Phil 120, Review Notes

Stuff

May 23, 2021

Contents

1 References

1.1 Basic Logic Operators

The followings are for A * B, where '*' is an operator, A is top row, B is left column.

$\wedge$	T	F
T	T	F
F	F	F

AND. Conjuction. A $\wedge$ B is true only when both A and B are true.

$\vee$	T	F
T	T	T
F	T	F

OR. Disjunction. A $\vee$ B is true when either A or B, or Both are true.

$\rightarrow$	T	F
T	T	T
F	F	T

IMPLIES. If A then B. A implies B.
A implies B is true when A is true and B is true, or when A is false.
Note: $A \rightarrow B = \neg A \vee B$

$\leftrightarrow$	T	F
T	T	F
F	F	T

IFF, A if and only B. A is logically equivalent to B, two way implication.
A $\leftrightarrow$ B is true exactly when the truth value of A is the same as B.
Note: $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A) = (A \wedge B) \vee (\neg A \wedge \neg B)$

1.2 Some Logic Identities

$A \vee \neg A = T$	Excluded Middle, either A or not A must be true. (Not true in K3, since it could be the case that both A and $\neg A$ are not true)
$\neg(A \wedge \neg A)$	Non-contradiction. It is true that not both A and not A hold at the same time. (Not true in K3)
$A \to B, A \implies B$	Modus ponens, to prove. If A implies and B and A is true, then B is true.
$A \to B, \neg B \implies \neg A$	Modus tollens, to disprove. If the conclusion is false, then the premise must be false also.
$A \to A \vee B$	Disjunctive Introduction If A is true, then A or B is also true, regardless if what B is.
$A \vee B, \neg A \implies B$	Disjunctive syllogism. If at least one of A or B is true, then if one of them is false, the other must be true.
$(A \to B) \iff (\neg B \to \neg A)$	Contrapositive. Similar to Modus tollens.
$A, \neg A \implies B$	Explosion. From a false premise you can arrive at any conclusion.
$\neg(A \vee B) \iff \neg A \wedge \neg B$ $\neg(A \wedge B) \iff \neg A \vee \neg B$	De Morgan's Law: $\neg$ distributes over $\vee, \wedge$
$A \vee (B \wedge C) \iff (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$	Distributability: $\vee$ distrubutes over $\wedge$, and vice versa.

Note: In LP, all of modus ponens, modus tollens, contrapositive, and explosion are all not valid, because disjunctive syllogism does not hold in LP.

1.3 Tableaux Identities

$\mathfrak{t}\land$:	$\begin{array}{c} \mathfrak{t}A \land B \\ \vert \\ \mathfrak{t}A \\ \mathfrak{t}B \end{array}$	$\mathfrak{f}\land$:	$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{f}A \land B} \\ \diagup & \diagdown \\ \mathfrak{f}A & \mathfrak{f}B \end{array}$
$\mathfrak{t}\lor$:	$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{t}A \lor B} \\ \diagup & \diagdown \\ \mathfrak{t}A & \mathfrak{t}B \end{array}$	$\mathfrak{f}\lor$:	$\begin{array}{c} \mathfrak{f}A \lor B \\ \vert \\ \mathfrak{f}A \\ \mathfrak{f}B \end{array}$
$\mathfrak{t}\rightarrow$:	$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{t}A \rightarrow B} \\ \diagup & \diagdown \\ \mathfrak{t}\neg A & \mathfrak{t}B \end{array}$	$\mathfrak{f}\rightarrow$:	$\begin{array}{c} \mathfrak{f}A \rightarrow B \\ \vert \\ \mathfrak{f}\neg A \\ \mathfrak{f}B \end{array}$
$\mathfrak{t}\leftrightarrow$:	$\begin{array}{cccc} \multicolumn{4}{c}{\mathfrak{t}A \leftrightarrow B} \\ \mathfrak{t}\neg A & \mathfrak{t}\neg A & \mathfrak{t}\neg B & \mathfrak{t}A \\ \vert & \vert & \vert & \vert \\ \mathfrak{t}\neg B & \mathfrak{t}A & \mathfrak{t}B & \mathfrak{t}B \end{array}$	$\mathfrak{f}\leftrightarrow$:	$\begin{array}{cc} \multicolumn{2}{c}{\mathfrak{f}A \leftrightarrow B} \\ \diagup & \diagdown \\ \mathfrak{f}\neg A & \mathfrak{f}\neg B \\ \vert & \vert \\ \mathfrak{f}B & \mathfrak{f}A \end{array}$
$\mathfrak{t}\forall$:	$\begin{array}{c} \mathfrak{t}\forall x\ Ax \\ \vert \\ \mathfrak{t}A\alpha \end{array}$	$\mathfrak{f}\forall$:	$\begin{array}{c} \mathfrak{f}\forall x\ Ax \\ \vert \\ \mathfrak{f}A\eta \end{array}$
$\mathfrak{t}\exists$:	$\begin{array}{c} \mathfrak{t}\exists x\ Ax \\ \vert \\ \mathfrak{t}A\eta \end{array}$	$\mathfrak{f}\exists$:	$\begin{array}{c} \mathfrak{f}\exists x\ Ax \\ \vert \\ \mathfrak{f}A\alpha \end{array}$
$\mathfrak{t} =$:	$\begin{array}{c} \mathfrak{t}a = b \\ \mathfrak{t}A(a) \\ \vert \\ \mathfrak{t}A(b) \end{array}$	$\mathfrak{f} =$:	$\begin{array}{c} \mathfrak{t}a = b \\ \mathfrak{f}A(a) \\ \vert \\ \mathfrak{f}A(b) \end{array}$

Quantifier & Identity Note:

- α means an arbitrary choice of variable name could used in place of α, might be a good idea to just use a.
- η means a new variable name that has previously unused in the current branch must be used to replace η.
- Use the following to deal with $\neg$:

$$\neg\exists x\ Ax \iff \forall x\ \neg Ax$$

$$\neg\forall x\ Ax \iff \exists x\ \neg Ax$$

$$(\mathfrak{f}\neg(a = b)) \iff (\mathfrak{t}a = b)$$

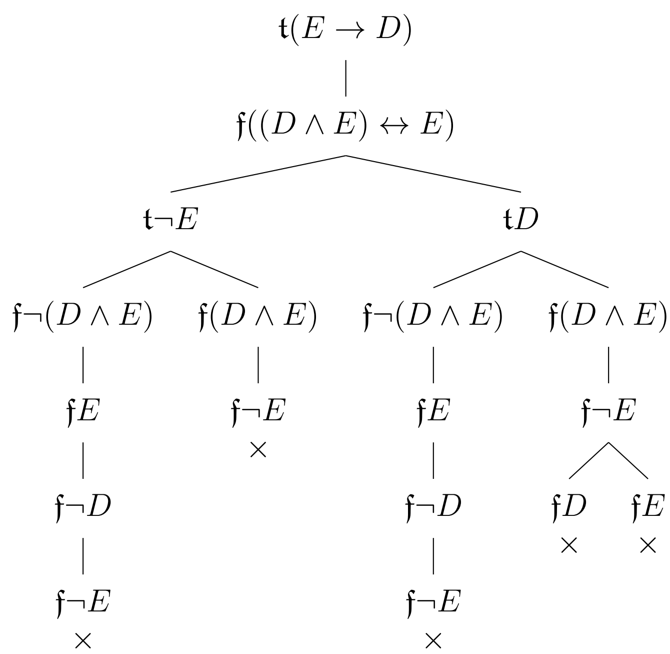
- Identiy closure condidtions: $\mathfrak{f}a = a,\ \mathfrak{t}\neg(a = a)$.

- Remember that $=$ is symmetric, so that $a = b \iff b = a$.
- Branches involving quantifiers are undecidable, so some strategies and luck is involved in closing those branches.
- Quantifier order matters!:
  - Swapping from $\exists x \forall y$ to $\forall y \exists x$ is okay, the reserve is not, in general.

Note:

- Use De Morgan's Law to deal with negations($\neg$), also, $\neg(A \to B) = (\neg B \wedge A)$.
- To prove a consequence, set the premise to true, conclusion to false. This proves that there is no possible counter example, since the negation is never satisfied.
  - If there are atomic branches left open, then those are valid counter examples.
- A branch is closed any of following pairs occurs in a branch: $\{(\mathfrak{f}A, \mathfrak{t}A), (\mathfrak{f}A, \mathfrak{f}\neg A), (\mathfrak{t}A, \mathfrak{t}\neg A)\}$, use a '$\times$' to indicate a closed branch.

Example: Use tableaux to prove $(E \to D) \vdash_1 ((D \wedge E) \leftrightarrow E)$



Since all branches are closed, the negation of the conclusion is never satisfied, thus the relation always holds for all values D and E might take on.

2 Notable Definitions from Part 1

2.1 Consequences

Logical Consequence: $A_1 \ldots A_n$ implies B, and B is a consequence of $A_1 \ldots A_n$, means when $A_1 \ldots A_n$ are all true, then B must be true also.

Case: A case be loosely interpreted a particular combination of values for variables.

Valid Argument: An argument consist of a set of premises and a single conclusion, this argument is *valid* if the conclusion is a *logical consequence* of the premises.

Counter Example: A counter example to an argument is a case where all premises are truth, but the conclusion is false.

Sound Argument: An argument is sound if the premises are true in *all* cases, and the arugment is valid. An argument cannot be sound if its not already valid.

2.2 Language

Syntax: Syntax consist of a basic set of symbols, and a rule set to create more complex words & sentences from symbols. Syntax is not concerned with *meaning* of any symbols or sentences

Semantics: Semantics of a language assigns meaning to a sentence in the language.

Atom, Connectives, Molecules: An atomic sentence is the mostly basic sentence that cannot be reduced further, like 'sky is blue' or 'Bob is eating', atomic sentence do not have connectives. A molecular sentences is made with a number of atomic sentences linked with connectives, like 'Bob is sleeping *or* eating', 'Sun is bright *and* hot'.

2.3 Basics of Set Theory

Set: A set is an arbitrary, unordered *collection* of unique *things*, depending on context, duplicates are usually ignored. 2 sets are equal if they contain indentical items. For example:

$$Food := \{apple, cookie, burger\} = \{apple, apple, apple, burger, cookie\}$$

Membership($\in, \notin$): For any set it is possible to tell if an item belongs in the set. For exmaple:

$$cookie \in Food, dirt \notin Food$$

Which means that 'cookie' is in the set of Food(cookie is a member of Food), but dirt is not.

Set builder notation: A notation used to contruct sets from definitions. For exmaple:

$$L = \{n \in \mathbb{N} : n > 44\}$$

Here the ':' means 'such that', so the set L is the set all natural numbers, n, such that n is larger than 44.

Union($\cup$): The union of 2 sets is a set containing items from either sets:

$$\{1, 3, 7\} \cup \{2, 3, 2\} = \{1, 2, 7, 3\}$$

Intersect($\cap$): The intersection of 2 sets is a set containing items that belongs to both sets:

$$\{1, 3, 7\} \cup \{1, 2, 3, 4\} = \{1, 3\}$$

Subsets($\subseteq$): $A \subseteq B$ if A is contained in B, that is, every item in A is also in B.
Note: $A = B \iff (A \subseteq B) \wedge (B \subseteq A)$.

Proper Subset($\subset$): A is a proper subset of B if A $\subseteq$ B, and B is strictly bigger, that is, contains at least one item A does not.

2.4 Pairs and Relations

Ordered Pair: Unlike sets, ordered pairs/n-tuple are ordered. So $\{a, b\} = \{b, a\}$, however, $\langle a, b\rangle \neq \langle b, a\rangle$. N-tuples contains n ordered items.

Cartesian Product: $A \times B$ is the cartesian product of A and B, which is a set containing all possible ordered pairs $\langle a, b\rangle, a \in A, b \in B$. $\times$ can be applied more than 2 times. For exmaple:

$$\{a, b, c\} \times \{1, 2\} = \{\langle a, 1\rangle, \langle a, 2\rangle, \langle b, 1\rangle, \langle b, 2\rangle, \langle c, 1\rangle, \langle c, 2\rangle\}$$

$$D \times E \times F = \{\langle d_1, e, f_1\rangle, \langle d_1, e, f_2\rangle, \langle d_1, e, f_3\rangle, \langle d_2, e, f_1\rangle, \langle d_2, e, f_2\rangle, \langle d_2, e, f_3\rangle\}$$

$$\text{Where } D = \{d_1, d_2\}, E = \{e\}, F = \{f_1, f_2, f_3\}$$

Relations: A relation $\mathcal{R}$ on sets A and B, is a way to relate elements of A and B. For $a \in A, b \in B$, a and b are in relation $\mathcal{R} \iff \langle a, b\rangle \in \mathcal{R}$, and we can write $a\mathcal{R}b$.
Note: $\mathcal{R} \subseteq A \times B$.

Reflexivity: A relation $\mathcal{R}$ is reflexive when $x\mathcal{R}x$ for all x.

Symmetry: $\mathcal{R}$ is symmetric when $x\mathcal{R}y \iff y\mathcal{R}x$

Transitivity: $\mathcal{R}$ is transitive when $x\mathcal{R}y, y\mathcal{R}z \implies x\mathcal{R}z$.

Equivalence: $\mathcal{R}$ is an equivalence relation if $\mathcal{R}$ is reflexive, symmetric, and transitive.

Function: Like functions in calculus, $f : x \to y$ sends each x to 1 y only, that is, the value of $f(x)$ is not ambiguous.

3 Classical Logics

For logic operators and tableaux references, see Section 1.

3.1 Turntile($\vdash$) vs Double turntile($\models$)

Turntile: '$\vdash$' denotes *syntatic* implication. $A \vdash_1 B$ means with only information from A, it is possible to prove B. Or alternatively, it is possible to obtain B from 'rearranging' symbols of A.

Double turntile: '$\models$', denotes *semantic* implication, or models. $A \models_1 B$ means that B is true whenever A is true.

Notes: A logic system is *sound* if $A \vdash B \implies B$, is *complete* if $A \models B \implies A \vdash B$. Classical logics is sound and complete so there isn't a big difference between the two symbols used.

3.2 Cases

True/False in case: For a particular case c, $c \models_1 A$ means 'A is true in case c', while $c \models_0 A$ means 'A is false in case c'.

Complete/Consistant Cases:

- A case is *complete* if at least 1 of $c \models_1 A, c \models_0 A$ holds.
- A case is *consistant* if at most 1 of $c \models_1 A, c \models_0 A$ holds(so not both).
- All cases in classical logic is both complete and consistant.

Skipping things covered in Section 1 or informally covered in previous sections

3.3 Analytic Tableaux

For example and references on tableaux method of proof, see section 1.3.

Tableaux Method: The tableaux method provides a systematic way to determine if a statement is:

- true / a theorem, if all atomic leaves of the tree are not closed.
- false / a contradiction, if all branches are closed.
- contingent, in which case the remaining unclosed branches determines the truth value of the statement.

In particular, to prove a statement, it is helpful to start with the negation/counter example instead, then try to show the negation is never satisfied. So when proving $A \vdash_1 B$, start with:

$$\mathfrak{t}A$$

$$\mathfrak{f}B$$

Then follow the reference and example in section 1.3, if all branches closes, then that means the condition of A been true, while B is false, is never satisfied, thus no counter example exists, therefore $A \vdash_1 B$.

When there some branches left open, then that branche's nodes together forms a counter example when their truth value is same as their sign($\mathfrak{t}$ or $\mathfrak{f}$).

Sign of Node: For notation in this class, each node of a tableaux is *signed*($\mathfrak{t}$ or $\mathfrak{f}$), which are essentially assuming the truth value of that node.
Note, $\mathfrak{t}$ and $\mathfrak{f}$ signed nodes splits differently, and children nodes have same sgin as their parent. Also:

$$\mathfrak{t}A = \mathfrak{f}\neg A, \mathfrak{f}A = \mathfrak{t}\neg A$$

Skipping chapter 7, doesn't look important.

4 First Order Logic

4.1 Predicates

Unary Predicate: A single variabled *well formed formula*(or a boolean function) that evaluates to either true or false depending on the variable. In this class, a predicate looks like Pa where a is the variable, P is the predicate.

'Denotes': Suppose *Object* has name of *'Name'*, then *'Name'* denotes *Object*, and *Object* is the denotation of *'Name'*

Extension & AntiExtension: The *Extension* is the set of objects(denotations) that makes the predicate true. The anti-extension makes the predicate false. For example, suppose Px is true if x is red:

$$+(P) = \{apple, cherry, stop\ sign, \cdots\},\ -(P) = \{sky, orange, Phil\ 120\ midterm, \cdots\}$$

Domain of Discourse: Domain of Discourse, D, for a predicate is a set of objects applicable to the predicate. So if the predicate is 'x is the first day of the week', the D might be the set of all days in a week. Note: $+(P) \cup -(P) = D$, and $+(P) \cap -(P) = \emptyset$, in order for the predicate to behave classically.

Denotation Function(δ): The denotation function, $\delta(a)$, gets the denotation from the domain of Discourse which a denotes. For a predicate P, Pa is true iff $\delta(a) \in +(P)$

Interpretation: A interpretation(*case*) contains a domain of discourse and a denotation function($\langle D, \delta \rangle$), which is a more formal definition of a *case*, which previous parts have been mentioning.

Interpretations can be used to satisfy statements, or to serve as counter examples, like in A2 Q2:

a) Find an interpretation which satisfy $\Gamma = \{(\neg Ha \vee Hb), (\neg Ha \to \neg Hb), (\neg Hc \vee Hd)\}$

$$\begin{aligned}
&\text{Define } \mathcal{I} = \langle D, \delta \rangle \\
&\text{Where } D = \{o_1, o_2, o_3, o_4\} \\
&\text{and } \delta(a) = o_1, \delta(b) = o_2, \delta(c) = o_3, \delta(d) = o_4 \\
&\text{such that } +(H) = \{o_1, o_2, o_3, o_4\}, -(H) = \{\}
\end{aligned}$$

Now, $Ha, Hb, Hc, Hd = T$
$\implies \mathcal{I} \models_1 (\neg Ha \vee Hb),\ \mathcal{I} \models_1 (\neg Ha \to \neg Hb),\ \mathcal{I} \models_1 (\neg Hc \vee Hd)$
$\implies \mathcal{I} \models_1 \Gamma$

b) Find a counter exmaple to $\{\neg Ga, (Ga \to Pb), (\neg Pb \to Ga)\} \models_1 (Ga \vee \neg Pb)$

$$\begin{aligned}
&\text{Define } \mathcal{I} = \langle D, \delta \rangle \\
&\text{Where } D = \{o_1, o_2\} \\
&\text{and } \delta(a) = o_1, \delta(b) = o_2 \\
&\text{such that } +(G) = \{\}, -(G) = \{o_1\}, +(P) = \{o_2\}, -(P) = \{\}
\end{aligned}$$

So that in this interpreation Ga is false, while Gb is true
$\implies \neg Ga = T,\ (Ga \to Pb) = T,\ (\neg Pb \to Ga) = T.$
However $(Ga \vee \neg Pb) = F$, so in this interpretation, the premise is true but the conclusion is false, which makes this a counter example.

4.2 Quantifiers

Motivation: To decribe statements like "Every apple is sweet", "Some cities rains all the time.", which contains quantifiers.

Universal and Existential($\forall, \exists$):

- '$\forall$' translates into 'for all', 'every', 'any', is used to specify any and all of the items in a set:

  $$\forall n \in \mathbb{N}, 2n > n. \text{ Meaning for all natural number n, 2n is larger than n.}$$

- '$\exists$' translates in to 'exists', 'at least one', 'some', is used to specify at least one of an item in a set:

  $$\exists z \in \mathbb{Z} : z^2 = z. \text{ There exist an integer } z \text{, such that } z^2 = z.$$

- Quantifiers are tied to variables within its scope. In $P(z) := \exists x(Px \lor Pz) \land (\forall y \neg Py \to Px)$, $\exists x$ applies to the whole formula, but $\forall y$ only applies to the inner bracket. Here z is a *free* variable, since it is not bound by a quantifier.
- Order matters for quantifiers, for example:

  $$\exists x \forall y \in \mathbb{R}, x < y \text{ is false, since there is no smallest element in } \mathbb{R}$$

  $$\forall y \exists x \in \mathbb{R}, x < y \text{ however holds, since for all } y \text{, there is a } x \text{ that is smaller.}$$

- To prove or find counter example:
  - To prove $\forall x, Px$, it is necessary to show Px holds for any arbitrary x.
  - To disprove/find counter example for $\forall x, Px$, it is sufficient to find a single x such that Px does not hold.
  - To prove $\exists x, Px$, it is sufficient to show there is at least 1 x such that Px holds.
  - To disprove $\exists x, Px$ it is necessary to show Px is not satisfied for all x.

Name vs Variable: Name is refering to a static named reference of a object, while a variable is an unknown. For example : "Bob is short, and for any person, p, p is friends with Bob. x is even shorter than Bob". Here Bob is *named*, while p is a variable bound by quantifier, and x is a free variable since it is not bound by a quantifier.

Valuation Function: For a particular interpretation, a valuation function may be added $\to \langle D, \delta, v \rangle$. $\delta(a)$ is used to get the object in D that is used in place of names, while $v(x)$ is used to get object to be used in place of *variable.* v is similar to δ, but for variable.

Prof's Definitions for Quantifier: First, for any particular valuation function v, define v' such that $v' {\sim_x} v$, v' returns the same value as v, except for $v'(x)$. Now, universal and Existential quantifier may be define like so:

- $w \models_1^v \forall x Gx \iff$ for any $v' {\sim_x} v, w \models_1^{v'} Gx$.
  So no matter which object in D is put in place of x, the statement holds still.
- $w \models_1^v \exists y Gy \iff$ there is at least 1 $v' {\sim_y} v, w \models_1^{v'} Gy$.
  So at least 1 object exist in D, such that when put in place of y, the statement holds.

Predicate with Multiple Free Variable: A predicate may have more than 1 variable, for example Gab, for the predicate of '$a < b$. Predicate with multiple variable could be written like 'P^n', where n is the number of variable. For a predicate of n variable, then the extension/anti-extension should contain *n-tuples* since the predicate needs n variables.

4.3 Identity

Motivation: To determine if 2 objects are 'Equal', that they behave identically, or have the same properties.

Extention and AntiExtention:

$$+(=) \text{ is } \{\langle o, o\rangle : o \in D\} \text{ (the identity relation, or } Id_D)$$

$$-(=) \text{ is } \{\langle o_1, o_2\rangle : o_1, o_2 \in D, o_1 \neq o_2\}$$

Such that everything is identical to itself, and nothing else. Recall that a equivalence relation is reflexive($x = x$), symmetric($x = y \iff y = x$), and transitive($x = y \wedge y = z \implies x = z$).

First Order Leibniz Law: if $a = b$, then a is interchangable with b:

$$Pa, a = b \models_1 Pb$$

Properties of $\neq$:

- $(a \neq b \iff \neg(a = b))$
- $\forall x, x \neq x$ is always an contradiction, $\neq$ is not reflexive
- $\forall x, y \quad x \neq y \iff y \neq x$, $\neq$ is symmetric
- In general $\exists x, y, z : x \neq y \wedge y \neq z \not\Longrightarrow x \neq z$, $\neq$ is not transitive
- $Pa, a \neq b \not\models_1 Pb$

For tableaux formulas involving quantifier and equality, see the reference section.

5 Other Logcial systems

This section examines logical systems with 3(Neither) or 4(Neither and Both) logical values. For example, "it is raining right now." is either true or false, but "it will rain tormorrow" is indeterminant, since it's not possible to know for sure if it will rain tormorrow.

5.1 CL, K3, LP, FDE

Options for alternative logic systems:

Consistant	Complete	Logical Theory
Yes	Yes	CL Classical Logics
Yes	No	K3 Strong Kleene
No	Yes	LP Logics of Paradox
No	No	FDE First-Degree Entailment

Paracomplete & Paraconsistent:

- A logical theroy is *paracomplete* if it is not complete, that is for some statement A, $\exists c \ : \ c \not\models_1 A \wedge c \not\models_0 A$, so some statement in some cases cases can be both not true, and not false at the same time.
- A logical theory is *paraconsistent* if it is not consistent, that is, for some statement A, $\exists c \ : \ c \models_1 A \wedge c \models_0 A$, so that under some cases, some statements can be both true and false at once.
- Note: K3 is *Paracomplete*, LP is *Paraconsistent*, FDE is both(so neither and both true and false cases exists).

Logical Consequences: Let $\vdash_{Cl} \quad \vdash_{K3} \quad \vdash_{LP} \quad \vdash_{FDE}$ denote consequence relation in each logical system. Where $w \vdash_T A$ means that under the logical theory T, there no case where w is satisfied, but A isn't; or in other words, no counter examples exists.

- $w \vdash_{K3} A \implies w \vdash_{CL} A$. This holds because logical consequence really means that no counter example exist, and also that classical cases are subset of Incomplete cases. So no incomplete counter examples exists $\implies$ no classical counter examples exists.
  (This shows that a language can be incomplete(Like most natural languages) yet still have cases that is entirely precies and classical.)
- $w \vdash_{LP} A \implies w \vdash_{CL} A$. This holds similar to reason above. (A language can be inconsistent/ have paradox yet still have a subset of cases that behaves classically.)
- Similarly, $w \vdash_{FDE} A \implies w \vdash_{CL} A, \ w \vdash_{LP} A, \ w \vdash_{K3} A$

Semantic Values: Each theory have a set of *semantic values*, which are results of interpreting a statement, in classical theory, a statement is either true or false, so $\mathcal{V}_{CL} = \{T, F\}$. The semantic values for other thories are as follows:

Theory	Semantic Value
CL	$\{T, F\}$
K3	$\{T, F, N\}$
LP	$\{T, F, B\}$
FDE	$\{T, F, B, N\}$

Where N is semantically nethier($c \not\models_1 A$ and $c \not\models_0 A$), B is both true and false($c \models_1 A$ and $c \models_0 A$).
Logical conjunctions holds the same way as in classical theory, for exmaple:
$c \models_1 A \vee B$ iff $c \models_1 A$ or $c \models_1 B$

5.2 Logic Identities in K3 and LP

When moving from classical logics to a paraconsistent or paracomplete theory, some logical identities are sacrificed.

K3: In K3, Excluded Middle and Non-contradiction are both lost compared to classical logics. See section 1.2.

LP: In LP, its notable that all classical true statements are also truth in LP. Also, *modus ponens, modus tollens*, Disjunctive Syllogism, and Explosion are all lost when moving from classical to LP. For a counter example, if A is both true and false, and B is false only, then $A \vee B$ is not false, then $\neg A$ is not false, but B is false.