First Order Logic – Implication (4A)

Young W. Lim
5/30/17

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

This document was produced by using LibreOffice

Based on

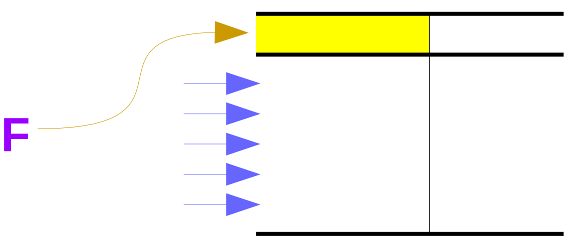
Contemporary Artificial Intelligence,
R.E. Neapolitan & X. Jiang

Logic and Its Applications,
Burkey & Foxley

PL: A Model

A **model** or a **possible world**:

Every atomic proposition is assigned a value **T** or **F**

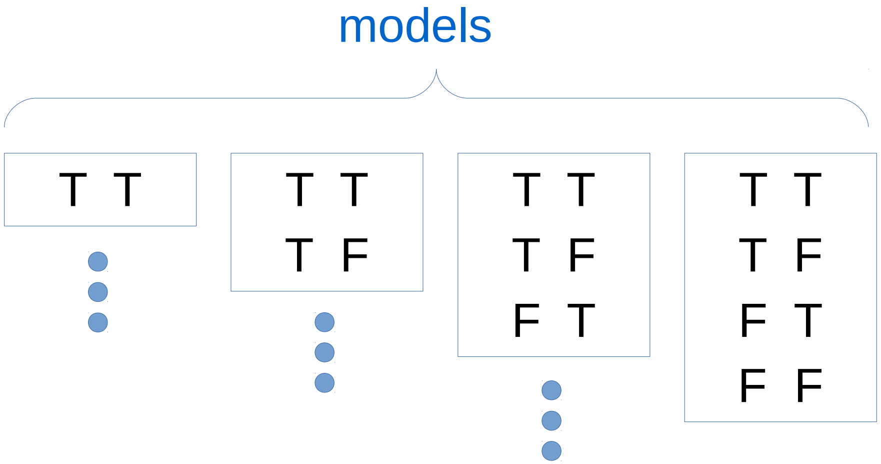


The set of **all** these assignments constitutes
A **model** or a **possible world**

All possible worlds (assignments) are permissible

A	B	A∧B	A∧B ⇒ A
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

Every atomic proposition : A, B



$$2^4 = 16$$

PL: Interpretation

Semantics : the meaning of formulas

Truth values are assigned to the atoms of a formula
in order to evaluate the truth value of the formula

	A	B	
Interpretation I_1 ➡	T	T	
Interpretation I_2 ➡	T	F	
Interpretation I_3 ➡	F	T	
Interpretation I_4 ➡	F	F	

An **interpretation** for A is a **total function** I_A: P_A → {**T**, **F**}
that assigns the truth values **T** or F to every atom in P_A

A ∈ F a formula
P_A the set of atoms in A

https://en.wikipedia.org/wiki/Syntax_(logic)#Syntactic_consequence_within_a_formal_system

PL: Material Implication vs Logical implication

Given two propositions A and B,

If $A \Rightarrow B$ is a tautology

It is said that A **logically implies** B $(A \Rrightarrow B)$

Material Implication $A \Rightarrow B$ (not a tautology)

Logical Implication $A \Rrightarrow B$ (a tautology)

A	B	$A \Rightarrow B$
T	T	T
T	F	F
F	T	T
F	F	T

A	B	$A \land B$	$A \land B \Rightarrow A$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

tautology

$A \land B \Rrightarrow A$

PL: Entailment

if **A**→**B** holds in every model then **A** ⊨ **B**,
and conversely if **A** ⊨ **B** then **A**→**B** is true in every model

any model that makes **A∧B** true

also makes **A** true **A∧B ⊨ A**

No case : True ⇒ False

A	B	A⇒B
T	T	T
T	F	F
F	T	T
F	F	T

A	B	A∧B	A∧B ⇒ A
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

Entailment **A∧B ⊨ A**, or **A∧B ⇛ A**

PL: Validity of Arguments (1)

An **argument form** is valid if and only if

whenever the premises are **all true**, then conclusion is **true**.

An **argument** is valid **if** its **argument form** is **valid**.

If premises : **true**	⇨	**then** conclusion : **true**		
false		**true**		
false		**false**		
If **true**	⇨	**then** **never** **false**		

http://math.stackexchange.com/questions/281208/what-is-the-difference-between-a-sound-argument-and-a-valid-argument

PL: Validity of Arguments (2)

A deductive argument is said to be **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**.

If **true** ➡ **then never** **false**

Otherwise, a deductive argument is said to be **invalid**.
for the premises to be **true** and the conclusion is **false**.

http://www.iep.utm.edu/val-snd/

PL: Soundness of Arguments

An argument is sound if and only if

it is valid and **all** its premises are **true**.

If	premises : **true**	➡	then	conclusion : **true**
	false			**true**
	false			**false**
If	**true**	➡	**then never**	**false**

All premises : **true**

http://math.stackexchange.com/questions/281208/what-is-the-difference-between-a-sound-argument-and-a-valid-argument

PL: Validity and Soundness of Arguments (3)

A	B	A⇒B	A∧(A⇒B)	A∧(A⇒B)⇒B
T	T	T	T	**T**
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

valid

If premises : **true then <u>never</u>** conclusion : **false**

A	B	A⇒B	A∧(A⇒B)	A∧(A⇒B)⇒B
T	T	T	**T**	**T**
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

sound

Always premises : **true therefore** conclusion : **true**

http://www.iep.utm.edu/val-snd/

Interpretation

an **interpretation**

(a) an entity in D is assigned to each of the constant symbols.
Normally, every entity is assigned to a constant symbol.

(b) for each **function**,
an entity is assigned to each possible input of entities to the **function**

(c) the predicate '**True**' is always assigned the value T
The predicate '**False**' is always assigned the value F

(d) for every other **predicate**,
the value T or F is assigned
to each possible input of entities to the **predicate**

Formulas and Sentences

free variables

An **formula**
- A **atomic formula**
- The operator ¬ followed by a **formula**
- Two formulas separated by ∧, ∨, ⇒, ⇔
- A quantifier following by a variable followed by a formula

A **sentence**
- A **formula** with no free variables

∀x love(x,y)	: free variable y	: not a sentence
∀x tall(x)	: no free variable	: a sentence

Satisfiability of a sentence

If a sentence φ evaluates to **True**
under a given interpretation M,
one says that M **satisfies** φ;

this is denoted M ⊨ φ

A sentence is **satisfiable**
if there is some interpretation
under which it is **True**.

Satisfiability of a formula

Satisfiability of formulas with free variables
is more *complicated*,
because an interpretation on its own
does *not* determine the truth value of such a formula.

The most common convention is that
a formula with free variables is said
to be **satisfied** by an interpretation
if the formula *remains* **true**
regardless which individuals
from the domain of discourse
are assigned to its free variables.

This has the same effect as
saying that a formula is **satisfied**
if and only if its universal closure is **satisfied**.

https://en.wikipedia.org/wiki/First-order_logic

Validity of a formula

A formula is **logically valid** (or simply **valid**)

if it is **valid** in every interpretation, or

if it is **satisfied** by every interpretation

These formulas play a role *similar* to
tautologies in propositional logic.

Valid formula examples

A formula is **valid**
if it is **satisfied** by *every* interpretation

free variables

Every **tautology** is a **valid** formula

A **valid** sentence: human(John) ∨ ¬human(John)

A **valid** sentence: ∃x **(** human(x) ∨ ¬human(x) **)**

A **valid** formula: loves(John, **y**) ∨ ¬loves(John, **y**)

True regardless of which individual
in the domain of discourse is assigned to y
This formula is true in every interpretation

Contradiction

A sentence is a **contradiction** if there is **no** interpretation that satisfies it

$\exists x$ **(** human(x) $\land$ $\neg$human(x) **)**

not satisfiable under any interpretation

Logical implication of a formula

A **formula B** is a **logical consequence** of a formula **A**
if every interpretation that makes **A true** also makes **B true**.

In this case one says that **B** is **logically implied** by **A**.

Given two formulas **A** and **B**, if **A ⇒ B** is **valid**:

A logically implies B **A ⇛ B**

free variables

A formula **A ⇒ B** is **valid**
if it is **satisfied** by *every* interpretation

Logical implication examples

Given two formulas **A** and **B**, if **A** ⇒ **B** is **valid**:

A logically implies B **A** ⭆ **B**

human(John) ∧ **(** human(John) ⇒ mortal(John) **)** ⭆ mortal(John)

A **B**

human(x) ∧ **(** human(x) ⇒ mortal(x) **)** ⭆ mortal(x)

valid if it is **satisfied** by *every* interpretation

Logical equivalence examples

Given two formulas A and B,   if **A** ⇔ **B** is **valid**:

A is **logically equivalent** **B**   **A** ≡ **B**

(human(John) ⇒ mortal(John) **)** ≡ **(** ¬ human(John) ∨ mortal(John) **)**

valid if it is **satisfied** by *every* interpretation

Some Logical Equivalences

A and B are **variables** representing *arbitrary predicates*
A and B could have other arguments besides x

$\neg\exists x\ A(x) \equiv \forall x\ \neg A(x)$

$\neg\forall x\ A(x) \equiv \exists x\ \neg A(x)$

$\exists x\ (A(x) \vee B(x)) \equiv \exists x\ A(x) \vee \exists x\ B(x)$

$\forall x\ (A(x) \wedge B(x)) \equiv \forall x\ A(x) \wedge \forall x\ B(x)$

$\forall x\ A(x) \equiv \forall y\ A(y)$

$\exists x\ A(x) \equiv \exists y\ A(y)$

Logical Validity and Tautology

Tautology

- defined in the context of *proposition*
- can be extended to sentences in the *first order logic*

In *propositional* logic the following two coincide
In *first order logic*, they are distinguished

Logical Validities
Sentences that are **true** in every model (in every interpretation)

Tautologies
A proper *subset* of the first-order logical validities

Logical Validity & Tautology

¬, ∧, ∨

A unary relation symbols R, S, T

$(((\exists x Rx) \land \neg(\exists x Rx)) \to (\forall x Tx)) \iff ((\exists x Rx) \to ((\neg\exists x Sx) \to (\forall x Tx)))$
: **logical validity** in first order logic

$(\exists x Rx)$: A
$(\neg\exists x Sx)$: B
$(\forall x Tx)$: C

$((A \land B) \to C) \iff (A \to (B \to C))$
: a **tautology** in propositional logic

Logical Validity & Tautology

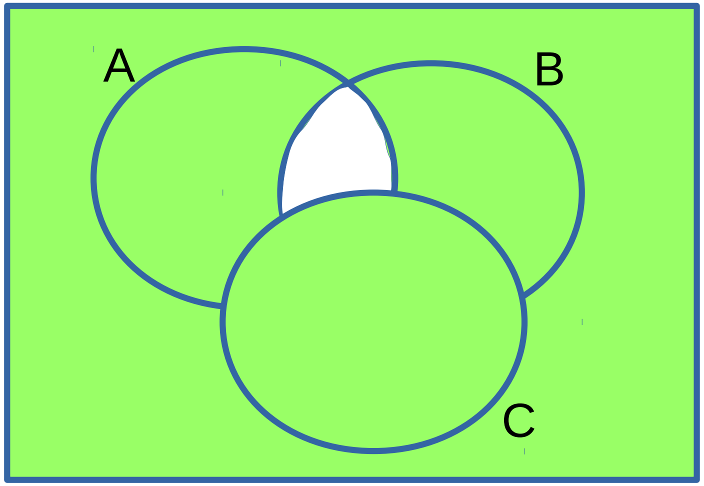
$((A \land B) \to C) \Leftrightarrow (A \to (B \to C))$

¬, ∧, ∨

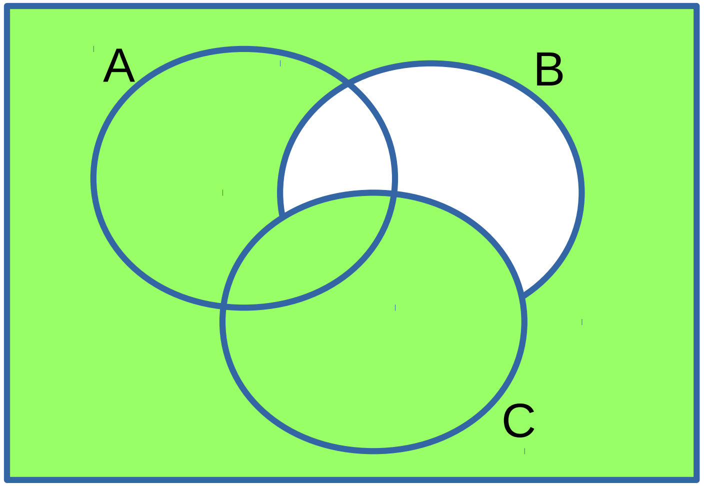
A	B	C	A∧B	C	(A∧B)→C	A	B→ C	A→(B→C)
T	T	T	T	T	T	T	T	T
T	T	F	T	F	F	T	F	F
T	F	T	F	T	T	T	T	T
T	F	F	F	F	T	T	T	T
F	T	T	F	T	T	F	T	T
F	T	F	F	F	T	F	F	T
F	F	T	F	T	T	F	T	T
F	F	F	F	F	T	F	T	T

Logical Validity & Tautology

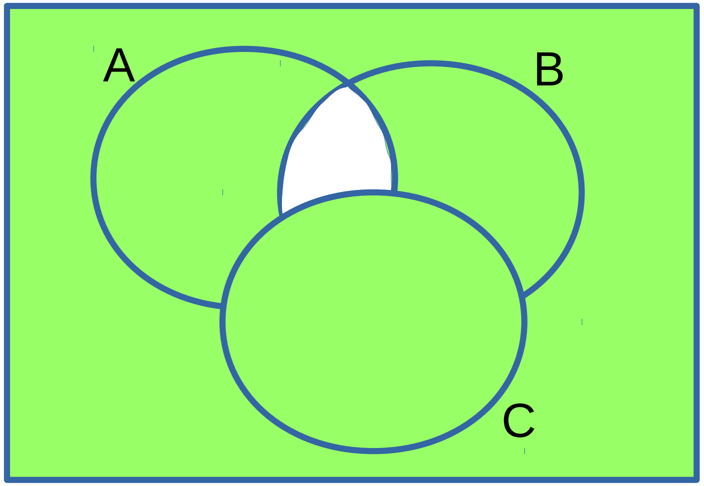
$(A \land B) \rightarrow C$



$B \rightarrow C$



$A \rightarrow (B \rightarrow C)$



Not all logical validities are tautologies

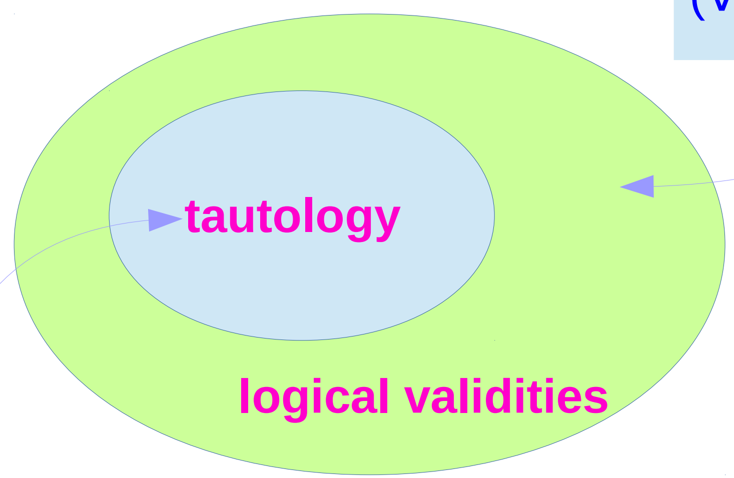
(∀*x* Rx) → ¬ ∃*x* ¬Rx **logical validities** in first order logic
A → B the corresponding propositional sentence is **not** a **tautology**

¬, ∧, ∨

(∀*x* Rx) → ¬ ∃*x* ¬Rx

A → B

~~**tautology**~~



(((∃*x*Rx)∧¬(∃*x*Rx)) → (∀*x*Tx)) ⟺
((∃*x*Rx) → ((¬∃*x*Sx) → (∀*x*Tx)))

((A∧B) → C) ⟺
(A → (B → C)) **tautology**

Tautology in first order logic

A tautology in first order logic

A sentence that can be obtained
by taking a **tautology** of propositional logic
and uniformly replacing each propositional variable
by a first order formula
(one formula per propositional variable)

$A \lor \neg A$: a tautology of propositional logic

$\forall x\ (x = x) \lor \neg \forall x\ (x = x)$ is a tautology in first order logic

References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, “Lecture Notes : Introduction to Prolog Programming”
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html
[8] http://ilppp.cs.lth.se/, P. Nugues,` An Intro to Lang Processing with Perl and Prolog