neg \exists x A \leftrightarrow \forall x \neg A$$

$$\models \forall x A \leftrightarrow \neg \exists x \neg A$$

$$\models \exists x A \leftrightarrow \neg \forall x \neg A$$

Note: These equivalences show that, with the help of negation, either quantifier can be expressed by the other.

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
$\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \to q(x)) \to \forall x(p(x) \to \neg r(x))$ in *ND*,
where γ denotes $\forall x(\neg p(x) \vee \neg q(x)) \wedge \forall x(r(x) \to q(x))$.

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

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.

Soundness and Completeness

Let System *ND* also include the quantifier rules.

Theorem

System ND is sound, i.e., if a sequent $\Gamma \vdash \Delta$ is provable in ND, then $\Gamma \vdash \Delta$ is valid.

Theorem

System ND is complete, i.e., if a sequent $\Gamma \vdash \Delta$ is valid, then $\Gamma \vdash \Delta$ is provable in ND.

Note: assume no equality in the logic language.

Compactness

IM NTU

Theorem

For any (possibly infinite) set Γ of formulae, if every finite non-empty subset of Γ is satisfiable then Γ is satisfiable.

Consistency

Recall that a set Γ of formulae is *consistent* if there exists some formula B such that the sequent $\Gamma \vdash B$ is not provable. Otherwise, Γ is *inconsistent*.

Lemma

For System ND, a set Γ of formulae is inconsistent if and only if there is some formula A such that both $\Gamma \vdash A$ and $\Gamma \vdash \neg A$ are provable.

Theorem

For System ND, a set Γ of formulae is satisfiable if and only if Γ is consistent.

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*.

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)
- $(Z, \{+, 0\})$ and $(Q \setminus \{0\}, \{\times, 1\})$ are models of the theory.
- 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:
  - $\forall a \forall b \forall c((a \cdot b = a \cdot c) \to b = c)$.
  - $\forall a \forall b \forall c(((a \cdot b = e) \wedge (b \cdot a = e) \wedge (a \cdot c = e) \wedge (c \cdot a = e)) \to b = c)$, which says that every element has a unique inverse.

References

- J.H. Gallier. *Logic for Computer Science: Foundations of Automatic Theorem Proving*, Harper & Row Publishers, 1985.
- J. Goubault-Larrecq and I. Mackie. *Proof Theory and Automated Deduction*, Kluwer Academic Publishers, 1997.
- M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 2004.