Propositional, First-Order And Higher-Order Logics: Basic Definitions, Rules of Inference, and Examples*

Stuart C. Shapiro
Department of Computer Science and Engineering
University at Buffalo, The State University of New York
226 Bell Hall
Buffalo, NY 14260-2000
shapiro@cse.buffalo.edu

1 What is Logic?

Logic is the study of correct reasoning. It is not a particular KRR language. Thus, it is not proper to say "We are using (or not using) logic as our KRR language." There are, indeed, many different logics. For more details on logics, see (Haack, 1978), (McCawley, 1981), and the various articles on Logic in (Shapiro, 1992) beginning with (Rapaport, 1992).

2 Requirements to Define a Logic

A logic consists of two parts, a language and a method of reasoning. The logical language, in turn, has two aspects, syntax and semantics. Thus, to specify or define a particular logic, one needs to specify three things:

Syntax: The atomic symbols of the logical language, and the rules for constructing well-formed, nonatomic expressions (symbol structures) of the logic.

Semantics: The meanings of the atomic symbols of the logic, and the rules for determining the meanings of nonatomic expressions of the logic.

Syntactic Inference Method: The rules for determining a subset of logical expressions, called **theorems** of the logic.

3 CarPool World

We will use CarPool World as a simple example. In CarPool World, Tom and Betty carpool to work. On any day, either Tom drives Betty or Betty drives Tom. In the former case, Tom is the driver and Betty is the passenger. In the latter case, Betty is the driver and Tom is the passenger.

*This is a preprint version of Stuart C. Shapiro, Propositional, First-Order And Higher-Order Logics: Basic Definitions, Rules of Inference, and Examples. In Lucja M. Iwańska & Stuart C. Shapiro, Eds., *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language,* AAAI Press/The MIT Press, Menlo Park, CA, 2000, 379-395, and may differ slightly from the final, published version. All quotes should be from and all citations should be of the published version.

4 Propositional Logic

4.1 Propositional CarPool World

Propositional logics (sometimes called Sentential Logics) conceptualize domains at, but not below the level of sentences (or propositions). So the finest analysis of CarPool World is that there are six sentences:

Betty drives Tom.	*Tom drives Betty.*
Betty is the driver.	*Tom is the driver.*
Betty is the passenger.	*Tom is the passenger.*

4.2 Syntax

The syntactic expressions of propositional logics consist of **atomic propositions** and nonatomic, **well-formed propositions (wfps)**.

Syntax of Atomic Propositions

- Any letter of the alphabet, e.g.: P
- Any letter of the alphabet with a numeric subscript, e.g.: Q_3
- Any alphanumeric string, e.g.: *Tom is the driver*

is an atomic proposition.

Syntax of Well-Formed Propositions (WFPs)

1. Every atomic proposition is a wfp.
2. If P is a wfp, then so is $\neg P$.
3. If P and Q are wfps, then so are

   (a) $(P \wedge Q)$ (b) $(P \vee Q)$

   (c) $(P \Rightarrow Q)$ (d) $(P \Leftrightarrow Q)$
4. Nothing else is a wfp.

We will not bother using parentheses when there is no ambiguity, in which case $\wedge$ and $\vee$ will have higher priority than $\Rightarrow$, which, in turn will have higher priority than $\Leftrightarrow$. For example, we will write $P \wedge Q \Leftrightarrow \neg P \Rightarrow Q$ instead of $((P \wedge Q) \Leftrightarrow (\neg P \Rightarrow Q))$.

An example wfp in CarPool World is

$$\textit{Tom is the driver} \Leftrightarrow \neg \textit{Betty is the driver}$$

4.3 Semantics

To specify the semantics of a propositional logic, we must give the semantics of each atomic proposition and the rules for deriving the semantics of the wfps from their constituent propositions. There are actually two levels of semantics we must specify: **extensional semantics** and **intensional semantics**.

The **extensional semantics** (value or **denotation**) of the expressions of a logic are relative to a particular interpretation, model, or situation. The extensional semantics of CarPool World, for example, are relative to a particular day. The denotation of a proposition is either True or False. If P is an expression of some logic, we will use $[\![P]\!]$ to mean the denotation of P. If we need to make explicit that we mean the denotation relative to situation S, we will use $[\![P]\!]_S$.

The **intensional semantics** (or **intension**) of the expressions of a logic are independent of any specific interpretation, model, or situation, but are dependent only on the domain being conceptualized. If P is an expression of some logic, we will use $[P]$ to mean the intension of P. If we need to make explicit that we mean the intension relative to domain D, we will use $[P]_D$. Many formal people consider the intension of an expression to be a function from situations to denotations. For them, $[P]_D(S) = [\![P]\!]_S$. However, less formally, the intensional semantics of a wfp can be given as a statement in a previously understood language (for example, English) that allows the extensional value to be determined in any specific situation. Intensional semantics are often omitted when a logic is specified, but they shouldn't be.

4.3.1 Intensional Semantics of Atomic Propositions

The intensional semantics of atomic propositions must be specified for each particular propositional logic. For example, the intensional semantics of the atomic propositions of CarPool World are:

[*Betty drives Tom*] = The person named "Tom" gets a ride in to work with the person named "Betty".

[*Tom drives Betty*] = The person named "Betty" gets a ride in to work with the person named "Tom".

[*Betty is the driver*] = The person named "Betty" is the driver of the car.

[*Tom is the driver*] = The person named "Tom" is the driver of the car.

[*Betty is the passenger*] = The person named "Betty" is a passenger in the car.

[*Tom is the passenger*] = The person named "Tom" is a passenger in the car.

Note that each atomic proposition is a single indivisible symbol; the fact that the atomic propositions look like English sentences whose meanings are paraphrases of the intensional semantics is purely for mnemonic purposes. One should never rely on "pretend it's English" semantics.

4.3.2 Intensional Semantics of WFPs

Since the **logical connectives** $\neg$, $\wedge$, $\vee$, $\Rightarrow$, and $\Leftrightarrow$ are commonly used, the following clauses are the standard ones for deriving the intensional semantics of wfps from the intensional semantics of their constituents:

- $[\neg P]$ = It is not the case that $[P]$.
- $[P \wedge Q]$ = $[P]$ and $[Q]$.
- $[P \vee Q]$ = Either $[P]$ or $[Q]$ or both.
- $[P \Rightarrow Q]$ = If $[P]$ then $[Q]$.
- $[P \Leftrightarrow Q]$ = $[P]$ if and only if $[Q]$.

4.3.3 Extensional Semantics of Atomic Propositions

The denotation of an atomic proposition is a truth value, True or False. Each way of assigning a truth value to each atomic proposition forms one situation. For example, each column of the following table gives one situation of CarPool World.

	Denotation in Situation				
Proposition	1	2	3	4	5
Betty drives Tom	True	True	True	False	False
Tom drives Betty	True	True	False	True	False
Betty is the driver	True	True	True	False	False
Tom is the driver	True	False	False	True	False
Betty is the passenger	True	False	False	True	False
Tom is the passenger	True	False	True	False	False

This shows 5 situations. Since there are 6 propositions, and each one can have either of 2 truth values, there are $2^6 = 64$ different situations in CarPool World. We will see below how to limit these to the two that "make sense."

4.3.4 Extensional Semantics of WFPs

Just as there is a standard way to derive the intensional semantics of wfps from their constituents, so is there a standard way to compute the denotations of wfps from their constituents. These are:

- $[\![\neg P]\!]$ is True if $[\![P]\!]$ is False. Otherwise, it is False.
- $[\![P \wedge Q]\!]$ is True if $[\![P]\!]$ is True and $[\![Q]\!]$ is True. Otherwise, it is False.
- $[\![P \vee Q]\!]$ is False if $[\![P]\!]$ is False and $[\![Q]\!]$ is False. Otherwise, it is True.
- $[\![P \Rightarrow Q]\!]$ is False if $[\![P]\!]$ is True and $[\![Q]\!]$ is False. Otherwise, it is True.
- $[\![P \Leftrightarrow Q]\!]$ is True if $[\![P]\!]$ and $[\![Q]\!]$ are both True, or both False. Otherwise, it is False.

These can also be shown in the following **truth tables.**

P	True	False
$\neg P$	False	True

P	True	True	False	False
Q	True	False	True	False
$P \wedge Q$	True	False	False	False
$P \vee Q$	True	True	True	False
$P \Rightarrow Q$	True	False	True	True
$P \Leftrightarrow Q$	True	False	False	True

Notice that each column of these tables represents a different situation.

4.3.5 Semantic Properties of WFPs

A wfp is either **satisfiable, contingent, valid,** or **contradictory** according to the situations in which it is True. A wfp is **satisfiable** if it is True in at least one situation, **contingent** if it is True in at least one situation and False in at least one situation, **valid** if it is True in every situation, and **contradictory** of it is False in every situation. For example, as the following table shows, $\neg P$, $Q \Rightarrow P$, and $P \Rightarrow (Q \Rightarrow P)$ are satisfiable, $\neg P$ and $Q \Rightarrow P$ are contingent, $P \Rightarrow (Q \Rightarrow P)$ is valid, and $P \wedge \neg P$ is contradictory.

P	True	True	False	False
Q	True	False	True	False
$\neg P$	False	False	True	True
$Q \Rightarrow P$	True	True	False	True
$P \Rightarrow (Q \Rightarrow P)$	True	True	True	True
$P \wedge \neg P$	False	False	False	False

If A is a well-formed expression of a logic $\mathcal{L}$, it is standard to write $\models_{\mathcal{L}} A$ (The symbol "$\models$" is called a "double turnstyle".) to indicate that A is valid in logic $\mathcal{L}$. The subscript may be omitted if it is clear from context. Thus, the above truth table shows that $\models P \Rightarrow (Q \Rightarrow P)$. Valid wfps are also called **tautologies.**

Related to the notion of validity is the notion of **logical implication**. The set of wfps $\{A_1, \ldots, A_n\}$ logically implies the wfp B in logic $\mathcal{L}$ (written $A_1, \ldots, A_n \models_{\mathcal{L}} B$) if and only if B is True in every situation in which every A_i is True. This is how domain knowledge can be used to reduce the set of situations to only those that "make sense." For example, in CarPool World, we want to specify that:

- Betty is the driver or the passenger, but not both:

$$\textit{Betty is the driver} \Leftrightarrow \neg \textit{Betty is the passenger}$$

- Tom is the driver or the passenger, but not both:

$$\textit{Tom is the driver} \Leftrightarrow \neg \textit{Tom is the passenger}$$

- If Betty drives Tom, then Betty is the driver and Tom is the passenger:

$$\textit{Betty drives Tom} \Rightarrow \textit{Betty is the driver} \wedge \textit{Tom is the passenger}$$

- If Tom drives Betty, then Tom is the driver and Betty is the passenger:

$$\textit{Tom drives Betty} \Rightarrow \textit{Tom is the driver} \wedge \textit{Betty is the passenger}$$

- Finally, either Tom drives Betty or Betty drives Tom:

$$\textit{Tom drives Betty} \vee \textit{Betty drives Tom}$$

The following table shows the only two situations of CarPool World (numbered as in the previous table) in which all five of these wfps are True.

	Denotation in Situation	
Proposition	3	4
Betty drives Tom	True	False
Tom drives Betty	False	True
Betty is the driver	True	False
Tom is the driver	False	True
Betty is the passenger	False	True
Tom is the passenger	True	False

Notice that these are precisely the two commonsense situations.

Logical implication and logical validity are related by the following

Metatheorem 1: $A_1, \ldots, A_n \models_{\mathcal{L}} B$ if and only if $\models_{\mathcal{L}} A_1 \wedge \cdots \wedge A_n \Rightarrow B$

The significance of this is that if one is interested in determining either logical validity or logical implication, one may solve the other problem instead.

4.4 Inference in Propositional Logics

There are two basic varieties of inference methods in propositional logics, **Hilbert-style** methods, and **natural deduction** methods. Hilbert-style inference methods use a large number of **(logical) axioms** and a small number of **rules of inference,** whereas natural deduction methods use a small number of (logical) axioms (or even none at all) and a large number of rules of inference. Usually there are two rules of inference for each **logical connective**, $\neg$, $\wedge$, $\vee$, $\Rightarrow$, and $\Leftrightarrow$, an **introduction rule**, and an **elimination rule**. These are usually abbreviated by writing the logical connective before "I" or "E", respectively. For example $\neg$**I** is the "negation introduction" rule, and $\wedge$**E** is the "and elimination" rule. The rule $\Rightarrow$ **E** is also often called **modus ponens**.

In Hilbert-style methods, a **derivation** of a wfp, A, from a set of assumptions (or **non-logical axioms**,) Γ, is a list of wfps in which each wfp in the list is either a logical axiom, or a non-logical axiom, or follows from previous wfps in the proof according to one of the rules of inference. A Hilbert-style **proof** of a wfp, A, is a derivation of A from an empty set of assumptions. If A can be derived from Γ in the logic $\mathcal{L}$, we write $\Gamma \vdash_{\mathcal{L}} A$, (The symbol "$\vdash$" is called a "turnstyle".) while if A can be proved in $\mathcal{L}$, we write $\vdash_{\mathcal{L}} A$. If A can be proved in $\mathcal{L}, A$ is called a **theorem** of $\mathcal{L}$.

I will present a natural deduction inference method in more detail. This method is based on methods due to Gentzen, Kleene, and Fitch (see (Kleene, 1950, 86–99,442–443) and (Fitch, 1952)). In this presentation, A and B will stand for any wfps of some propositional logic, $\Gamma, \Delta, \Theta, \Lambda$, and Φ will stand for any (possibly empty) sets of wfps of this logic, and "," will stand for set union, where if either argument of "," is A or B, the singleton set $\{A\}$ or $\{B\}$, respectively, should be understood instead. A derivation, in this method is a list of **expressions** of the form $\Gamma \vdash_{\mathcal{L}} \Delta$, where each expression in the list is either an instance of the

$$\textbf{Axiom: } \Gamma, \Delta \vdash_{\mathcal{L}} \Delta$$

or follows from previous expressions according to one of the following rules of inference. (I will omit the subscript, since all these rules deal with the same logic.)

Hyp: If $\Gamma \vdash \Theta$ then $\Gamma, \Delta \vdash \Theta$

Thin: If $\Gamma \vdash \Phi, \Theta$ then $\Gamma \vdash \Theta$

Cut: If $\Gamma \vdash \Phi, \Lambda$ and $\Delta, \Phi \vdash \Theta$ then $\Gamma, \Delta \vdash \Lambda, \Theta$

$\wedge$**I:** If $\Gamma \vdash A, B, \Theta$ then $\Gamma \vdash A \wedge B, \Theta$

$\wedge$**E:** If $\Gamma \vdash A \wedge B, \Theta$ then $\Gamma \vdash A, B, \Theta$ or $\Gamma \vdash A, \Theta$ or $\Gamma \vdash B, \Theta$

$\vee$**I:** If $\Gamma \vdash A, \Theta$ then $\Gamma \vdash A \vee B, \Theta$ or $\Gamma \vdash B \vee A, \Theta$

$\vee$**E:** If $\Gamma, A \vdash \Theta$ and $\Gamma, B \vdash \Theta$ then $\Gamma, A \vee B \vdash \Theta$

$\neg$**I:** If $\Gamma, A \vdash B, \neg B, \Theta$ then $\Gamma \vdash \neg A$

$\neg$**E:** If $\Gamma \vdash \neg\neg A, \Theta$ then $\Gamma \vdash A, \Theta$

$\Rightarrow$ **I:** If $\Gamma, A \vdash \Delta, B$ then $\Gamma \vdash A \Rightarrow B$

$\Rightarrow$ **E:** If $\Gamma \vdash A, A \Rightarrow B, \Delta$ then $\Gamma \vdash B, \Delta$

$\Leftrightarrow$ **I:** If $\Gamma \vdash A \Rightarrow B, B \Rightarrow A, \Delta$ then $\Gamma \vdash A \Leftrightarrow B, \Delta$

$\Leftrightarrow$ **E:** If $\Gamma \vdash A, A \Leftrightarrow B, \Delta$ then $\Gamma \vdash B, \Delta$
and if $\Gamma \vdash B, A \Leftrightarrow B, \Delta$ then $\Gamma \vdash A, \Delta$

Every line of such a derivation of the form $\Gamma \vdash_{\mathcal{L}} \Delta$ indicates that every wfp in the set Δ can be derived in the logic $\mathcal{L}$ from the set of assumptions Γ, and every line of the form $\vdash_{\mathcal{L}} \Delta$ indicates that every wfp in Δ is a theorem of $\mathcal{L}$. The expression $\Gamma \vdash_{\mathcal{L}} \Delta$ may also be interpreted as a **knowledge base** in which the wfps in Γ are assumptions, or **domain rules** and the wfps in $\Delta - \Gamma$ are derived propositions.

As for logical implication and logical validity, derivation and proof in either Hilbert-style or natural deduction inference methods are related by the following

Metatheorem 2: $A_1, \ldots, A_n \vdash_{\mathcal{L}} B_1, \ldots, B_m$ if and only if $\vdash_{\mathcal{L}} A_1 \wedge \cdots \wedge A_n \Rightarrow B_1 \wedge \cdots \wedge B_m$

Again, the significance of this is that if one is interested in finding either a derivation or a proof, one may solve the other problem instead.

4.4.1 Example Derivation

As an example, I'll show a derivation from the CarPool World domain knowledge (non-logical axioms) of the proposition *Tom drives Betty* $\Rightarrow \neg$*Betty drives Tom*. To save space, I'll use the following abbreviations.

$$
\begin{array}{ll}
BdT: & \textit{Betty drives Tom} \\
TdB: & \textit{Tom drives Betty} \\
Bd: & \textit{Betty is the driver} \\
Td: & \textit{Tom is the driver} \\
Bp: & \textit{Betty is the passenger} \\
Tp: & \textit{Tom is the passenger}
\end{array}
$$

1.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp \vdash TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp$	*Axiom*
2.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp \vdash Td \wedge Bp, Td \Leftrightarrow \neg Tp$	$\Rightarrow E, 1$
3.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp \vdash Td, Td \Leftrightarrow \neg Tp$	$\wedge E, 2$
4.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp \vdash \neg Tp$	$\Leftrightarrow E, 3$
5.	$BdT, BdT \Rightarrow Bd \wedge Tp \vdash BdT, BdT \Rightarrow Bd \wedge Tp$	*Axiom*
6.	$BdT, BdT \Rightarrow Bd \wedge Tp \vdash Bd \wedge Tp$	$\Rightarrow E, 5$
7.	$BdT, BdT \Rightarrow Bd \wedge Tp \vdash Tp$	$\wedge E, 6$
8.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp, BdT, BdT \Rightarrow Bd \wedge Tp \vdash \neg Tp, Tp$	*Cut, 4, 7*
9.	$TdB, TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp, BdT \Rightarrow Bd \wedge Tp \vdash \neg BdT$	$\neg I, 8$
10.	$TdB \Rightarrow Td \wedge Bp, Td \Leftrightarrow \neg Tp, BdT \Rightarrow Bd \wedge Tp \vdash TdB \Rightarrow \neg BdT$	$\Rightarrow I, 9$

This derivation actually only uses three of the five assumptions. The others may be added by the rule of *Hyp*.

5 First-Order Predicate Logic

5.1 Predicate CarPool World

First-Order Predicate Logics (FOPLs) conceptualize domains at and below the level of propositions, down to the level of individuals, properties, and relations. In FOPL CarPool World, there are two individuals, Betty and Tom, two unary properties, being a driver and being a passenger, and one binary relation, drives.

5.2 Syntax

The syntactic expressions of FOPLs consist of **terms, atomic formulas,** and nonatomic, **well-formed formulas (wffs).** Terms, in turn, consist of **individual constants, variables, arbitrary individuals, undetermined individuals** and **functional terms.** Functional terms, atomic formulas, and wffs are nonatomic symbol structures. The atomic symbols of FOPLs are individual constants, variables, arbitrary individuals, undetermined individuals, **function symbols,** and **predicate symbols.** (Note: arbitrary individuals and undetermined individuals are not usually mentioned, but I introduce them so that every formula in a derivation can have a clear semantics.)

Syntax of Atomic Symbols

Individual Constants:

- Any letter of the alphabet (preferably early),
- any (such) letter with a numeric subscript,
- any character string not containing blanks nor other punctuation marks.

is an individual constant, for example: $a, B_{12}, Betty, Tom's\_mother\_in\_law$.

Variables:

- Any letter of the alphabet (preferably late),
- any (such) letter with a numeric subscript.

is a variable, for example: x, y, u_6.

Arbitrary Individuals and Undetermined Individuals

- Any symbol that could be used as an individual constant or as a variable could also be used as an arbitrary individual or an undetermined individual,

for example, $someone, a\_driver, someone's\_mother$.

Function Symbols:

- Any letter of the alphabet (preferably early middle)
- any (such) letter with a numeric subscript
- any character string not containing blanks.

is a function symbol, for example: $f, g_2, mother\_of$.

Predicate Symbols:

- Any letter of the alphabet (preferably late middle),
- any (such) letter with a numeric subscript,
- any character string not containing blanks.

is a predicate symbol, for example: $P, Q_4, odd, Driver$.

Each function symbol and predicate symbol must have a particular **arity,** which may be shown explicitly as a superscript, for example: $mother\_of^1, Drives^2, g_2^3$. The arity need not be shown explicitly if it is understood.

In any specific predicate logic language individual constants, variables, arbitrary individuals, undetermined individuals, function symbols, and predicate symbols must be disjoint.

Syntax of Terms

- Every individual constant, every variable, every arbitrary individual, and every undetermined individual is a term.
- If f^n is a function symbol of arity n, and $t_1, \ldots, t_n$ are terms, then $f^n(t_1, \ldots, t_n)$ is a (functional) term.
  (The superscript may be omitted if no confusion results.)
  For example: $\mathit{Drives}^2(\mathit{Tom}, \mathit{mother\_of}^1(\mathit{Betty}))$.
- Nothing else is a term.

Syntax of Atomic Formulas

- If P^n is a predicate symbol of arity n, and $t_1, \ldots, t_n$ are terms, then $P^n(t_1, \ldots, t_n)$ is an atomic formula.

For example, *child-in*^{2}*(Sally, family-of*^{2}*(John, mother-of*^{1}*(Sally)))* (The superscript may be omitted if no confusion results.)

Syntax of Well-Formed Formulas (WFFs)

- Every atomic formula is a wff.
- If P is a wff, then so is $\neg P$.
- If P and Q are wffs, then so are

(a)	$(P \wedge Q)$	(b)	$(P \vee Q)$
(c)	$(P \Rightarrow Q)$	(d)	$(P \Leftrightarrow Q)$

- If P is a wff and x is a variable, then $\forall x(P)$ and $\exists x(P)$ are wffs. $\forall$ is called the **universal quantifier**. $\exists$ is called the **existential quantifier**. P is called the **scope** of quantification.
- Nothing else is a wff.

We will not bother using parentheses when there is no ambiguity, in which case $\forall$ and $\exists$ will have the highest priority, then $\wedge$ and $\vee$ will have higher priority than $\Rightarrow$, which, in turn will have higher priority than $\Leftrightarrow$. For example, we will write $\forall x P(x) \wedge \exists y Q(y) \Leftrightarrow \neg P(a) \Rightarrow Q(b)$ instead of $((\forall x(P(x)) \wedge \exists y(Q(y))) \Leftrightarrow (\neg P(a) \Rightarrow Q(b)))$.

Every occurrence of x in P, not in the scope of some occurrence of $\forall x$ or $\exists x$, is said to be **free** in P and **bound** in $\forall x P$ and $\exists x P$. Every occurrence of every variable other than x that is free in P is also free in $\forall x P$ and $\exists x P$.

A wff with at least one free variable is called **open**. A wff with no free variables is called **closed**, An expression with no variables, arbitrary individuals, or undetermined individuals is called **ground**.

5.2.1 Syntax of FOPL CarPool World

In FOPL CarPool World, we will use the following atomic symbols:

Individual constants: *Betty*, *Tom*

Variables: x, y

Arbitrary Individuals: *anyone*

Undetermined Individuals: *someone*

Unary predicate symbols: *Driver*, *Passenger*

Binary predicate symbol: *Drives*

In FOPL CarPool World, the six atomic propositions of Propositional CarPool World become the six wffs:

$$\begin{array}{ll} Drives(Betty, Tom) & Drives(Tom, Betty) \\ Driver(Betty) & Driver(Tom) \\ Passenger(Betty) & Passenger(Tom) \end{array}$$

5.3 Substitutions

A **substitution** is a set of **pairs,** $\{t_1/v_1, \ldots, t_n/v_n\}$ where the t_i are terms, the v_i are variables, arbitrary individuals, or undetermined individuals, and $\forall i, j[i \neq j \Rightarrow v_i \neq v_j]$. The result of **applying** a substitution to a wff A, written $A\{t_1/v_1, \ldots, t_n/v_n\}$ is obtained by *simultaneously* replacing every occurrence of each arbitrary or undetermined individual v_i in A by t_i, and every free occurrence of each variable v_j in A by t_j, as long as any variable that occurs in t_i or t_j remains free in the result.

Examples

- $P(x, y)\{x/y, y/x\} = P(y, x)$
- $(Drives(x, someone) \Rightarrow \exists x Passenger(x))\{Betty/x,\ Tom/someone\}$
  $= (Drives(Betty,\ Tom) \Rightarrow \exists x Passenger(x))$

5.4 Semantics

Although the intensional semantics of a FOPL depends on the domain being formalized, and the extensional semantics depends also on a particular situation, we can specify the types of the entities usually given as the intensional and as the extensional semantics of FOPL expressions.

5.4.1 Semantics of the "Standard" Predicate Logic

The usual semantics of FOPL assumes a **Domain**, $\mathcal{D}$, of individuals, functions on individuals, sets of individuals, and relations on individuals. Let $\mathcal{I}$ be the set of all individuals in the domain $\mathcal{D}$.

Semantics of Atomic Symbols

Individual Constants:
If a is an individual constant, $[\![a]\!]$ is some particular individual in $\mathcal{I}$.

Function Symbols:
If f^n is a function symbol of arity n, $[\![f^n]\!]$ is some particular function in $\mathcal{D}$,
$[\![f^n]\!]$: $\underbrace{\mathcal{I} \times \cdots \times \mathcal{I}}_{n \text{ times}} \rightarrow \mathcal{I}$

Predicate Symbols:

- If P^1 is a unary predicate symbol, $[\![P^1]\!]$ is some particular subset of $\mathcal{I}$.
- If P^n is a predicate symbol of arity n, $[\![P^n]\!]$ is some particular subset of the relation

$$\underbrace{\mathcal{I} \times \cdots \times \mathcal{I}}_{n \text{ times}}.$$

Semantics of Ground Terms

Individual Constants:
If a is an individual constant, $[\![a]\!]$ is some particular individual in $\mathcal{I}$.

Functional Terms:
If f^n is a function symbol of arity n, and $t_1, \ldots, t_n$ are ground terms, then $[\![f^n(t_1, \ldots, t_n)]\!] = [\![f^n]\!]([\![t_1]\!], \ldots, [\![t_n]\!])$.

Semantics of Ground Atomic Formulas

- If P^1 is a unary predicate symbol, and t is a ground term, then $[\![P^1(t)]\!]$ is True if $[\![t]\!] \in [\![P^1]\!]$, and False otherwise.
- If P^n is an n-ary predicate symbol, and $t_1, \ldots, t_n$ are ground terms, then $[\![P^n(t_1, \ldots, t_n)]\!]$ is True if $\langle [\![t_1]\!], \ldots, [\![t_n]\!] \rangle \in [\![P^n]\!]$, and False otherwise.

Semantics of WFFs

- If P is a ground wff, then $[\![\neg P]\!]$ is True if $[\![P]\!]$ is False, otherwise, it is False.
- If P and Q are ground wffs, then $[\![P \wedge Q]\!]$ is True if $[\![P]\!]$ is True and $[\![Q]\!]$ is True, otherwise, it is False.
- If P and Q are ground wffs, then $[\![P \vee Q]\!]$ is False if $[\![P]\!]$ is False and $[\![Q]\!]$ is False, otherwise, it is True.
- If P and Q are ground wffs, then $[\![P \Rightarrow Q]\!]$ is False if $[\![P]\!]$ is True and $[\![Q]\!]$ is False, otherwise, it is True.
- If P and Q are ground wffs, then $[\![P \Leftrightarrow Q]\!]$ is True if $[\![P]\!]$ and $[\![Q]\!]$ are both True or both False, otherwise, it is False.
- If P is a wff containing the arbitrary individual x, then $[\![P]\!]$ is True if $[\![P\{t/x\}]\!]$ is True for every ground term, t. Otherwise, it is False.
- If P is a wff containing the undetermined individual x, then $[\![P]\!]$ is True if there is some ground term, t such that $[\![P\{t/x\}]\!]$ is True. Otherwise, it is False.
- $[\![\forall x P]\!]$ is True if $[\![P\{t/x\}]\!]$ is True for every ground term, t. Otherwise, it is False.
- $[\![\exists x P]\!]$ is True if there is some ground term, t such that $[\![P\{t/x\}]\!]$ is True. Otherwise, it is False.

Recall that the intensional semantics of an expression can be given as an English statement that allows the extensional value to be determined in any specific situation.

In this presentation, I do not give semantics to non-ground expressions. Some people do, but I think that is confusing.

5.4.2 Intensional Semantics of FOPL CarPool World

The intensional semantics of the atomic symbols of FOPL CarPool World are:

Individual constants:

Betty: The individual named Betty.

Tom: The individual named Tom.

Unary predicate symbols:

Driver: The set of drivers on a given day.

Passenger: The set of passengers on a given day.

Binary predicate symbol:

Drives: The relation that holds between a driver and a passenger when the former drives the latter to work on a given day.

The intensional semantics of other ground expressions in FOPL CarPool World can be derived from these according to the format of the previous subsection.

5.4.3 Extensional Semantics of FOPL CarPool World

The extensional semantics of some expressions in FOPL CarPool World in four different situations (that is, on four different days) are:

	Denotation in Situation (Day)			
Expression	1	2	3	4
Driver	{Betty, Tom}	{Betty}	{Tom}	{}
Passenger	{Betty, Tom}	{Tom}	{Betty}	{}
Drives	{⟨ Betty, Tom ⟩, ⟨ Tom, Betty ⟩}	{⟨ Betty, Tom ⟩}	{⟨ Tom, Betty ⟩}	{}
Driver(Betty)	True	True	False	False
Driver(Tom)	True	False	True	False
Passenger(Betty)	True	False	True	False
Passenger(Tom)	True	True	False	False
Drives(Betty, Tom)	True	True	False	False
Drives(Tom, Betty)	True	False	True	False

5.4.4 Semantic Properties of WFFs

Just as for wfps, a wff is **satisfiable** if it is True in at least one situation, **contingent** if it is True in at least one situation and False in at least one situation, **valid** if it is True in every situation, and **contradictory** of it is False in every situation. The terms **tautology** and **logical implication** are also defined for wffs as they are for wfps, as is the notation $\models_{\mathcal{L}} A$ and $A_1, \ldots, A_n \models_{\mathcal{L}} B$. Metatheorem 1 also applies to FOPL as well as to Propositional Logic.

In a logical language without function symbols, with n Individual Constants and k_j predicates of arity j,there are $\sum_j (k_j \times n^j)$ ground atomic propositions, and $2^{\sum_j (k_j \times n^j)}$ situations. In CarPool World, this comes to $2^{2\times 2^1 + 1\times 2^2} = 256$ situations. If we add even one function symbol, we get an infinite number of ground terms, and, therefore, an infinite number of situations. For example, if we add the unary function symbol *mother_of* to CarPool World, we get the ground terms, *mother_of(Tom)*, *mother_of(mother_of(Tom))*, *mother_of(mother_of(mother_of(Tom)))*, ...

5.4.5 Domain Rules of FOPL CarPool World

In Propositional CarPool World five sentences were needed to constrain the situations to the two common-sense ones. In FOPL CarPool World, only three **domain rules** are needed:

- Each person is the driver or the passenger, but not both:

$$\forall x(\mathit{Driver}(x) \Leftrightarrow \neg \mathit{Passenger}(x))$$

- If one person drives the other, then the former is the driver and the latter is the passenger:

$$\forall x \forall y(\mathit{Drives}(x, y) \Rightarrow \mathit{Driver}(x) \wedge \mathit{Passenger}(y))$$

- And again, either Tom drives Betty or Betty drives Tom:

$$\mathit{Drives}(\mathit{Tom}, \mathit{Betty}) \vee \mathit{Drives}(\mathit{Betty}, \mathit{Tom})$$

One can (relatively) easily verify that, of the 256 situations of FOPL CarPool World, only Situations 2 and 3 above make all these three wffs True.

5.5 Inference

Inference in Predicate Logics is just like inference in Propositional Logics, with the addition of axioms and/or rules of inference for the universal and existential quantifiers. For the Natural Deduction system given above, the additional rules of inference are

∀I If $\Gamma \vdash A\{t/x\}$, where t is any arbitrary individual that does not occur in Γ, then $\Gamma \vdash \forall x A$.

∀E If $\Gamma \vdash \forall x A$ then $\Gamma \vdash A\{t/x\}$, where t is any term.

∃I If $\Gamma \vdash A\{t/x\}$, where t is any term, then $\Gamma \vdash \exists x A$.

∃E If $\Gamma \vdash \exists x A$ then $\Gamma \vdash A\{t/x\}$, where t is any undetermined individual that does not occur in Γ.

Confusingly, each of these four rules commonly goes by two different names:

Abbreviation	Our Name	Other Common Name
∀I	Universal Introduction	Universal Generalization
∀E	Universal Elimination	Universal Instantiation
∃I	Existential Introduction	Existential Generalization
∃E	Existential Elimination	Existential Instantiation

Metatheorem 2 applies to FOPL as well as it does to Propositional Logic.

5.5.1 Example Proof

In this example proof of the theorem $\neg\forall x A(x) \Rightarrow \exists x \neg A(x)$, the following atomic symbols are used:

Variable: x

Arbitrary individual: b

Predicate symbol: A

1.	$\neg\forall x A(x)$	$\vdash$	$\neg\forall x A(x)$	$Axiom$
2.	$\neg\exists x \neg A(x)$	$\vdash$	$\neg\exists x \neg A(x)$	$Axiom$
3.	$\neg A(b)$	$\vdash$	$\neg A(b)$	$Axiom$
4.	$\neg A(b)$	$\vdash$	$\exists x \neg A(x)$	$\exists I, 3$
5.	$\neg\exists x \neg A(x), \neg A(b)$	$\vdash$	$\exists x \neg A(x), \neg\exists x \neg A(x)$	$Cut, 2, 4$
6.	$\neg\exists x \neg A(x)$	$\vdash$	$\neg\neg A(b)$	$\neg I, 5$
7.	$\neg\exists x \neg A(x)$	$\vdash$	$A(b)$	$\neg E, 6$
8.	$\neg\exists x \neg A(x)$	$\vdash$	$\forall x A(x)$	$\forall I, 7$
9.	$\neg\forall x A(x), \neg\exists x \neg A(x)$	$\vdash$	$\forall x A(x), \neg\forall x A(x)$	$Cut, 1, 8$
10.	$\neg\forall x A(x)$	$\vdash$	$\neg\neg\exists x \neg A(x)$	$\neg I, 9$
11.	$\neg\forall x A(x)$	$\vdash$	$\exists x \neg A(x)$	$\neg E, 10$
12.		$\vdash$	$\neg\forall x A(x) \Rightarrow \exists x \neg A(x)$	$\Rightarrow I, 11$

5.5.2 Example Derivation

As an example, I'll present the FOPL version of the derivation from Section 4.4.1 of

$$Drives(Tom, Betty) \Rightarrow \neg Drives(Betty, Tom)$$

from the CarPool World domain rules. To save space, I'll use the following abbreviations.

$B:$	$Betty$
$T:$	Tom
$Dr:$	$Driver$
$Pr:$	$Passenger$
$D:$	$Drives$

1.	$D(T,B) \vdash D(T,B)$	*Axiom*
2.	$\forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y))$	*Axiom*
3.	$\forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash \forall y(D(T,y) \Rightarrow Dr(T) \wedge Pr(y))$	$\forall E, 2$
4.	$\forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash D(T,B) \Rightarrow Dr(T) \wedge Pr(B)$	$\forall E, 3$
5.	$D(T,B), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash D(T,B), D(T,B) \Rightarrow Dr(T) \wedge Pr(B)$	*Cut, 1, 4*
6.	$D(T,B), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Dr(T) \wedge Pr(B)$	$\Rightarrow E, 5$
7.	$D(T,B), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Dr(T)$	$\wedge E, 6$
8.	$\forall x(Dr(x) \Rightarrow \neg Pr(x)) \vdash \forall x(Dr(x) \Rightarrow \neg Pr(x))$	*Axiom*
9.	$\forall x(Dr(x) \Rightarrow \neg Pr(x)) \vdash Dr(T) \Rightarrow \neg Pr(T)$	$\forall E, 8$
10.	$D(T,B), \forall x(Dr(x) \Rightarrow \neg Pr(x)), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Dr(T), Dr(T) \Rightarrow \neg Pr(T)$	*Cut, 7, 9*
11.	$D(T,B), \forall x(Dr(x) \Rightarrow \neg Pr(x)), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash \neg Pr(T)$	$\Rightarrow E, 10$
12.	$D(B,T) \vdash D(B,T)$	*Axiom*
13.	$\forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash \forall y(D(B,y) \Rightarrow Dr(B) \wedge Pr(y))$	$\forall E, 2$
14.	$\forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash D(B,T) \Rightarrow Dr(B) \wedge Pr(T))$	$\forall E, 13$
15.	$D(B,T), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash D(B,T), D(B,T) \Rightarrow Dr(B) \wedge Pr(T))$	*Cut, 12, 14*
16.	$D(B,T), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Dr(B) \wedge Pr(T)$	$\Rightarrow E, 15$
17.	$D(B,T), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Pr(T)$	$\wedge E, 16$
18.	$D(B,T), D(T,B), \forall x(Dr(x) \Rightarrow \neg Pr(x)), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash Pr(T), \neg Pr(T)$	*Cut, 10, 17*
19.	$D(T,B), \forall x(Dr(x) \Rightarrow \neg Pr(x)), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash \neg D(B,T)$	$\neg I, 18$
20.	$\forall x(Dr(x) \Rightarrow \neg Pr(x)), \forall x \forall y(D(x,y) \Rightarrow Dr(x) \wedge Pr(y)) \vdash D(T,B) \Rightarrow \neg D(B,T)$	$\Rightarrow I, 19$

Again, the final domain rule may be added by the rule of *Hyp*.

6 Some Properties of Logics

Three important properties of logics are **soundness**, **consistency** and **completeness**.

- A logic $\mathcal{L}$ is **sound** if $\Gamma \vdash_{\mathcal{L}} A$ implies $\Gamma \models_{\mathcal{L}} A$. This also implies that $\vdash_{\mathcal{L}} A$ implies $\models_{\mathcal{L}} A$. That is, every theorem is valid.
- A logic $\mathcal{L}$ is **consistent** if there is no wff A for which both $\vdash_{\mathcal{L}} A$ and $\vdash_{\mathcal{L}} \neg A$. That is, there is no wff such that both it and its negation are provable. If at most one of A and $\neg A$ can be valid, soundness implies consistency.
- A logic $\mathcal{L}$ is **complete** if, for every wff A, $\models_{\mathcal{L}} A$ implies $\vdash_{\mathcal{L}} A$. That is, every valid wff is provable.

It is the essence of what we mean by correct reasoning that any logic we use be sound. Completeness, however, is less important because it says nothing about how long a proof will take. We might give up on a proof that is taking too long, and then we will not know if the wff we were trying to prove is a theorem or not. Nevertheless, all propositional logics, as well as FOPL CarPool World both sound and complete. The famous Gödel Incompletness Theorem says that any formal system that is strong enough to represent arithmetic is either inconsistent or incomplete. However, none of these logics is that strong.

7 Type Theory and Higher-Order Logics

Type theory was developed by Bertrand Russell (Russell, 1908) in order to ban such paradoxical formulas as

$$\exists x \forall y[x(y) \Leftrightarrow \neg y(y)]$$

which may be read as

> *There is a property, x, which holds of every property, y, that does not hold of itself.*

or as

There is a set, x, that contains all sets that don't contain themselves.

Note that an instance of this formula is

$$R(R) \Leftrightarrow \neg R(R)$$

That is, the Russell Set is a member of itself if and only if it isn't a member of itself.

In type theory, terms denoting individuals are assigned the type 0. A predicate symbol, P, may take terms and other predicate symbols as arguments, but there must be some positive integer, i such that P takes at least one argument that is of type i and no argument of type greater than i. P is then assigned the type $i+1$. n^{th}-Order Predicate Logic allows the use of predicate symbols of type at most n, and allows variables to range over terms of type at most $n-1$. Thus, both

$$\exists R[Kinship(R) \land R(Lou, Stu)]$$

and

$$\forall R[Symmetric(R) \Leftrightarrow \forall x \forall y (R(x,y) \Leftrightarrow R(y,x))]$$

are second-order formulas. Ω-Order Predicate Logic does not limit the size of n, but still requires that a predicate of type i take one argument of type $i-1$ and no argument of type i or greater. So the Russell Paradox is not well-formed in n^{th}-Order Predicate Logic for any n.

Acknowledgments

The material in this Appendix is based on the tutorial, "Foundations of Logic & Inference" given at the 14th International Joint Conference on Artificial Intelligence, August 20, 1995, and the course, Reasoning in Artificial Intelligence, given at the First International Summer Institute on Cognitive Science (FISI), University at Buffalo, Buffalo, NY, July, 1994.

References

Fitch, F. B. (1952). *Symbolic Logic: An Introduction.* Ronald Press, New York.

Haack, S. (1978). *Philosophy of Logics.* Cambridge University Press, New York.

Kleene, S. C. (1950). *Introduction to Metamathematics.* D. Van Nostrand, Princeton, NJ.

McCawley, J. D. (1981). *Everything that Linguists have Always Wanted to Know about Logic* *but were ashamed to ask.* The University of Chicago Press, Chicago.

Rapaport, W. J. (1992). Logic. In Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, pages 851–853. John Wiley & Sons, New York, second edition.

Russell, B. (1908). Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222–262.

Shapiro, S. C., editor (1992). *Encyclopedia of Artificial Intelligence.* John Wiley & Sons, New York, second edition.