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: $A \rightarrow B = \neg A \vee B$

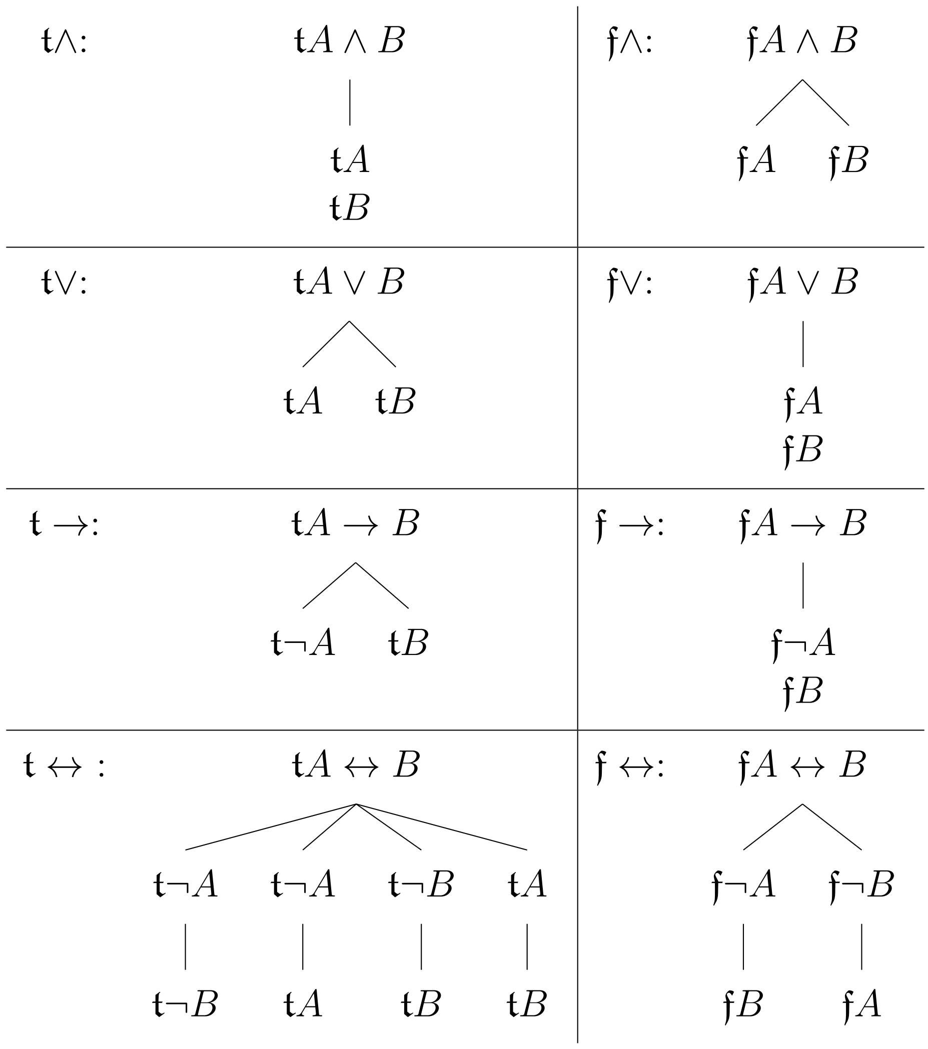
$\leftrightarrow$	T	F
T	T	F
F	F	T

IFF, A if and only B. A is logically equivalent, 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.
$\neg(A \wedge \neg A)$	Non-contradiction. It is true that not both A and not A hold at the same time.
$A \rightarrow B, A \implies B$	Modus ponenes, to prove. If A implies and B and A is true, then B is true.
$A \rightarrow B, \neg B \implies \neg A$	Modus tollens, to disprove. If the conclusion is false, then the premise is false also.
$A \vee B, \neg A \implies B$	Disjunctive syllogism. If at least one of A or B is true, then if one of them is false, the other must be true.
$(A \rightarrow B) \iff (\neg B \rightarrow \neg A)$	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.
$A \vee (B \wedge C) \iff (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$	Distributability

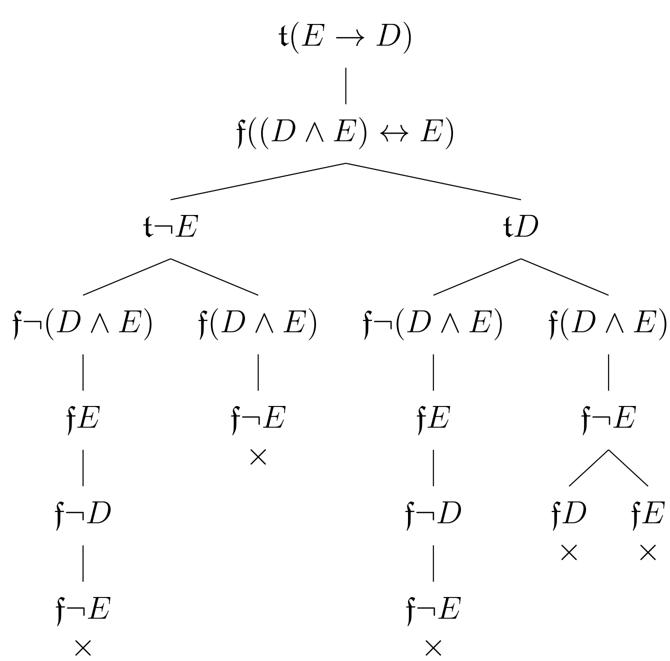
1.3 Tableaux Identities



Note:

- Use De Morgan's Law to deal with negations($\neg$), also, $\neg(A \rightarrow B) = (\neg B \wedge A)$.
- 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 $(E \rightarrow D) \vdash_1 ((D \wedge E) \leftrightarrow E)$



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:

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

Intersec($\cap$): The intersection of 2 sets is a contain 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