Logic Background (1C)

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice and Octave.

Syntactic Consequence

Syntactic consequence within a formal system

A **formula A** is a syntactic consequence
within some formal system ***FS*** of a set **Γ** of **formulas**
if there is a **derivation** in formal system ***FS*** of **A** from the set **Γ**.

$\Gamma \vdash_{FS} A$

Syntactic consequence does not depend
on any **interpretation** of the formal system.

.

Interpretations

An interpretation of a formal system is
the **assignment** of **meanings** to the **symbols**,
and **truth values** to the **sentences** of a formal system.

The study of interpretations is called **formal semantics**.

Giving an **interpretation** is synonymous with
constructing a **model**.

An interpretation is expressed in a metalanguage,
which may itself be a formal language,
and as such itself is a syntactic entity.

Satisfiability and Validity

In mathematical logic, satisfiability and validity are elementary concepts of semantics.

A **formula** is **satisfiable** if it is possible to find **an** interpretation (model) that makes the formula true. **some** S are P

A **formula** is **valid** if **all** interpretations make the formula true. **every** S is a P

A **formula** is **unsatisfiable** if **none** of the interpretations make the formula true. **no** S are P

A **formula** is **invalid** if some such interpretation makes the formula false. **some** S are **not** P

a **theory** is **satisfiable** if **one** of the interpretations makes each of the axioms of the theory **true**.

a **theory** is **valid** if **all** of the interpretations make each of the axioms of the theory **true**.

a **theory** is **unsatisfiable** if **all** of the interpretations make each of the axioms of the theory **false**.

a **theory** is **invalid** if **one** of the interpretations makes each of the axioms of the theory **false**.

http://en.wikipedia.org/wiki/

Reduction of Validity to Unsatisfiability

For classical logics,

can reexpress the **validity** of a formula to **satisfiability**,

because of the relationships between the concepts expressed in the square of opposition.

In particular φ is **valid** if and only if ¬φ is **unsatisfiable**,

which is to say it is not true that ¬φ is satisfiable.

Put another way, φ is **satisfiable** if and only if ¬φ is **invalid**.

http://en.wikipedia.org/wiki/

Sound, Complete

There are many deductive systems for first-order logic that are sound (all provable statements are true in all models) and complete (all statements which are true in all models are provable). Although the logical consequence relation is only semidecidable, much progress has been made in automated theorem proving in first-order logic. First-order logic also satisfies several metalogical theorems that make it amenable to analysis in proof theory, such as the Löwenheim–Skolem theorem and the compactness theorem.

Sound – all *provable* statements are **true** in all models

Complete – all statements which are *true* in all models are **provable**

http://en.wikipedia.org/wiki/

Syntactic Completeness

Syntactic completeness of a formal system

A formal system ***S*** is **syntactically complete**
iff for each **formula A** of the language of the system
either A or ¬A is a **theorem** of ***S***.

In another sense, a formal system is **syntactically complete**
iff no **unprovable axiom** can be added to it
as an **axiom** without introducing an inconsistency.

Truth-functional **propositional logic**
and **first-order predicate logic** are
semantically complete,
but **not syntactically complete**

(for example the **propositional logic** statement
consisting of a single variable "a" is not a theorem,
and neither is its negation, but these are not tautologies).

deductively complete,
maximally complete,
negation complete
complete

Semantic Completeness

from en.wikipedia.org

In logic, semantic completeness is the converse of soundness for formal systems.

a tautology (from the Greek word ταυτολογία) is a formula which is true in every possible interpretation.

A formal system is "**semantically complete**" when all its **tautologies** are **theorems**

A formal system is "**sound**" when all **theorems** are **tautologies**

semantically complete

every tautology → *theorem*

sound

every theorem → *tautology*

(that is, they are **semantically valid formulas**: formulas that are true under every interpretation of the language of the system that is **consistent** with the rules of the system).

A formal system is **consistent**
if for all formulas φ of the system,
the formulas φ and ¬φ (the negation of φ)
are not both theorems of the system
(that is, they cannot be both proved with the rules of the system).

http://en.wikipedia.org/wiki/

Premise

A **premise** : an assumption that something is true.

an **argument** requires

a set of (at least) **two** declarative sentences ("propositions") known as the **premises**

along with **another** declarative sentence ("proposition") known as the **conclusion**.

two premises and **one conclusion** :
the basic **argument** structure

From Middle English, from Old French premisse, from Medieval Latin premissa ("set before") (premissa propositio ("the proposition set before")), feminine past participle of Latin praemittere ("to send or put before"), from prae- ("before") + mittere ("to send").

Because all men are mortal and Socrates is a man,
Socrates is mortal.

2 premises
1 conclusion

3 propositions

Valid Argument and Valid Formula

an **argument** is **valid** if and only if
it takes a form that makes it impossible
for the **premises** to be **true** and
the **conclusion** nevertheless to be **false**.

It is not required for a **valid argument**
to have **premises** that are actually **true**,
but to have premises that,
if they were true, would guarantee
the truth of the argument's conclusion.

A **formula** is **valid** if and only if
it is **true** under **every interpretation**,

an **argument form** (or schema) is **valid** if and only if
every argument of that **logical form** is **valid**.

the premises true ➡
the conclusion true

Logical Form

A **logical form** of a syntactic expression is
a precisely-specified **semantic version** of
that expression in a formal system

Informally, the **logic form** attempts
to formalize a possibly ambiguous statement
into a statement with a precise, unambiguous
logical interpretation
with respect to a unambiguously from syntax alone.

In an ideal formal language, the **meaning** of a **logical form**
can be determined unambiguously from **syntax alone**

Logical forms are **semantic**, not **syntactic** constructs, therefore, there may
be more than one string that represents the same logical form in a given
language

Logical Form Example

Original argument

All humans are mortal
Socrates is human
Therefore, Socrates is mortal

Argument form

All **H** are **M**
S is **H**
Therefore, **S** is **M**

Valid Argument Forms (Propositional)

Modus ponens (MP) If A, then B A Therefore, B	**Hypothetical syllogism (HS)** If A, then B If B, then C Therefore, if A, then C
Modus tollens (MT) If A, then B Not B Therefore, not A	**Disjunctive syllogism (DS)** A or B Not A Therefore, B

Modus ponens
(Latin) “the way that affirms by affirming"

Modus tollens
(Latin) "the way that denies by denying"

Syllogism
(Greek: συλλογισμός syllogismos) – "conclusion," "inference"

Modus Ponens

The Prolog resolution algorithm
based on the modus ponens form of inference

a general **rule** – the major premise and
a specific **fact** – the minor premise

All men are mortal	rule
Socrates is a man	fact
Socrates is mortal	

modus ponendo ponens
(Latin) “the way that affirms by affirming";
often abbreviated to **MP** or **modus ponens**

P implies Q;
P is asserted to be true,
so therefore Q must be true

one of the accepted mechanisms for the construction of deductive proofs
that includes the "rule of definition" and the "rule of substitution"

Facts	a	a
Rules	$a \rightarrow b$	b :- a
Conclusion	b	b

Facts	man(’Socrates’).
Rules	mortal(X) :- man(X).
Conclusion	mortal(’Socrates’).

Modus Ponens (revisited)

Facts	a	a	minor term
Rules	a → b	b :- a	
Conclusion	b	b	major term

Syllogism : etymology

syllogism (*plural* **syllogisms**)

1. (*logic*) An inference in which one proposition (the conclusion) follows necessarily from two other propositions, known as the premises. [quotations ▼]
1. (*obsolete*) A trick, artifice.

Etymology [edit]

Wikipedia has an article
syllogism

From Old French *silogisme* ("syllogism"), from Latin *syllogismus*, from Ancient Greek συλλογισμός (*sullogismós*, "inference, conclusion").

http://en.wikipedia.org/wiki/

Syllogism

A syllogism (Greek: συλλογισμός – syllogismos – "conclusion," "inference") is

a kind of **logical argument** that applies deductive reasoning
to arrive at a conclusion based on two or more **propositions**
that are asserted or assumed to be true.

In its earliest form, defined by Aristotle,
from the combination of
a general statement (the major premise) and ⟺ rule
a specific statement (the minor premise), ⟺ fact
a conclusion is deduced.

For example, knowing
that all men are mortal (major premise) and ⟺ rule
that Socrates is a man (minor premise), ⟺ fact
we may validly conclude that Socrates is mortal.

Syllogism – major & minor terms

A categorical syllogism consists of three parts:

Major premise:	All humans are mortal.	⟺	major term	(the predicate of the conclusion)
Minor premise:	All Greeks are humans.	⟺	minor term	(the subject of the conclusion)
Conclusion:	All Greeks are mortal.			

Syllogism – Categorical Propositions

Each part - a categorical proposition - two categorical terms

In Aristotle, each of the premises is in the form

"All A are B"	universal proposition
"Some A are B"	particular proposition
"No A are B"	universal proposition
"Some A are not B"	particular proposition

Syllogism – common terms

each of the premises has one term
in common with the conclusion:

this common term is called

a major term in a major premise (the predicate of the conclusion)
a minor term in a minor premise (the subject of the conclusion)

Mortal is the major term,
Greeks is the minor term.
Humans is the middle term

Major premise:	All humans are mortal.
Minor premise:	All Greeks are humans.
Conclusion:	All Greeks are mortal.

Soundness

An argument is **sound** if and only if

- The argument is **valid**.
- All of its premises are **true**.

http://en.wikipedia.org/wiki/

Sound Argument Example

All men are mortal.	(true)
Socrates is a man.	(true)
Therefore, Socrates is mortal.	(sound)

The argument is **valid**
because the conclusion is true
based on the premises,
that is, that the conclusion follows the premises

since the premises are in fact true,
the argument is **sound**.

http://en.wikipedia.org/wiki/

Non-sound Argument Example

The following argument is valid but not sound:

All organisms with wings can fly.	(false)
Penguins have wings.	(true)
Therefore, penguins can fly.	(valid)

Since the first premise is actually false,
the argument, though valid, is not **sound**.

http://en.wikipedia.org/wiki/

Soundness and Completeness

The crucial properties of this set of rules are
that they are sound and complete.

Informally this means
that the rules are correct
and that no other rules are required.

http://en.wikipedia.org/wiki/

Derivation

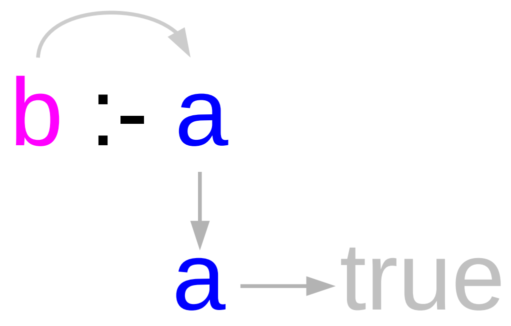
A **reversed modus ponens** is used in Prolog

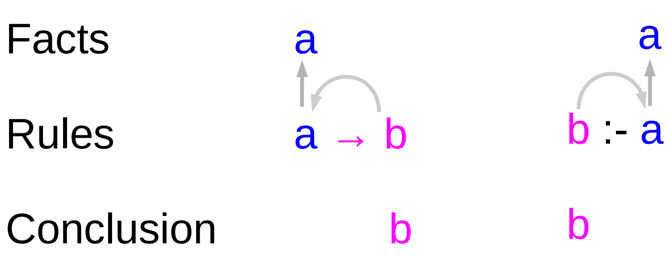
Prolog tries to prove that
a query (b) is a consequence of
the database content (a, a ⇒ b).

Using the major premise, it goes from b to a,
and using the minor premise, from a to true.

Such a sequence of goals is called a **derivation**.

A derivation can be **finite** or **infinite**.





Facts	a
Rules	b :- a
Conclusion	b

Horn Clause

A **Horn clause** with exactly one **positive literal** is
a **definite clause** or a **strict Horn clause**;

a **definite clause** with no **negative literals** is
sometimes called a **unit clause**,

and a **unit clause** without **variables** is
sometimes called a **fact**;

and a **Horn clause** without a **positive literal** is
sometimes called a **goal clause**

(note that the empty **clause** consisting of **no literals** is a **goal clause**).

Horn Clause

	Disjunction form	Implication form	Read intuitively as
Definite clause	**¬p ∨ ¬q ∨ ... ∨ ¬t ∨ u**	**u ← p ∧ q ∧ ... ∧ t**	assume that, if p and q and ... and t all hold, then also u holds
Fact	**u**	**u**	assume that, u holds
Goal clause	**¬p ∨ ¬q ∨ ... ∨ ¬t**	**false ← p ∧ q ∧ ... ∧ t**	show that p and q and ... and t all hold

a **definite clause** a **Horn clause** with exactly one **positive literal** is
a **unit clause** a **definite clause** with no **negative literals**
a **fact** a **unit clause** without **variables** is
a **goal clause** a **Horn clause** without a **positive literal**

Horn Clause

In the **non-propositional** case,
all variables in a clause are implicitly **universally quantified** with the scope being the entire clause

¬ human(X) ∨ mortal(X)

stands for:

∀X(¬ human(X) ∨ mortal(X))

which is logically equivalent to:

∀X (human(X) → mortal(X))

Resolution (1)

the **resolution rule** in propositional logic is
a single valid inference rule that produces a new clause
implied by two clauses containing **complementary literals**.

A **literal** is a propositional variable or
the negation of a propositional variable.

Two literals are said to be **complements**
if one is the negation of the other
(in the following, $\neg c$ is taken to be the complement to c

Resolution (2)

The resulting clause contains all the literals that do not have complements. Formally:

$$\frac{a_1 \lor a_2 \lor \cdots \lor c \,,\; b_1 \lor b_2 \lor \cdots \lor \neg c}{a_1 \lor a_2 \lor \cdots \lor b_1 \lor b_2 \lor \cdots}$$

all a_i, b_i, and c are literals,
the dividing line stands for "entails".

Resolvent

The resulting clause contains all the literals
that do not have complements. Formally:

$$\frac{a_1 \lor a_2 \lor \cdots \lor c\ ,\ b_1 \lor b_2 \lor \cdots \lor \neg c}{a_1 \lor a_2 \lor \cdots \lor b_1 \lor b_2 \lor \cdots}$$

all a_i, b_i, and c are literals,
the dividing line stands for "entails".

The above may also be written as:

$$\frac{(\neg a1 \land \neg a2 \land \cdots) \rightarrow c\ ,\ c \rightarrow (b1 \lor b2 \lor \cdots)}{(\neg a1 \land \neg a2 \land \cdots) \rightarrow (b1 \lor b2 \lor \cdots)}$$

The clause produced by the resolution rule is called
the **resolvent** of the two input clauses.
It is the principle of consensus applied to clauses rather than terms

Resolvent

When the two clauses contain
more than one pair of complementary literals,
the resolution rule can be applied
(independently) for each such pair;
however, the result is always a tautology.

Modus ponens can be seen
as a special case of resolution
(of a one-literal clause and a two-literal clause).

$$\frac{p \rightarrow q \;,\; p}{q}$$

is equivalent to

$$\frac{\neg p \lor q \;,\; p}{q}$$

Horn Clause (1)

the **resolvent** of **two Horn clauses**
is itself **a Horn clause**
the **resolvent** of **a goal clause** and
a definite clause
is **a goal clause**

These properties of Horn clauses can lead
to greater efficiencies in proving a theorem
(represented as the negation of a goal clause).

Horn Clause (2)

Propositional Horn clauses are also of interest
in computational complexity,

the problem of finding truth value assignments
to make a conjunction of **propositional Horn clauses** true
is a **P-complete** problem (in fact solvable in linear time),
sometimes called **HORNSAT**.

The **unrestricted Boolean satisfiability** problem is
an **NP-complete** problem however.

Satisfiability of **first-order Horn clauses** is undecidable.

Horn Clause (3)

By iteratively applying the resolution rule, it is possible
to tell whether a propositional formula is **satisfiable**
to prove that a first-order formula is unsatisfiable;

this method may prove the **satisfiability** of a first-order formula,
but not always, as it is the case for all methods for first-order logic

Turnstile

In mathematical logic and computer science the symbol $\vdash$ has taken the name **turnstile** because of its resemblance to a typical turnstile if viewed from above. It is also referred to as **tee** and is often read as "yields", "proves", "satisfies" or "entails". The symbol was first used by Gottlob Frege in his 1879 book on logic, *Begriffsschrift*.[1]

Martin-Löf analyzes the $\vdash$ symbol thus: "...[T]he combination of Frege's Urteilsstrich, judgement stroke [|], and Inhaltsstrich, content stroke [—], came to be called the assertion sign."[2] Frege's notation for a judgement of some content A

$$\vdash A$$

can be then be read

I know A is true".[3]

In the same vein, a conditional assertion

$$P \vdash Q$$

can be read as:

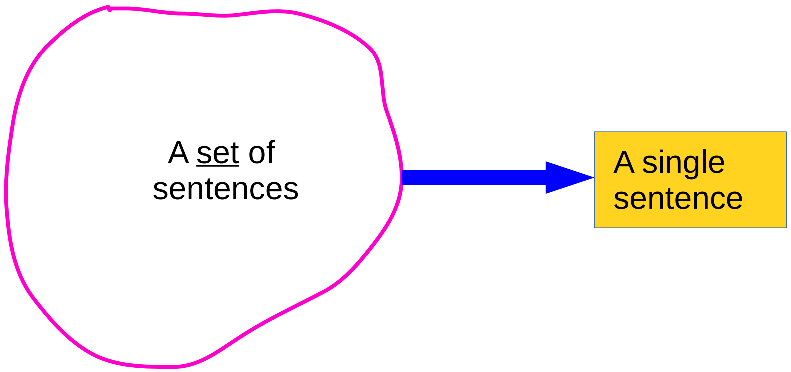
From P, I know that Q

In logic, the symbol $\vDash$, $\models$ or $\models$ is called the **double turnstile**. It is closely related to the turnstile symbol $\vdash$, which has a single bar across the middle. It is often read as "entails", "models", "is a semantic consequence of" or "is stronger than".[1] In TeX, the turnstile symbols $\vDash$ and $\models$

http://en.wikipedia.org/wiki/

Double Turnstile

In logic, the symbol ⊨, ⊨ or ⊨ is called the **double turnstile**. It is closely related to the turnstile symbol ⊢, which has a single bar across the middle. It is often read as "entails", "models", "is a semantic consequence of" or "is stronger than".[1] In TeX, the turnstile symbols ⊨ and ⊨



The double turnstile is a binary relation. It has several different meanings in different contexts:

- To show semantic consequence, with a set of sentences on the left and a single sentence on the right, to denote that if every sentence on the left is true, the sentence on the right must be true, e.g. $\Gamma \models \varphi$. This usage is closely related to the single-barred turnstile symbol which denotes syntactic consequence.
- To show satisfaction, with a model (or truth-structure) on the left and a set of sentences on the right, to denote that the structure is a model for (or satisfies) the set of sentences, e.g. $\mathcal{A} \models \Gamma$.
- To denote a tautology, $\models \varphi$. which is to say that the expression φ is a semantic consequence of the empty set.

http://en.wikipedia.org/wiki/

References

[1] http://en.wikipedia.org/
[2]