Study Guide for Phil 279 L01 F16 (Ver. 3.2)

Teppei Hayashi
Department of Philosophy
University of Calgary

Email: thayashi@ucalgary.ca

October 21, 2016

If you find any mistakes in this document, please let me know IMMEDIATELY.

Contents

1 Language L_1

We are dealing with a language called L_1. So let's start with very basic aspects of this language.

L_1 can be investigated from two different angles: *syntax* and *semantics*. In short, the syntax and semantics of L_1 tell us:

Syntax: What kind of expression is considered as a sentence of L_1 (vocabulary and formation rules of a sentence of L_1).

Semantics: How to assign the truth values {T, F} to a sentence of L_1 (truth table).

Let's take a bit closer look at these two aspects.

1.1 Syntax of L_1

As we've seen in the above, the syntactical aspect of L_1 tells us what kind of expression is considered as a sentence of L_1. As in ordinary languages, the syntax of L_1 comprises of the vocabulary and formation rules (namely, grammar) of L_1-sentences. Let's start with the vocabulary of L_1.

1.1.1 The Vocabulary of L_1

The vocabulary of L_1 comprises of

Logical Symbols: $-, \wedge, \vee, \rightarrow, \leftrightarrow$

Non-Logical Symbols: Sentence letters $A, B, C, \ldots$ with or without subscript (e.g., A_1, B_2, Z_{1028}).

Auxiliary Symbols: Brackets (and). Sentence variables $\alpha, \beta, \gamma, \ldots$ (the lower Greek letters) with or without subscript.

The logical symbols $-, \wedge, \vee, \rightarrow, \leftrightarrow$ are called *logical connectives*, and called "denial" (or "negation"), "conjunction", "disjunction", "conditional", and "bi-conditional" respectively. A sentence variable is used for referring to an arbitrary sentence of L_1. So, the sentence variable α may refer to a sentence letter like A, or a rather complicated sentence like $-(A \wedge (B \leftrightarrow (C \vee D)))$.

1.1.2 The formation Rules of L_1-Sentences

The formation rules of L_1-sentences are as follows.

1. Every sentence letter is a sentence of L_1.
2. If α and β are sentences of L_1, so are $-\alpha, (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$.
3. Nothing else is a sentence of L_1.

Thus, $-(A \wedge (B \leftrightarrow C))$ is a sentence of L_1 but $((\rightarrow A$ is not.

The sentences in the item 2 roughly mean "not α" (or "it's not the case that α"), "α and β", "α or β", "if α, then β" (or "α only if β"), and "α if and only if β" respectively.

Some authors call sentence letters and their denials *literals*.

1.1.3 The Main Connective

In writing truth tables or truth trees, you need to be able to answer the following question: "What kind of a sentence is this?" In the case of a denial sentence, it's easy to answer because every denial sentence has a form "$-(\ldots)$". In the case of a simple sentences like $(A \wedge B)$ or $(A \rightarrow B)$, it's easy as well; the former is a conjunctive sentence, the latter conditional. However, in the case of a more complicated sentence, figuring out what sentence we're thinking of can be a substantial task. Let's take $(((B \vee C) \rightarrow A) \leftrightarrow ((B \rightarrow A) \wedge (-A \rightarrow -C)))$ as an example.

First, note that a sentence of L_1 is built in a bottom-up way; in other words, a bigger sentence should be built from smaller ones. For example, in order to build $((A \wedge B) \rightarrow -C)$, we need to build $(A \wedge B)$ and $-C$ first, and then, connect them by $\rightarrow$. In the case of our example, we need to build $-A$ and $-C$ first, and then $(B \vee C)$, $(B \rightarrow A)$ and $(-A \rightarrow -C)$, and then $((B \vee C) \rightarrow A)$ and $((B \rightarrow A) \wedge (-A \rightarrow -C))$, and lastly connect them by $\leftrightarrow$. This building process can be shown graphically as follows.

$$\underbrace{\underbrace{(\underbrace{(\underbrace{(B \vee C)}_{(2)} \rightarrow A)}_{(3)} \leftrightarrow \underbrace{(\underbrace{(B \rightarrow A)}_{(2)} \wedge \underbrace{(\underbrace{-A}_{(1)} \rightarrow \underbrace{-C}_{(1)})}_{(2)})}_{(3)}}_{(4)}$$

In the stage 4 of the above building process, we connect the two sentences built in the stage 3 by $\leftrightarrow$; in other words, the last connective we used in building the sentence is $\leftrightarrow$. We call this last connective the *main connective*, and say, "the sentence is a bi-conditional".

1.1.4 Problems

Determine the main connectives of the following sentences.

1. $((- - A \wedge B) \rightarrow (-A \leftrightarrow B))$
2. $(-D \wedge (-H \vee (D \wedge E)))$
3. $(-(D \leftrightarrow (-A \wedge B)) \vee (-D \vee -B))$
4. $((J \wedge ((E \vee F) \wedge (-E \wedge -F))) \rightarrow -J)$
5. $-(-A \leftrightarrow -(B \leftrightarrow -(A \leftrightarrow (B \wedge C))))$

1.2 Semantics of L_1

The semantics of L_1 tells us how to assign the truth values {T,F} to a sentence of L_1. Such assignments of the truth values to a sentence are called *interpretations*. Let's start with the simplest one.

1.2.1 Interpretations of a Sentence Letter of L_1

Every sentence letter of L_1 has two truth values {T,F} (but of course not at the same time). The truth table for its interpretations is

A
T
F

1.2.2 Interpretations of a Denial

The interpretations of a denial of some arbitrary sentence α are

α	$-\alpha$
T	F
F	T

1.2.3 Interpretations of a Conjunction

The truth value of a conjunction is going to be true if and only if both of its components (called *conjuncts*) are true; otherwise, it's going to be false.

α	β	$(\alpha \wedge \beta)$
T	T	T
T	F	F
F	T	F
F	F	F

1.2.4 Interpretations of a Disjunction

The truth value of a disjunction is going to be false if and only if both of its components (called *disjuncts*) are false; otherwise, it's going to be true.

α	β	$(\alpha \vee \beta)$
T	T	T
T	F	T
F	T	T
F	F	F

Note that our disjunction is going to be true if both disjuncts are true.

1.2.5 Interpretations of a Conditional

The truth value of a disjunction is going to be false if and only if the first of its components (called *antecedent*) is true but the first of its components (called *consequent*) is false; otherwise, it's going to be true.

α	β	$(\alpha \rightarrow \beta)$
T	T	T
T	F	F
F	T	T
F	F	T

Note that if its antecedent is false, a conditional is going to be true no matter what truth value its consequent takes.

1.2.6 Interpretations of a Bi-Conditional

The truth value of a bi-conditional is going to be true if and only if the truth values of its components are both true or both false; otherwise, it's going to be false.

α	β	$(\alpha \leftrightarrow \beta)$
T	T	T
T	F	F
F	T	F
F	F	T

In other words, if the truth values of its components match up, a bi-conditional is going to be true.

1.2.7 Example

Let's construct the truth table for $(((B \vee C) \rightarrow A) \leftrightarrow ((B \rightarrow A) \wedge (-A \rightarrow -C)))$. In constructing it, you need to proceed from smaller sentences to bigger ones. So, you need to fill out the truth values for $-A$ and $-C$ first. (If you're not so sure why you need to do so, read Section 1.1.3 again.)

A	B	C	$(((B \vee C) \rightarrow A) \leftrightarrow ((B \rightarrow A) \wedge (-A \rightarrow -C)))$
T	T	T	
T	T	F	
T	F	T	
T	F	F	
F	T	T	
F	T	F	
F	F	T	
F	F	F	

And then, $(B \vee C)$, $(B \rightarrow A)$, and $(-A \rightarrow -C)$.

A	B	C	$(((B$	$\vee$	$C)$	$\rightarrow$	$A)$	$\leftrightarrow$	$((B$	$\rightarrow$	$A)$	$\wedge$	$($	$-$	A	$\rightarrow$	$-$	$C)))$
T	T	T		T						T				F		T	F	
T	T	F		T						T				F		T	T	
T	F	T		T						T				F		T	F	
T	F	F		F						T				F		T	T	
F	T	T		T						F				T		F	F	
F	T	F		T						F				T		T	T	
F	F	T		T						T				T		F	F	
F	F	F		F						T				T		T	T	

And then, $((B \vee C) \rightarrow A)$ and $((B \rightarrow A) \wedge (-A \rightarrow -C))$.

A	B	C	$(((B$	$\vee$	$C)$	$\rightarrow$	$A)$	$\leftrightarrow$	$((B$	$\rightarrow$	$A)$	$\wedge$	$($	$-$	A	$\rightarrow$	$-$	$C)))$
T	T	T		T		T				T		T		F		T	F	
T	T	F		T		T				T		T		F		T	T	
T	F	T		T		T				T		T		F		T	F	
T	F	F		F		T				T		T		F		T	T	
F	T	T		T		F				F		F		T		F	F	
F	T	F		T		F				F		F		T		T	T	
F	F	T		T		F				T		F		T		F	F	
F	F	F		F		T				T		T		T		T	T	

And finally, based on what you've got in the above, you're gonna fill out the truth values for the entire sentence.

A	B	C	$(((B$	$\vee$	$C)$	$\rightarrow$	$A)$	$\leftrightarrow$	$((B$	$\rightarrow$	$A)$	$\wedge$	$($	$-$	A	$\rightarrow$	$-$	$C)))$
T	T	T		T		T		**T**		T		T		F		T	F	
T	T	F		T		T		**T**		T		T		F		T	T	
T	F	T		T		T		**T**		T		T		F		T	F	
T	F	F		F		T		**T**		T		T		F		T	T	
F	T	T		T		F		**T**		F		F		T		F	F	
F	T	F		T		F		**T**		F		F		T		T	T	
F	F	T		T		F		**T**		T		F		T		F	F	
F	F	F		F		T		**T**		T		T		T		T	T	

As you see, it turned out that $(((B \vee C) \rightarrow A) \leftrightarrow ((B \rightarrow A) \wedge (-A \rightarrow -C)))$ is a valid sentence. (For what is a valid sentence, see Section 2.3.)

1.2.8 Problems

Construct truth tables for the sentences in Problems 1.1.4.

2 Semantical Properties of (a Set of) L_1-Sentences

There are some properties of L_1-sentences which concern a semantical aspect of sentences. In short, those properties tell us how to classify a sentence according to what kind of truth-value assignments the sentence has. Let's take a look at them one by one.

2.1 Satisfiability

A sentence (or a set of sentences) is called *satisfiable* if there is at least one interpretation where the sentence (or the set of sentences) is going to be true.

Here, a clarification as to when a set of sentences is considered as true (or false) would be in order.

A set of sentences is going to be true if and only if all sentences in the set are true; otherwise, it's going to be false. Thus, even one occurrence of a false sentence in a set makes the set false no matter how many true sentences are there. In other words, the truth value for a set of sentences can be identified with that of the conjunction of the sentences in the set.

For example, the set of L_1-sentences $\{(A \to B), A\}$ is satisfiable because, in the following truth table,

A	B	$(A \to B)$	
T	T	T	$\Leftarrow$
T	F	F	
F	T	T	
F	F	T	

you find an interpretation (which is the line 1) where $A \to B$ and A are both true.

Note that any sentence letter is satisfiable. (See Section 1.2.1 for the truth table.)

Sometimes a satisfiable sentence (resp. a satisfiable set of sentences) is called a *consistent* sentence (resp. a consistent set of sentences).

2.2 Unsatisfiability

A sentence (or a set of sentences) is called *unsatisfiable* if there is no interpretation where the sentence (or the set of sentences) is going to be true. In other words, an unsatisfiable sentence (or an unsatisfiable set of sentences) is always false no matter what truth value we assign to sentence letters.

A paradigmatic example of unsatisfiable sentences would be $(A \wedge -A)$.

$($	A	$\wedge$	$-$	$A)$
	T	**F**	F	
	F	**F**	T	

What we find in the truth value assignments for $(A \wedge -A)$ is nothing but F.

Sometimes an unsatisfiable sentence (resp. an unsatisfiable set of sentences) is called an *inconsistent* sentence (resp. an inconsistent set of sentences).

2.3 Validity

Validity is the exact opposite of unsatisfiability.

A sentence (or set of sentences) is called *valid* if and only if there's no interpretation where the sentence is going to be false; in other words, a valid sentence (or a valid set of sentences) is always true in all interpretation.

A paradigmatic example of valid sentences would be $(A \vee -A)$.

$($	A	$\vee$	$-$	$A)$
	T	**T**	F	
	F	**T**	T	

What we find in the truth value assignments for $(A \vee -A)$ is nothing but T.

Sometimes a valid sentence is called a *tautology*.

2.4 Invalidity

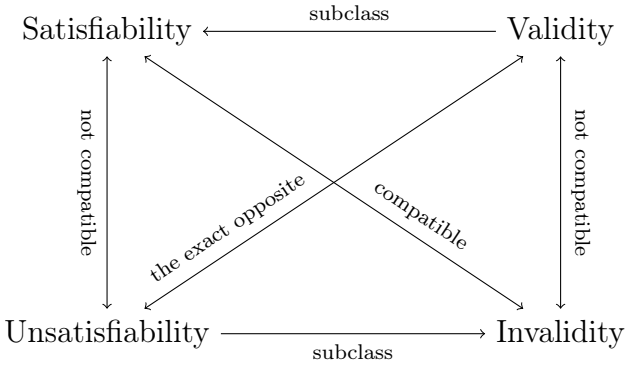
Invalidity is a kind of relatives of satisfiability.

> A sentence (or a set of sentences) is called *invalid* if there is at least one interpretation where the sentence (or the set of sentences) is going to be false.

For example, the set of L_1-sentences $\{(A \to B), A\}$ is invalid because, in its truth table (see Section 2.1), we find Fs in the lines 2–4.

2.5 Interrelations among the Properties

As you may notice from the wordings of their definitions, there are some interrelations among satisfiability/unsatisfiability/validity/invalidity as follows.



Note that the directions of the arrows matter. For example, there is a single-headed arrow from "Validity" to "Satisfiability"; this means that, together with the description "subclass" at the above of the arrow, validity is a subclass of satisfiability (meaning, every valid sentence is satisfiable but not every satisfiable sentence is valid). On the other hand, there is a double-headed arrow between "Satisfiability" and "Invalidity"; this means that, together with the description "compatible", a sentence can be both satisfiable and invalid at the same time.

2.6 Problems

Classify the following sentences into satisfiable/unsatisfiable/valid/invalid. (Note that a sentence can be classified into more than one category.)

1. $(J \to (K \to J))$
2. $((E \leftrightarrow H) \to (-E \to -H))$
3. $((((C \to D) \wedge (D \to E)) \wedge C) \wedge -E)$
4. $-(((A \vee B) \wedge (B \vee B)) \wedge (-A \wedge -B))$
5. $(-B \to ((B \vee D) \to D))$

3 Logical Implication and Logical Equivalence

3.1 Logical Implication

> α *logically implies* β if and only if there's no interpretation where α is true but β is false. In other words, α implies β if and only if a conditional $\alpha \to \beta$ is valid.

We usually omit "logically" and simply say "α implies β". In the below, I use the notation "$\Rightarrow$" for denoting logical implication. (Note that this is *my* notation; if you want to use this notation in the exam, please explicitly state that $\Rightarrow$ denotes logical implication.)

The left side of an implication can be a set of sentences; in such cases, we write $\{\alpha, \beta, \ldots, \gamma\} \Rightarrow \delta$ or $\Gamma \Rightarrow \delta$ (we use the upper Greek letters to denote a set of sentences). Obviously, here, you need to be able to assign truth values to a set of sentences. (If you're not so sure how to assign truth values to a set of sentences, see Section 2.1.)

In order to figure out whether α really implies β or not, you can use the truth table method. For example, you can figure out whether or not $-(A \to B)$ implies A just by taking a look at the following truth table.

A	B	$(A \to B)$	$-(A \to B)$
T	T	T	F
T	F	F	T
F	T	T	F
F	F	T	F

In the above table, there's no interpretation where $-(A \to B)$ is true but A is false; thus, you conclude that $-(A \to B)$ implies A.

Although the truth-table method comes in handy, writing a truth table can be quite tedious. If there's a handier way to figure out whether $\alpha \Rightarrow \beta$ or not, we want to use it. Is there any such method? Actually there is. Let's think about $((B \to C) \to (A \to B)) \Rightarrow (A \to B)$ and try to show that $((B \to C) \to (A \to B))$ doesn't imply $(A \to B)$. In order for the implication to fail, $(A \to B)$ must be false; and in order for $(A \to B)$ to be false, A must be true and B must be false. Thus, we get the following truth-value assignments.

$$((\underset{\text{F}}{B} \to C) \to \underbrace{(\underset{\text{T}}{A} \to \underset{\text{F}}{B})}_{\text{F}}) \Rightarrow \underbrace{(\underset{\text{T}}{A} \to \underset{\text{F}}{B})}_{\text{F}}$$

On the other hand, we want the left side of the implication to be true; thus, since $(A \to B)$ has been already assigned the truth value F, $(B \to C)$ must be false in order for $((B \to C) \to (A \to B))$ to be true. However, we've already assigned F to B; consequently, it's impossible to make $(B \to C)$ false no matter what truth value we assign to C because a conditional with a false antecedent never be false. Therefore, it's impossible to assign T to the left side of the implication and F to the right side; in other words, it's impossible to make the implication fail; $((B \to C) \to (A \to B))$ actually implies $(A \to B)$.

How about $(A \vee -(B \wedge C)) \Rightarrow ((A \leftrightarrow C) \vee B)$? Let's see if we can make this implication fail. In order for the implication to fail, the right side $((A \leftrightarrow C) \vee B)$ must be false; and in order for $((A \leftrightarrow C) \vee B)$ to be false, both of its conjuncts, namely $(A \leftrightarrow C)$ and B, must be false.

$$(A \vee -(\underset{\text{F}}{B} \wedge C)) \Rightarrow (\underbrace{(\underset{?}{A} \leftrightarrow \underset{?}{C})}_{\text{F}} \vee \underset{\text{F}}{B})$$

Here, for $(A \leftrightarrow C)$ to be false, there are two options; A is true and C is false, or A is false and C is true. If we choose the first option, we instantly know that the left side disjunction is going to be true. Therefore, in this case, the implication fails.

$$\underbrace{(\underset{\text{T}}{A} \vee -(\underset{\text{F}}{B} \wedge \underset{\text{F}}{C}))}_{\text{T}} \Rightarrow (\underbrace{(\underset{\text{T}}{A} \leftrightarrow \underset{\text{F}}{C})}_{\text{F}} \vee \underset{\text{F}}{B})$$

On the other hand, if we choose the second option (A is false, C is true), the truth value of the left-side disjunction is determined by $-(B \wedge C)$.

$$\underbrace{\underbrace{(\underset{\text{F}}{A} \vee \underbrace{-\underbrace{(\underset{\text{F}}{B} \wedge \underset{\text{T}}{C})}_{\text{F}}}_{\text{T}}}_{\text{T}} \Rightarrow \underbrace{(\underset{\text{F}}{A} \leftrightarrow \underset{\text{T}}{C})}_{\text{F}} \vee \underset{\text{F}}{B})}_{\text{T}}$$

In either way, the implication fails.

3.1.1 Problems

1. Prove:

1. $\alpha \Rightarrow \alpha$
2. If $\Gamma \Rightarrow \alpha$, then $\Gamma, \beta \Rightarrow \alpha$ (Here, "Γ, β" in the left side of the second implication means $\{\gamma_1, \gamma_2, \ldots, \gamma_n, \beta\}$ with $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$.
3. If $\Gamma \Rightarrow \alpha$ and $\alpha, \Delta \Rightarrow \beta$, then $\Gamma, \Delta \Rightarrow \beta$
4. $\Gamma, \alpha \Rightarrow \beta$ if and only if $\Gamma \Rightarrow (\alpha \rightarrow \beta)$
5. If $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \gamma$, then $\alpha \Rightarrow \gamma$

2. Determine whether the following implications hold. If the implication fails, provide a counterexample.

1. $\{A, (B \wedge C)\} \Rightarrow B$
2. $\{A, (B \vee C)\} \Rightarrow B$
3. $\{(K \vee J), -(K \vee J)\} \Rightarrow K$
4. $\{(W \vee J), ((W \rightarrow Z) \vee (J \rightarrow Z)), -Z\} \Rightarrow -(W \wedge J)$

3.2 Logical Equivalence

> α *logically equivalent* to β if and only if $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \alpha$; in other words, α is equivalent to β if and only if $\alpha \leftrightarrow \beta$ is valid.

We usually omit "logically" and simply say "α is equivalent to β". In the below, I use the notation "$\equiv$" for denoting logical equivalence. (Note that this is *my* notation; if you want to use this notation in the exam, please explicitly state that $\equiv$ denotes logical equivalence.)

As an example, let's prove that, if $S_1 \Rightarrow S_2$, S_1 is equivalent to $(S_1 \wedge S_2)$ and S_2 is equivalent to $(S_1 \vee S_2)$. In order to prove this, we have to consider the following four implications.

1. $S_1 \Rightarrow (S_1 \wedge S_2)$
2. $(S_1 \wedge S_2) \Rightarrow S_1$
3. $S_2 \Rightarrow (S_1 \vee S_2)$
4. $(S_1 \vee S_2) \Rightarrow S_2$

In order for the first implication to fail, $(S_1 \wedge S_2)$ must be false; and in order for the conjunction to be false, at least one of its conjuncts (namely, either S_1 or S_2) must be false. However, if S_1 is false, the implication is going to hold because, in order for the implication to fail, the left-side S_1 must be true. On the other hand, if S_2 is false, S_1 cannot be true because we're assuming the truth of $S_1 \Rightarrow S_2$. In either case, there's no interpretation where the left side is true but the right side is false.

In a similar fashion, you can make sure that there's no interpretation where the left sides of the items 2–4 are true but the right sides of the items are false. Therefore, S_1 is equivalent to $(S_1 \wedge S_2)$ and S_2 is equivalent to $(S_1 \vee S_2)$.

In this particular case, the truth-table method might be easier.

S_1	S_2	$(S_1 \wedge S_2)$	$(S_1 \vee S_2)$
T	T	T	T
~~T~~	~~F~~	~~F~~	~~T~~
F	T	F	T
F	F	F	F

Note that the situation depicted in the line 2 never happens because $S_1 \Rightarrow S_2$ (there's no interpretation where S_1 is true but S_2 is false). Thus, we can safely ignore the line 2. Obviously, the turth-value assignments of S_1 (resp. S_2) match up with those of $(S_1 \wedge S_2)$ (resp. $(S_1 \vee S_2)$). Therefore, S_1 (resp. S_2) is equivalent to $(S_1 \wedge S_2)$ (resp. $(S_1 \vee S_2)$).

Let's do one more example by constructing a truth table. This time we're gonna check whether $(A \rightarrow B)$ and $(-A \vee B)$ are logically equivalent or not.

A	B	$(A \to B)$	$(-A \vee B)$	$((A \to B) \leftrightarrow (-A \vee B))$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

The truth-value assignments for $((A \to B) \leftrightarrow (-A \vee B))$ are all true; namely, $((A \to B) \leftrightarrow (-A \vee B))$ is valid. This shows that $(A \to B)$ and $(-A \vee B)$ are logically equivalent.

3.2.1 Problems

1. Prove:

1. $--\alpha \equiv \alpha$ (Remember: "$\equiv$" is read as "is equivalent to".)
2. $\alpha \equiv -\beta$ if and only if $-\alpha \equiv \beta$
3. $-(\alpha \wedge \beta) \equiv (-\alpha \vee -\beta)$
4. $-(\alpha \vee \beta) \equiv (-\alpha \wedge -\beta)$
5. $(\alpha \to \beta) \equiv (-\alpha \vee \beta)$
6. $(\alpha \to \beta) \equiv (-\beta \to -\alpha)$
7. $(\alpha \leftrightarrow -\beta) \equiv (-\alpha \leftrightarrow \beta)$

(The above equivalences are all useful. I recommend you to memorize them.)

2. Determine whether or not the following equivalences are correct.

1. $(A \to (B \to A)) \equiv ((C \wedge -C) \vee (A \to A))$
2. $(C \wedge (B \vee A)) \equiv ((C \wedge B) \vee A)$
3. $(-(D \vee B) \to (C \to B)) \equiv (C \to (D \wedge B))$
4. $(A \to (B \to (A \to B))) \equiv (B \to (A \to (B \to A)))$
5. $(F \vee -(G \vee -H)) \equiv ((H \leftrightarrow -F) \vee G)$

4 Argument

An *argument* is comprised of two parts: a set of premises and one conclusion. An argument is called *valid* if there's no interpretation where a set of premises is true but the conclusion is false. If there's such an interpretation, an argument is called *invalid* and the interpretation is

called a *counterexample*.

We express an argument in the following form.

$$\begin{array}{|l|}\hline \text{Premise 1} \\ \text{Premise 2} \\ \vdots \\ \text{Premise } n \\ \hline \text{Conclusion} \\ \hline \end{array}$$

As is easily seen, the definitions of (valid) arguments and implications are almost identical; so, we sometimes write an argument in the form of $\{\text{Premise 1}, \text{Premise 2}, \ldots, \text{Premise } n\} \Rightarrow \text{Conclusion}$.

Let's think of the following simple argument.

$$\frac{\begin{array}{c}(A \to B) \\ A\end{array}}{B}$$

You can show its validity (or invalidity) either by the truth-table method or by the one explained above (let's call this method the *try-to-refute* method).

Truth-Table Method:

A	B	$(A \to B)$
T	T	T
T	F	F
F	T	T
F	F	T

As you see, there's no interpretation where B is false but both A and $(A \to B)$ are true.

Try-to-Refute Method:

$$\underbrace{\{\underbrace{(\underset{\text{F}}{A} \to \underset{\text{F}}{B})}_{\text{T}}, \underset{\text{F}}{A}\}}_{\text{F}} \Rightarrow \underset{\text{F}}{B}$$

In order for the above argument to be invalid, B must be false; and in order for $A \to B$ to be true with the truth value F of B, A must be false. However, such a truth-value assignment makes the set of premises false (why?); and consequently, the argument is going to be valid. In this way, we know that there's no way to assign T to the premises and F to the conclusion, which means the argument is valid.

An argument which has the form of $\{(\alpha \to \beta), \alpha\} \Rightarrow \beta$ is called *modus ponens*.

Let's do another example: $\{(A \to B), B\} \Rightarrow A$.

Truth-Table Method:

A	B	$(A \to B)$	
T	T	T	
T	F	F	
F	T	T	$\Leftarrow$
F	F	T	

Try-to-Refute Method:

$$\{\underbrace{\underbrace{(\underset{\text{F}}{A} \to \underset{\text{T}}{B})}_{\text{T}}, \underset{\text{T}}{B}}_{\text{T}}\} \Rightarrow \underset{\text{F}}{A}$$

In both ways, we find the interpretation which assigns T to the premises and F to the conclusion. Further, we know that it happens when A is false and B is true; that truth-value assignment would be your counterexample to the argument.

Here's a more complicated example: $\{(A \leftrightarrow (-B \vee C)), (C \leftrightarrow -A)\} \Rightarrow -A$.

Truth-Table Method:

A	B	C	$-A$	$-B$	$(-B \vee C)$	$(A \leftrightarrow (-B \vee C))$	$(C \leftrightarrow -A)$	
T	T	T	F	F	T	T	F	
T	T	F	F	F	F	F	T	
T	F	T	F	T	T	T	F	
T	F	F	**F**	T	T	**T**	**T**	$\Leftarrow$
F	T	T	T	F	T	F	T	
F	T	F	T	F	F	T	F	
F	F	T	T	T	T	F	F	
F	F	F	T	T	T	F	T	

In the line 4, we find a counterexample (A: T, B: F, C: F) which makes all the premises true but the conclusion false.

Try-to-Refute Method:

$$\{(\underset{\text{T}}{A} \leftrightarrow \underbrace{(\underbrace{-\underset{\text{F}}{B}}_{\text{T}} \vee \underset{\text{F}}{C})}_{\text{T}}), \underbrace{(\underset{\text{F}}{C} \leftrightarrow \underbrace{-\underset{\text{F}}{A}}_{\text{F}})}_{\text{T}}\} \Rightarrow \underbrace{-\underset{\text{T}}{A}}_{\text{F}}$$

In order for the argument to be invalid, $-A$ must be false; thus, A must be true. On the other hand, all the premises must be true. In order for the premises to be all true, both $(A \leftrightarrow (B \vee C))$ and $(C \leftrightarrow -A)$ must be true. In order for $(C \leftrightarrow -A)$ to be true, the truth value of C must match up with that of $-A$; thus, C must be false. In order for $(A \leftrightarrow (B \vee C))$ to be true, the truth values of the both sides of $\leftrightarrow$ must match up; and since the one side of the bi-conditional, namely A, is true, $(-B \vee C)$ must be true as well. Since C is false, in order for $(-B \vee C)$ to be true, $-B$ must be true; thus, B must be false. In this truth-value assignment (A: T, B: F, C: F), we successfully assigned T to the premises and F to the conclusion, which means the argument is invalid.

4.1 Problems

1. Prove:

1. $\{(\alpha \vee \beta), -\alpha\} \Rightarrow \beta$
2. $\{(\alpha \rightarrow \beta), -\beta\} \Rightarrow -\alpha$

3. $\{(\alpha \to \gamma), (\beta \to \gamma), (\alpha \vee \beta)\} \Rightarrow \gamma$

(The above forms of arguments are called *disjunctive syllogism*, *modus tollens*, and *constructive dillema* respectively.)

2. Determine whether or not the following arguments are valid.

1. $\{(B \vee (A \wedge -C)), ((C \to A) \leftrightarrow B), (-B \vee A)\} \Rightarrow -(A \vee C)$
2. $\{-(Y \leftrightarrow A), -Y, -A\} \Rightarrow (W \wedge -W)$
3. $\{(B \vee B), ((-B \to (-D \vee -C)) \wedge ((-D \vee C) \vee (B \vee C)))\} \Rightarrow C$
4. $\{(((J \wedge T) \wedge Y) \vee (-J \to -Y)), (J \to T), (T \to Y)\} \Rightarrow (Y \leftrightarrow T)$
5. $\{((A \wedge (B \vee C)) \leftrightarrow (A \vee B)), (B \to -B)\} \Rightarrow (C \vee A)$

5 Some Words on the Implication/Argument

I noticed that a considerable number of students are still having trouble with the following aspects.

1. The difference between the validities of a sentence and of an argument.
2. The condition in which an implication (resp. argument) holds.

In this section, I like to arouse such students' attention.

5.1 The Difference between the Validities of a Sentence and of an Argument

Let's replicate the definitions where the same word "valid" appears.

> A sentence (or set of sentences) is called *valid* if and only if there's no interpretation where the sentence is going to be false; in other words, a valid sentence is always true in all interpretation.

> An argument is called *valid* if there's no interpretation where a set of premises is true but the conclusion is false.

Some students seem to confuse the definition of the validity of a sentence with that of the validity of an argument and think that an argument

is going to be valid if and only if both the premises and the conclusion are true in all interpretations. Surely, its premises and its conclusion are always true, an argument is going to be valid; however, this is not the only case where the argument is going to be valid. Let's see when an argument is going to be valid in the next section.

5.2 The Condition in Which an Implication and an Argument Hold

First, let's take a look again at the definitions of implication and argument.

> α *logically implies* β if and only if there's no interpretation where α is true but β is false.

> An argument is called *valid* if there's no interpretation where a set of premises is true but the conclusion is false.

Note that they don't explicitly tell us when an implication or an argument holds. However, they *implicitly* tell us when α implies β or an argument is going to be valid; as long as there's no interpretation where α (resp. a set of premises) is true but β (resp. the conclusion) is false, an implication (resp. argument) holds. More explicitly, α implies β if the configuration of the truth values for α and β take one of the following forms.

$$\mathrm{T} \Rightarrow \mathrm{T},\ \mathrm{F} \Rightarrow \mathrm{T},\ \mathrm{F} \Rightarrow \mathrm{F}$$

Likewise, an argument is going to be valid if the configuration of the truth values for a set of premises and the conclusion take one of the following forms.

$$\frac{\mathrm{T}}{\mathrm{T}} \quad \frac{\mathrm{F}}{\mathrm{T}} \quad \frac{\mathrm{F}}{\mathrm{F}}$$

When you think about the situations where α implies β or the situations where an argument is valid, please don't forget the cases $\mathrm{F} \Rightarrow \mathrm{T}$, $\mathrm{F} \Rightarrow \mathrm{F}$, $\frac{\mathrm{F}}{\mathrm{T}}$, and $\frac{\mathrm{F}}{\mathrm{F}}$.

6 Complete Disjunctive Normal Form

6.1 Truth-Functional Connectives

By now we know perfectly well that there are four combinations of the truth values for a pair of sentence letters α, β: (T, T), (T, F), (F, T), and (F, F). Those combinations of the truth values give us different truth-value assignments according to which connective we're thinking of. For example, if we're thinking of $\wedge$, the truth-value combination (T, F) gives us the truth value F; and if we're thinking of $\vee$, it gives us T.

From the above consideration, we notice that we can think of a connective in terms of a *mathematical function*. A mathematical function receive some inputs and returns some outcome. For example, the conjunction $\wedge$ can be thought of as a function which receive two inputs and return one outcome: $f_{\wedge}(\alpha, \beta)$. This function gives us T when we input T to both α and β; in other cases, it gives us F.

Is there any mechanical (i.e. algorithmic) way to express an *arbitrary* function in our language L_1? In other words, if we're given some sentence letters and truth-value assignments to each combination of those letters, can we find an L_1-sentence which meet those specifications? In simple cases, we may be able to find such a sentence by trial and error. For example, we're give the following truth table, we can manage to figure out that it can be expressed as $(A \vee -(A \vee B))$ (and in turn, as $(B \to A)$).

A	B	γ
T	T	T
T	F	T
F	T	F
F	F	T

But what if we're given the following truth table and asked to construct an L_1-sentence which has the truth-value assignments given in the table?

A	B	C	δ
T	T	T	T
T	T	F	F
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	T
F	F	T	F
F	F	F	T

We're at a loss.

6.2 Complete Disjunctive Normal Form

Luckily, we have a mechanical method for finding an L_1-sentence which meets a given specification. In order to explain what that method is (and how it works), let's go back to the above example.

The sentence δ defined in the above truth table is going to be true if all the sentence letters in it are true. Thus, this situation can be expressed as $(A \wedge B \wedge C)$ because this sentence is going to be true only when we assign T to (A, B, C). From the line 4, we also notice that the sentence δ is going to be true when $(A \wedge -B \wedge -C)$ is true (why?). In this way, we know that δ is going to be true when at least one of the following L_1-sentences is true: $(A \wedge B \wedge C), (A \wedge -B \wedge -C), (-A \wedge B \wedge -C), (-A \wedge -B \wedge -C)$. Thus, we can express δ as $((A \wedge B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge -C) \vee (-A \wedge -B \wedge -C))$. This is exactly what is called a *complete disjunctive normal form.*

A *complete disjunctive normal form* of a sentence is
1. a disjunction
2. each of whose disjuncts is a conjunction
3. each of whose conjuncts is comprised of all the sentence letters, possibly with the denial, of the target sentence possibly with the denial
4. each of which appears only once in each conjunction.

Let's compare what we've got in the above with this definition. $((A \wedge B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge -C) \vee (-A \wedge -B \wedge -C))$ is obviously a disjunction; also, its disjuncts are all conjunctions; and these conjunctions are

comprised of the sentence letters and/or their denials of the target sentence δ; therefore, $((A \wedge B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge -C) \vee (-A \wedge -B \wedge -C))$ has a complete disjunctive normal form.

In fact, given any sentence (except one special case. We're gonna look at such a special case below), we can transform it into a complete disjunctive normal form in a mechanical way. Let's take $((A \wedge B) \vee -A)$ as an example. First, you need to construct its truth table.

A	B	$-A$	$(A \wedge B)$	$((A \wedge B) \vee -A)$
T	T	F	T	T
T	F	F	F	F
F	T	T	F	T
F	F	T	F	T

And then, look at the interpretations where the sentence $((A \wedge B) \vee -A)$ is going to be true. In our case, the sentence is going to be true in the lines 1, 3, and 4. In the line 1, the sentence letters A, B take the truth value T; and this situation can be expressed by $(A \wedge B)$. Similarly, the situations depicted in the lines 3–4, namely where the sentence letters A, B take the truth value (F, T) and (F, F) respectively, can be expressed as $-A \wedge B$ and $-A \wedge -B$ respectively ($(-A \wedge B)$ is going to be true if and only if A is false and B is true, and $(-A \wedge -B)$ is going to be true if and only if both A and B are false). Since $((A \wedge B) \vee -A)$ is going to be true in one of those situations, the sentence can be expressed as $((A \wedge B) \vee (-A \wedge B) \vee (-A \wedge -B))$ which is a complete disjunctive normal form.

To recap:

If you're asked to transform a sentence α into its complete disjunctive normal form,

1. Construct a truth table for the target sentence α.
2. Find rows where the target sentence is true.
3. In each row you've found above, look at the truth-value assignments for the sentence letters, and create a conjunction based on those truth-value assignments as follows.

   α. If the truth-value assignment for the sentence letter is true, the sentence letter itself is one of your conjuncts.

   β. If the truth-value assignment for the sentence letter is false, the sentence letter with negation is one of your conjuncts.

γ. Connect what you've got above with $\wedge$.

4. Connect the conjunctions you've got above with the disjunctive connective.

Now, let's consider the following truth table for $(A \wedge -A)$ and construct a complete disjunctive normal form out of it.

A	B	$(A \wedge -A)$
T	T	F
T	F	F
F	T	F
F	F	F

As is easily seen, there's no row where $(A \wedge -A)$ is going to be true; $(A \wedge -A)$ is an unsatisfiable sentence. However, according to the above instruction, we have to take a look at the rows where the target sentence (in this case, $(A \wedge -A)$) is going to be true. There's no row to look at. What should we do? In fact, there's no complete disjunctive normal form corresponding to an unsatisfiable sentence. Thus, if you're asked to construct a complete disjunctive normal form of an unsatisfiable sentence, simply reply, "There's no complete disjunctive normal form of it".

6.2.1 Problems

Find complete disjunctive normal forms for the following sentences.

1. $(-(A \rightarrow B) \vee (-A \wedge C))$
2. $((A \vee B) \wedge (-B \vee C))$
3. $((A \wedge -B) \vee (A \wedge C))$
4. $(-A \vee (B \rightarrow -C))$
5. $((A \vee B) \leftrightarrow -C)$

7 Expressive Completeness

Now we know that any sentence can be expressed in a disjunctive normal form; since a disjunctive normal form comprises of sentence letters, $-$, $\wedge$, and $\vee$, we also know that these connectives are sufficient to express any sentence. We call such a set of connective *expressively complete*.

A set of connectives is called *expressively complete* if you can express any sentence (i.e. any possible truth-value assignments) with the connectives in the set.

From Problems 3.2.1, we know that $\wedge$ can be defined in terms of $-$ and $\vee$. This means that $-$ and $\vee$ are sufficient for expressing any sentence; the set $\{-, \vee\}$ is expressively complete.

Given an expressively complete set Γ of connectives; if you can express the set Γ in terms of another set Δ of connectives, Δ is also expressively complete.

Is $\{-, \vee\}$ the smallest expressively complete set? There are actually two expressively complete set each of which is comprised of just one connective: $|$ (the *Shffer stroke*) and $\downarrow$ (the *Peirce arrow*). The truth tables for them are as follows.

α	β	$(\alpha \mid \beta)$
T	T	F
T	F	T
F	T	T
F	F	T

α	β	$(\alpha \downarrow \beta)$
T	T	F
T	F	F
F	T	F
F	F	T

Let's focus on the Sheffer stroke. As you may notice, the truth-value assignments for $(\alpha|\beta)$ are exactly the opposite of those of $(\alpha \wedge \beta)$. With this, we know that $(\alpha \wedge \beta)$ can be expressed as $-(\alpha|\beta)$. Thus, if we figure out how to express "$-$" (denial) with the Sheffer stroke alone, we can express "$\wedge$" with the stroke alone as well. So let's figure it out.

From the above truth table, we know that $(\alpha|\beta)$ is going to be true if α and β are both false and it's gonna be false if α and β are both true. Now let's replace β in the above table with α and see what's gonna happen.

α	$(\alpha \mid \alpha)$	$(\equiv -\alpha)$
T	F	
F	T	

This is exactly what we want for the denial.

Since we now know how to express "$-$" with the Sheffer stroke alone, we also know how to express "$\wedge$" with the stroke alone.

α	β	$((\alpha\|\beta)\|(\alpha\|\beta))$	$(\equiv (\alpha \wedge \beta))$
T	T	T	
T	F	F	
F	F	F	
F	F	F	

And we now know how to express "$\vee$" with $\{-, \wedge\}$ (recall that $(\alpha \vee \beta) \equiv -(-\alpha \wedge -\beta)$ and $(\alpha \wedge \beta) \equiv -(\alpha|\beta)$), we also know how to express "$\vee$" (and consequently "$\rightarrow$") with the Sheffer stroke alone.

In a similar fashion, you can show that the Peirce arrow $\downarrow$ is expressively complete.

7.1 Problems

1. Express $\vee, \rightarrow, \downarrow$ in terms of $|$.
2. Express $-, \wedge, \vee, \rightarrow, |$ in terms of $\downarrow$.
3. Express $\vee, \rightarrow, |, \downarrow$ in terms of $\{-, \wedge\}$.
4. Express $\wedge, \rightarrow, |, \downarrow$ in terms of $\{-, \vee\}$.
5. Express $\wedge, \vee, |, \downarrow$ in terms of $\{-, \rightarrow\}$.

8 The Truth-Tree Method

To figure out the satisfiability/unsatisfiability/validity/invalidity of a sentence, the validity of an argument, or the truth of a logical implication/equivalence, the truth-table method comes in handy. However, as the number of sentence letters in a sentence increases, constructing a truth table becomes a substantial task. It would be nice if there's a handier way to figure out the above properties of a sentence or an argument. The *truth-tree method* provides such a handier way.

A *truth tree* is, basically, a decomposed form of a set of sentences into literals. (Remember: literals refer to sentence letters and their denials. See 1.1.2.) In writing it, your tree grows, maybe branching into paths, and eventually stops growing.

By decomposing sentences into literals, it enables us to easily find whether a set of sentences is satisfiable (consistent) or unsatisfiable (inconsistent). If you find at least one contradiction (a contradictory pair of sentences like A and $-A$; namely, a pair of a sentence letter and its own denial) in each path of your tree, your set of sentences is unsatisfiable. On the other hand,

after finish writing your tree, if there's at least one path immune from contradiction, your set of sentences is satisfiable. The truth-tree method tells us how to decompose a sentence into its components for that purpose.

8.1 10 Rules of the Truth-Tree Method

There are 10 rules you need to memorize.

2 Rules for Denial

1. α
   $-\alpha$
   $\times$

2. $\checkmark$ $--\alpha$
   α

2 Rules for Conjunction

3. $\checkmark$ $(\alpha \wedge \beta)$
   α
   β

4. $\checkmark$ $-(\alpha \wedge \beta)$
   $-\alpha \quad -\beta$

2 Rules for Disjunction

5. $\checkmark$ $(\alpha \vee \beta)$
   $\alpha \quad \beta$

6. $\checkmark$ $-(\alpha \vee \beta)$
   $-\alpha$
   $-\beta$

2 Rules for Conditional

7. $\checkmark$ $(\alpha \rightarrow \beta)$
   $-\alpha \quad \beta$

8. $\checkmark$ $-(\alpha \rightarrow \beta)$
   α
   $-\beta$

2 Rules for Bi-Conditional

9. $\checkmark$ $(\alpha \leftrightarrow \beta)$
   $\alpha \quad -\alpha$
   $\beta \quad -\beta$

10. $\checkmark$ $-(\alpha \leftrightarrow \beta)$
    $\alpha \quad -\alpha$
    $-\beta \quad \beta$

Let's see how these rules work by doing some examples.

8.2 Testing the Satisfiability of sentences

To test the satisfiability of a set of sentences, start drawing a tree with the sentences themselves. If all paths of the tree are closed, the set is unsatisfiable; if there's at least one open path in the finished tree (namely, the tree which stops growing), the set is satisfiable; and such an open path tells you an interpretation which makes the set true.

Let's start with a simple example: $(A \wedge -B), C, (-A \vee -B)$.
You start drawing your tree by listing your sentences up vertically.

1.	$(A \wedge -B)$
2.	C
3.	$(-A \vee -B)$

And then, apply one of the rules to one of the sentences. Here, we have two options: apply the rule 3 to the line 1, or apply the rule 5 to the line 3. In general, if you can apply a non-branching rule (like the rules 3, 6, 8), apply it.

Rule of Thumb 1: Apply a non-branching rule when you can.

Let's apply the non-branching rule 3 to the line 1. (In the below I wrote down which rule is applied to which line. This is purely for pedagogical purpose. You don't have to do it in the exam or in the assignments.)

1.	✓ $(A \wedge -B)$	
2.	C	
3.	$(-A \vee -C)$	
4.	A	Rule 3 to Line 1
5.	$-B$	Rule 3 to Line 1

After applying a rule to a sentence, you should put a check mark to the left or right of the sentence in order to avoid applying a rule again to it.

Then, there's only one option left: apply the rule 5 to the line 3.

1.	✓ $(A \wedge -B)$	
2.	C	
3.	✓ $(-A \vee -C)$	
4.	A	Rule 3 to Line 1
5.	$-B$	Rule 3 to Line 1
6.	$-A$ $\quad$ $-C$	Rule 5 to Line 3

All sentences except literals are check-marked; there's no sentence left to which our rules can be apply. We finish drawing the tree. The next task is to find a contradiction (a sentence letter and its own denial) in the paths. In this example, it's obvious; we can easily find two contradictions: A in the line 4 and $-A$ in the line 6, and C in the line 2 and $-C$ in the line 6. Therefore, this set of sentences is not satisfiable; in other words, it's unsatisfiable.

> If you find a contradiction, you should put $\times$ under the path where you find it. When you do this, we say that you close the path. And if all the paths are closed, we say that the tree is closed.

1.	✓ $(A \wedge -B)$	
2.	C	
3.	✓ $(-A \vee -C)$	
4.	A	Rule 3 to Line 1
5.	$-B$	Rule 3 to Line 1
6.	$-A$ $\quad$ $-C$	Rule 5 to Line 3
	$\times$ $\quad$ $\times$	
	4,6 $\quad$ 2,6	

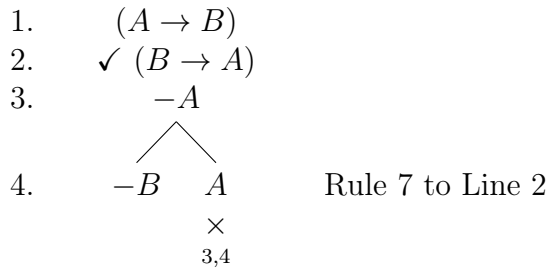
(In the above I wrote down which line and which line are contradictory. Again, this is just for pedagogical purpose. You don't have to do it in the exam or in the assignments.)

Let's do another example: the satisfiability check for $\{(A \to B), (B \to A), -A\}$. As before, we start drawing our tree by listing up the sentences vertically.

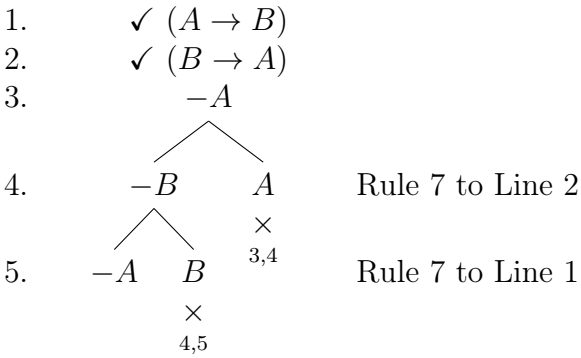
1.	$(A \to B)$
2.	$(B \to A)$
3.	$-A$

This time, unfortunately, there's no non-branching rule we can apply. Thus, it seems it doesn't matter which rule we apply first. However, if we apply the rule 7 to the line 2, we immediately come across a contradiction and we can close the path.

Rule of Thumb 2: Apply the rule which leads to a contradiction.



There's one more sentence to decompose in the tree.



We decomposed all the sentences except literals; we finish drawing the tree. However, the left-most path remains open; meaning, the set of sentences is satisfiable. Then, under what truth-value assignments is it going to be true? To figure this out, we need to go back the path to its root, and collecting literals along the way. From $-A$ to $(A \to B)$, we find the literals $-A$ and $-B$. That means that the set of sentences is going to be true when $-A$ and $-B$ are both true; in other words, the set of sentences is going to be true when A and B are both false (check this by putting F to A and B in the sentences).

One more example: $\{(A \to (B \leftrightarrow C)), -(C \to A)\}$. Since the rules for $-(C \to A)$ is non-branching, so we apply the rule 8 first.

1.	$\checkmark\ (A \to (B \leftrightarrow C))$	
2.	$\checkmark\ -(C \to A)$	
3.	C	Rule 8 to Line 2
4.	$-A$	Rule 8 to Line 2
5.	$-A \quad (B \leftrightarrow C)$	Rule 7 to Line 1

At this point, we notice that there's nothing to do for the left path because we decomposed all the sentences which are in the "root" positions as to the left path. It remains open no matter what happens on the right path. (Note that $(B \leftrightarrow C)$ belongs to the different path; we can't decompose it under $-A$ on the left path.) Thus, we can stop here (unless you're required to find all the open paths).

Rule of Thumb 3: You can stop decompose the tree once you find an open path (unless you're required to find all the open paths).

As in the previous example, we collect literals on the open path and find $-A$ and C. From this, we know that the set is going to be true when A is false and C is true. (You may wonder, "What about the truth value for B?" In this case, it really doesn't matter which truth value B takes because the truth value for $(A \to (B \leftrightarrow C))$ (the only sentence in which B appears) is already determined by that of A.)

According to Rule of Thumb 3 above, we can stop decomposing the tree here because we already know the set of sentences is satisfiable. Even so, let's keep decomposing (yet again, for pedagogical purpose).

1.	$\checkmark\ (A \to (B \leftrightarrow C))$	
2.	$\checkmark\ -(C \to A)$	
3.	C	Rule 8 to Line 2
4.	$-A$	Rule 8 to Line 2
5.	$-A \quad (B \leftrightarrow C)$	Rule 7 to Line 1
6.	$B \quad -B$	
7.	$C \quad -C$	
	$\times$ 3,7	

It turns out that the tree has one more open path and it tells us that the set is going to be true when A is false, B is true, and C is true.

If your tree has at least one open path, we say that the the tree is open.

8.3 Testing the Validity of a Sentence

> To test the validity of a sentence, test the satisfiability of the denial of the sentence. If the tree is closed, the sentence is valid; if not, it's not. In the latter case, an open path gives you an interpretation in which the sentence is going to be false.

Example: Check the validity of $((C \to R) \to (-R \to -(C \wedge J)))$. We should start our tree with the *denial* of the sentence.

1.	✓ $-((C \to R) \to (-R \to -(C \wedge J)))$	
2.	✓ $(C \to R)$	Rule 8 to Line 1
3.	✓ $-(-R \to -(C \wedge J))$	Rule 8 to Line 1
4.	$-R$	Rule 8 to Line 3
5.	✓ $--(C \wedge J)$	Rule 8 to Line 3
6.	✓ $(C \wedge J)$	Rule 2 to Line 6
7.	C	Rule 3 to Line 6
8.	J	Rule 3 to Line 6
9.	$-C$ $\quad$ R × $\quad$ × 7,9 $\quad$ 4,9	Rule 7 to Line 2

All the paths are closed; $((C \to R) \to (-R \to -(C \wedge J)))$ is valid.

One more example: Check the validity of $((L \vee (J \vee -K)) \wedge (K \wedge ((J \vee L) \to -K)))$.

1.	✓ $-((L \vee (J \vee -K)) \wedge (K \wedge ((J \vee L) \rightarrow -K)))$		
2.	✓ $-(L \vee (J \vee -K))$	$-(K \wedge ((J \vee L) \rightarrow -K))$	Rule 4 to Line 1
3.	$-L$		Rule 6 to Line 2
4.	✓ $-(J \vee -K)$		Rule 6 to Line 2
5.	$-J$		Rule 6 to Line 4
6.	✓ $--K$		Rule 6 to Line 4
7.	K		Rule 2 to Line 6

At this point, we notice that there's nothing to do for the left path because we decomposed all the sentences which are in the "root" positions as to the left path. The left path remains open no matter what happens to the rest. $((L \vee (J \vee -K)) \wedge (K \wedge ((J \vee L) \rightarrow -K)))$ is going to be false when J is false, K is true, and L is false.

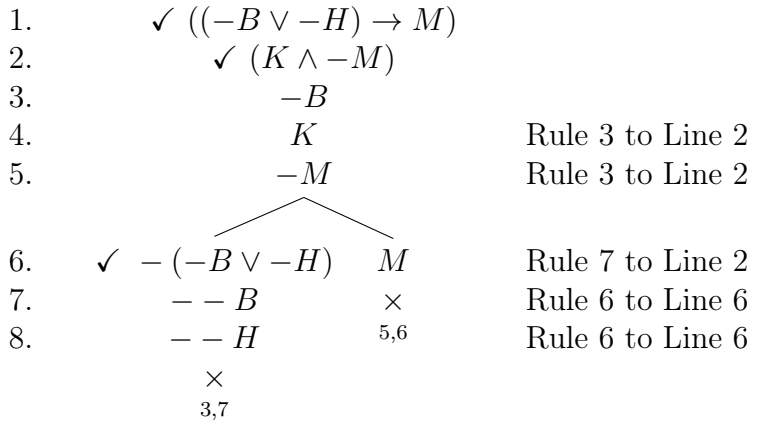
8.3.1 Problems

Do Problems 2.6 by the truth-tree method.

8.4 Testing the Validity of an Argument

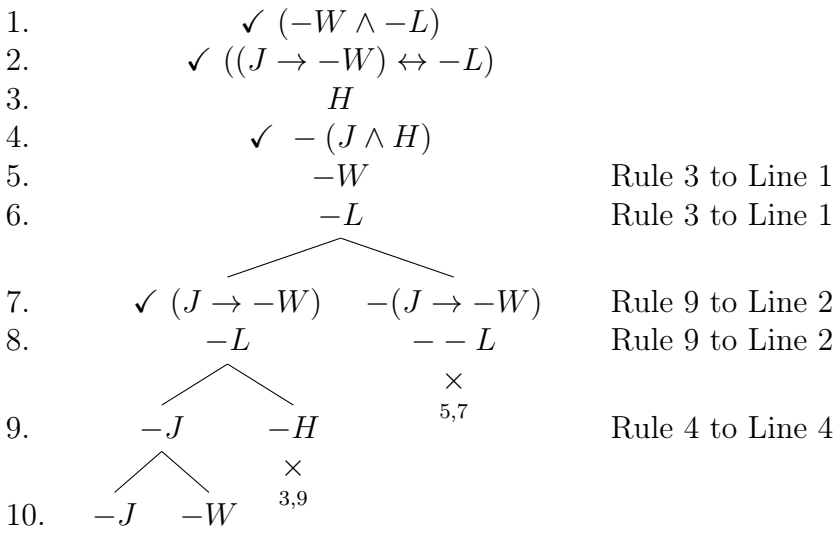
> To test the validity of an argument, test the satisfiability of the premises and the denial of the conclusion. If the tree is closed, the argument is valid; if not, it's not. In the latter case, an open path gives you a counterexample to the argument.

Example: Check the validity of the argument $\{((-B \vee -H) \rightarrow M), (K \wedge -M)\} \Rightarrow B$.



It's closed; the argument is valid.

Another example: Check the validity of $\{(-W \wedge -L), ((J \rightarrow -W) \leftrightarrow -L), H\} \Rightarrow (J \wedge H)$



We have two open paths according to both of which the arguments is invalid when H is true and J, L, W are false.

8.4.1 Problems

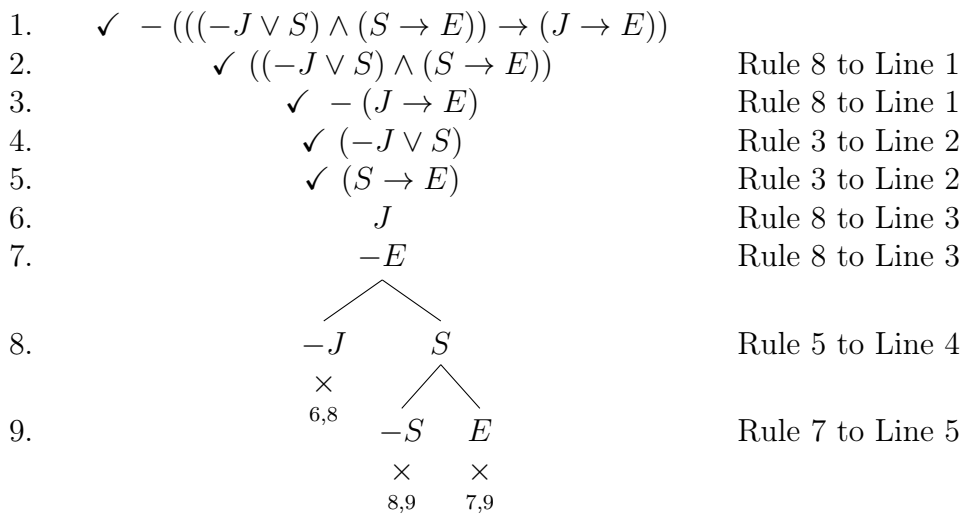
Do Problems 4.1 (2) by the truth-tree method.

8.5 Testing the Implication

Recall that an implication $\alpha \Rightarrow \beta$ holds if and only if $\alpha \rightarrow \beta$ is valid (see Section 3.1). Thus, checking the truth of the implication $\alpha \Rightarrow \beta$ amounts to that of the validity of $\alpha \rightarrow \beta$.

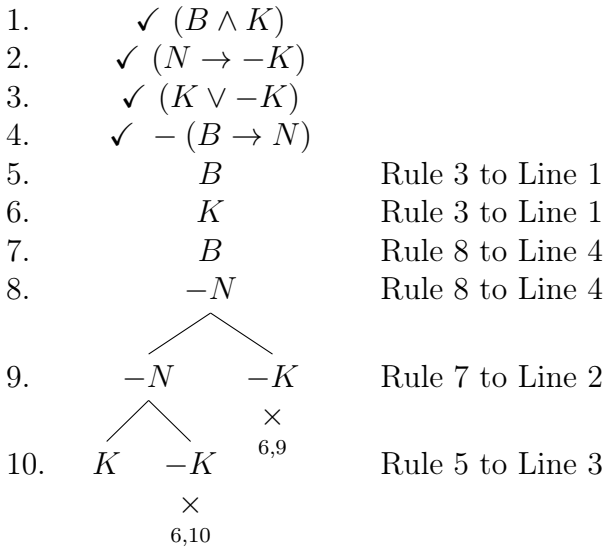
> To test the truth of an implication $\alpha \Rightarrow \beta$, test the validity of $\alpha \rightarrow \beta$. If the tree is closed, the implication holds; if not, it's not. In the latter case, an open path gives us an instance where the implication fails.

Example: Check the truth of the implication $\{(-J \vee S), (S \rightarrow E)\} \Rightarrow (J \rightarrow E)$. (Remember: the truth of a set of sentences amounts to that of the conjunction of the sentences.)



The tree is close; the implication $\{(-J \vee S), (S \rightarrow E)\} \Rightarrow (J \rightarrow E)$ actually holds.

Another example: $\{(B \wedge K), (N \rightarrow -K), (K \vee -K)\} \Rightarrow (B \rightarrow N)$ (This time I start the tree by vertically aligning the sentences of the left side and the denial of the sentence of the right side. You might not want to adopt this way in the exam unless you can justify this based on the definitions of implication and argument.)



We have an open path according to which the implication fails when B, K are true and N is false.

8.5.1 Problems

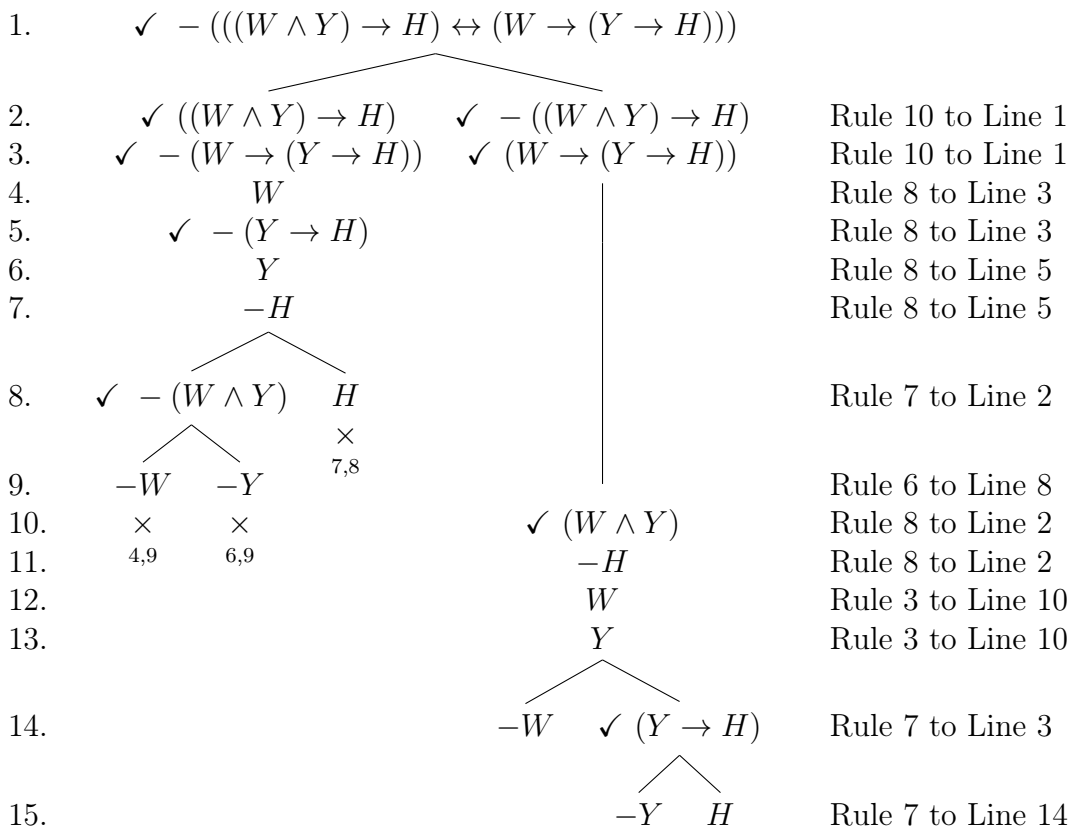
Do Problems 3.1.1 (2) by the truth-tree method.

8.6 Testing the Equivalence

Recall that an equivalence $\alpha \equiv \beta$ holds if and only if $\alpha \leftrightarrow \beta$ is valid (see Section 3.2). Thus, checking the truth of the equivalence $\alpha \equiv \beta$ amounts to that of the validity of $\alpha \leftrightarrow \beta$.

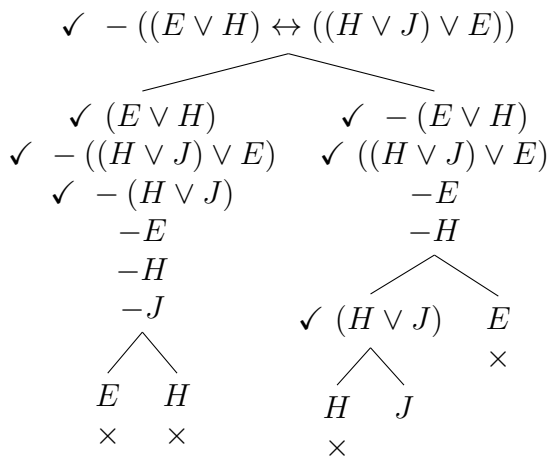
> To test the truth of an equivalence $\alpha \equiv \beta$, test the validity of $\alpha \leftrightarrow \beta$. If the tree is closed, the equivalence holds; if not, it's not. In the latter case, an open path gives us an instance where the equivalence fails.

Example: Check the truth of $((W \wedge Y) \rightarrow H) \equiv (W \rightarrow (Y \rightarrow H))$.



All paths are closed; the implication holds. (For some reasons, I couldn't display the closing marks under $-W, -Y, H$, but I guess it's obvious.)

Another (and last) example: $\{(E \vee H) \equiv ((H \vee J) \vee E)\}$. This time I omit the justifications and other annotations.



One path remains open; the equivalence doesn't hold when E, H are false, and J is true.

8.6.1 Problems

Do Problems 3.2.1 by the truth-tree method.

Solutions

Solutions for Problems 1.1.4

1. $\rightarrow$
2. $\wedge$
3. $\vee$
4. $\rightarrow$
5. $-$

Solutions for Problems 1.2.8

1.

A	B	$-A$	S_1 $(--A \wedge B)$	S_2 $(-A \leftrightarrow B)$	$(S_1 \rightarrow S_2)$
T	T	F	T	F	F
T	F	F	F	T	T
F	T	T	F	T	T
F	F	T	F	F	T

2.

D	E	H	$-D$	$-H$	S_1 $(D \wedge E)$	S_2 $(-H \vee S_1)$	$(-D \wedge S_2)$
T	T	T	F	F	T	T	F
T	T	F	F	T	T	T	F
T	F	T	F	F	F	F	F
T	F	F	F	T	F	T	F
F	T	T	T	F	F	F	F
F	T	F	T	T	F	T	T
F	E	T	T	F	F	F	F
F	E	F	T	T	F	T	T

3.

A	B	D	$-A$	$-B$	$-D$	S_1 $(-A \wedge B)$	S_2 $(-D \vee -B)$	S_3 $(D \leftrightarrow S_1)$	S_4 $-S_3$	$(S_4 \vee S_2)$
T	T	T	F	F	F	F	F	F	T	T
T	T	F	F	F	T	F	T	T	F	T
T	F	T	F	T	F	F	T	F	T	T
T	F	F	F	T	T	F	T	T	F	T
F	T	T	T	F	F	F	F	F	T	T
F	T	F	T	F	T	F	T	T	F	T
F	F	T	T	T	F	T	T	T	F	T
F	F	F	T	T	T	T	T	F	T	T

4.

E	F	J	$-E$	$-F$	$-J$	S_1 $(E \vee F)$	S_2 $(-E \wedge -F)$	S_3 $(S_1 \wedge S_2)$	S_4 $(J \wedge S_3)$	$(S_4 \rightarrow -J)$
T	T	T	F	F	F	T	F	F	F	T
T	T	F	F	F	T	T	F	F	F	T
T	F	T	F	T	F	T	F	F	F	T
T	F	F	F	T	T	T	F	F	F	T
F	T	T	T	F	F	T	F	F	F	T
F	T	F	T	F	T	T	F	F	F	T
F	F	T	T	T	F	F	T	F	F	T
F	F	F	T	T	T	F	T	F	F	T

5.

A	B	C	$-A$	S_1 $(B \wedge C)$	S_2 $(A \leftrightarrow S_1)$	S_3 $-S_2$	S_4 $(B \leftrightarrow S_3)$	S_5 $-S_4$	S_6 $(-A \leftrightarrow S_5)$	$-S_6$
T	T	T	F	T	T	F	F	T	F	T
T	T	F	F	F	F	T	T	F	T	F
T	F	T	F	F	F	T	F	T	F	T
T	F	F	F	F	F	T	F	T	F	T
F	T	T	T	T	F	T	T	F	F	T
F	T	F	T	F	T	F	F	T	T	F
F	F	T	T	F	T	F	T	F	F	T
F	F	F	T	F	T	F	T	F	F	T

Solutions for Problems 2.6

1. Valid/Satisfiable.
2. Valid/Satisfiable.
3. Invalid/Unsatisfiable.

4. Valid/Satisfiable.
5. Valid/Satisfiable.

Solutions for Problems 3.1.1

1. In order for this implication to fail, the right side α has to be false. However, if the right side α is false, so is the left side α. There's no way to make the left side α true and the right side α false. Therefore, the implication holds.

2. Suppose that $\Gamma \Rightarrow \alpha$ holds but $\Gamma, \beta \Rightarrow \alpha$ fails. In order to make this happen, α has to be false. Then, Γ has to be false as well; otherwise, $\Gamma \Rightarrow \alpha$ is going to fail. Now, if Γ is false, so is $\{\Gamma, \beta\}$ regardless of the truth value of β. Then, $\Gamma, \beta \Rightarrow \alpha$ is going to be valid. However, this is contradictory to our assumption. Therefore, there's no way to make $\Gamma \Rightarrow \alpha$ hold but $\Gamma, \beta \Rightarrow \alpha$ fail. This proves the statement.

3. Suppose that $\Gamma, \Delta \Rightarrow \beta$ fails. This means that Γ, Δ are true and β is false. Now, we're supposing the truth of $\Gamma \Rightarrow \alpha$ and $\alpha, \Delta \Rightarrow \beta$. In order for the latter to be th case, either α or Δ has to be true. However, we're assuming the truth of Δ. Therefore, α has to be false. However, if α is false, $\Gamma \Rightarrow \alpha$ is going to fail. This contradicts the assumption that $\Gamma \Rightarrow \alpha$ holds. Therefore, if $\Gamma \Rightarrow \alpha$ and $\alpha, \Delta \Rightarrow \beta$ hold, $\Gamma, \Delta \Rightarrow \beta$ must hold as well.

4. Suppose that $\Gamma \Rightarrow (\alpha \rightarrow \beta)$ fails. Then, Γ and α have to be true, and β has to be false. If so, $\Gamma, \alpha \Rightarrow \beta$ fails as well. This contradicts the assumption. On the other hand, from the supposition that $\Gamma, \alpha \Rightarrow \beta$ fails, we can draw the same conclusion. Therefore, the statement holds.

5. Suppose that $\alpha \Rightarrow \gamma$ fails. If so, α has to be true and γ has to be false. Now, we're assuming the truth of $\alpha \Rightarrow \beta$ and $\beta \Rightarrow \gamma$. If γ is false, β has to be true. However, if so, $\alpha \Rightarrow \beta$ is going to fail. This contradicts the assumption. Therefore, the statement holds.

Solutions for Problems 3.2.1

1. These are all easy. Try to derive a contradiction from the assumption that the implication doesn't hold.

2.

1. Correct.
2. Incorrect.
3. Incorrect.
4. Correct.
5. Incorrect.

Solutions for Problems 4.1

1.

1. Suppose that the argument is not valid; thus, $(\alpha \vee \beta)$ and $-\alpha$ are true, but β is false. Since $(\alpha \vee \beta)$ is true and β is false, α has to be true. However, if so, $-\alpha$ is going to be false. Contradiction. The argument is valid.

2. Suppose that the argument is not valid; thus, $(\alpha \rightarrow \beta)$ and $-\beta$ are true, but $-\alpha$ is false. Since $\alpha \rightarrow \beta$ is true and $-\alpha$ is false, β has to be true. However, if so, β is going to be false. Contradiction. The argument is valid.

3. Suppose that the argument is not valid; thus, $(\aleph \rightarrow \gamma), (\beta \rightarrow \gamma), (\alpha \vee \beta)$ are all true, but γ is false. Since $(\alpha \rightarrow \gamma), (\beta \rightarrow \beta)$ are both true and γ is false, both α and β have to be false. However, if so, $(\alpha \vee \beta)$ is going to be false. Contradiction. The argument is valid.

2.

1. Invalid. (Counterexample: A=T, B=T, C=T/F)
2. Valid.
3. Invalid. (Counterexample: B=T, C=F, D=F)
4. Valid.
5. Invalid. (Counterexample: A=F, B=F, C=F)

Solutions for Problems 6.2.1

1. $((A \wedge -B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge C))$
2. $((A \wedge B \wedge C) \vee (A \wedge -B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge C))$
3. $((A \wedge B \wedge C) \vee (A \wedge -B \wedge C) \vee (A \wedge -B \wedge -C))$
4. $((A \wedge B \wedge -C) \vee (A \wedge -B \wedge C) \vee (A \wedge -B \wedge -C) \vee (-A \wedge B \wedge C) \vee (-A \wedge B \wedge -C) \vee (-A \wedge -B \wedge C) \vee (-A \wedge -B \wedge -C))$
5. $((A \wedge B \wedge -C) \vee (-A \wedge B \wedge -C) \vee (-A \wedge -B \wedge C))$

Solutions for Problems 7.1

1.

1. $(\alpha \vee \beta) : ((\alpha|\alpha)|(\beta|\beta))$
2. $(\alpha \rightarrow \beta) : (\alpha|(\beta|\beta))$
3. $(\alpha \downarrow \beta) : (((\alpha|\alpha)|(\beta|\beta))|((\alpha|\alpha)|(\beta|\beta)))$

2.

1. $-\alpha : (\alpha \downarrow \alpha)$
2. $(\alpha \wedge \beta) : ((\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta))$
3. $(\alpha \vee \beta) : ((\alpha \downarrow \beta) \downarrow (\alpha \downarrow \beta))$
4. $(\alpha \rightarrow \beta) : (((\alpha \downarrow \alpha) \downarrow \beta) \downarrow ((\alpha \downarrow \alpha) \downarrow \beta))$
5. $((\alpha|\beta) : (((\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta)) \downarrow ((\alpha \downarrow \alpha) \downarrow (\beta \downarrow \beta)))$

3.

1. $(\alpha \vee \beta) : -(-\alpha \wedge -\beta)$
2. $(\alpha \rightarrow \beta) : -(\alpha \wedge -\beta)$
3. $(\alpha|\beta) : -(\alpha \wedge \beta)$
4. $(\alpha \downarrow \beta) : (-\alpha \wedge \beta)$

4.

1. $(\alpha \wedge \beta) : -(-\alpha \vee -\beta)$
2. $(\alpha \rightarrow \beta) : (-\alpha \vee \beta)$
3. $(\alpha|\beta) : (-\alpha \vee -\beta)$
4. $(\alpha \downarrow \beta) : -(\alpha \vee \beta)$

5.

1. $(\alpha \wedge \beta) : -(\alpha \rightarrow -\beta)$

2. $(\alpha \vee \beta) : (-\alpha \to \beta)$
3. $(\alpha|\beta) : (\alpha \to -\beta)$
4. $(\alpha \downarrow \beta) : -(-\alpha \to \beta)$

Solutions for Problems 8.3.1

1.

$$\begin{array}{c} \checkmark\ (J \to (K \to J)) \\ \diagup \quad \diagdown \\ -J \qquad (K \to J) \end{array}$$

This tree is clearly open (you don' have to decompose the right-side path because the left-side path is open no matter what happens to the right-side path. See Rule of Thumb 3); thus, the sentence is satisfiable. However, we still need to decide whether it is valid or not.

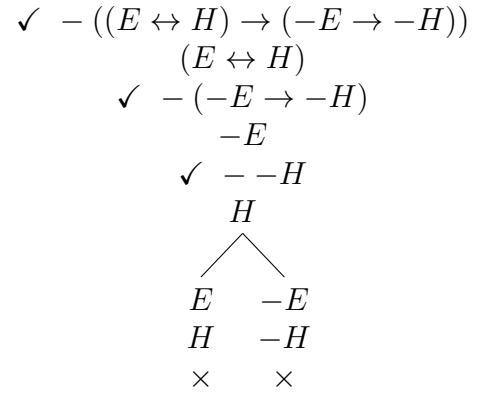
$$\begin{array}{c} \checkmark\ -(J \to (K \to J)) \\ J \\ \checkmark\ -(K \to J) \\ K \\ -J \\ \times \end{array}$$

This one is closed; thus, the sentence is valid.

2.

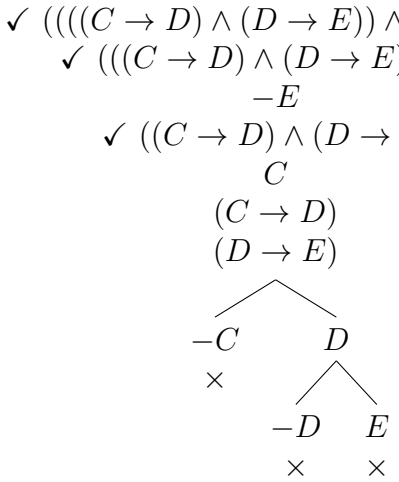
$$\begin{array}{c} \checkmark\ ((E \leftrightarrow H) \to (-E \to -H)) \\ \diagup \qquad\qquad \diagdown \\ \checkmark\ -(E \leftrightarrow H) \qquad \checkmark\ (-E \to -H) \\ \diagup \quad \diagdown \qquad\qquad \diagup \quad \diagdown \\ E \quad -E \qquad\qquad --E \quad -H \\ -H \quad H \end{array}$$

The tree is open; now we gonna test its validity.



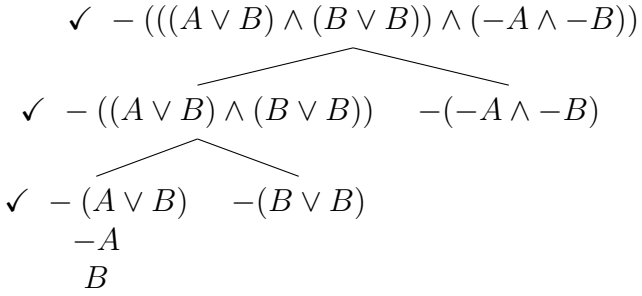
The tree is closed; the sentence is valid.

3.



The tree is closed; the sentence is unsatisfiable.

4.



The left-most path remains open no matter what happens to other paths; now for the test for its validity.

$$
\begin{array}{c}
--(((A \vee B) \wedge (B \vee B)) \wedge (-A \wedge -B)) \\
\checkmark\ (((A \vee B) \wedge (B \vee B)) \wedge (-A \wedge -B)) \\
\checkmark\ ((A \vee B) \wedge (B \vee B)) \\
\checkmark\ (-A \wedge -B) \\
\checkmark\ (A \vee B) \\
(B \vee B) \\
-A \\
-B \\
\diagup\ \diagdown \\
A \quad B \\
\times \quad \times
\end{array}
$$

The tree is closed; the sentence is valid.

5.

$$
\begin{array}{c}
\checkmark\ (-B \to ((B \vee D) \to D)) \\
\diagup\ \diagdown \\
-B \quad ((B \vee D) \to D)
\end{array}
$$

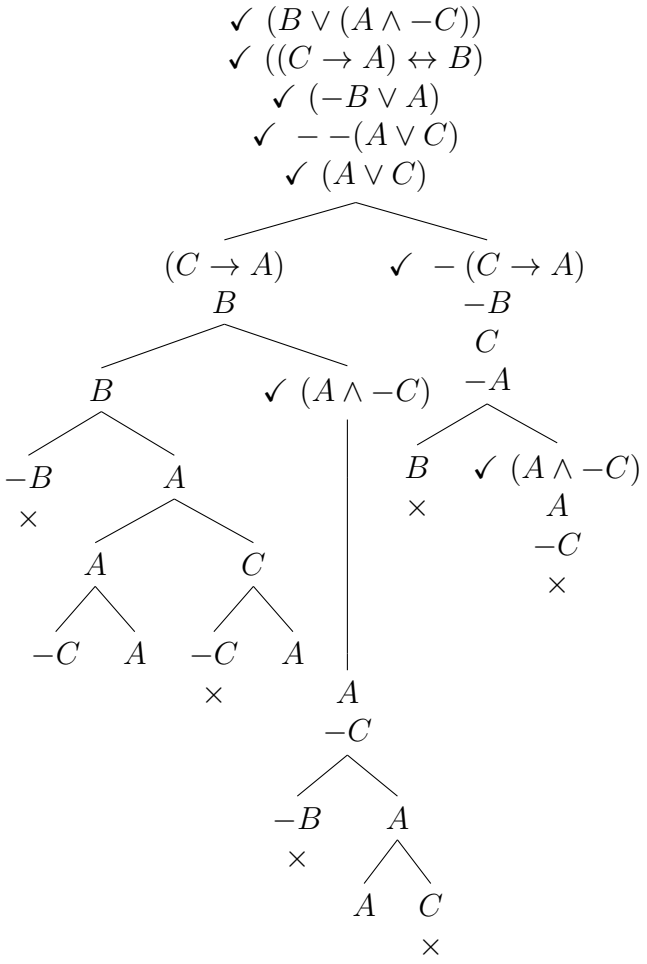
The tree is open (there's nothing to do with the left-side path); we're gonna test its validity.

$$
\begin{array}{c}
\checkmark\ -(-B \to ((B \vee D) \to D)) \\
-B \\
\checkmark\ -((B \vee D) \to D) \\
\checkmark\ (B \vee D) \\
-D \\
\diagup\ \diagdown \\
B \quad D \\
\times \quad \times
\end{array}
$$

The tree is closed; the sentence is valid.

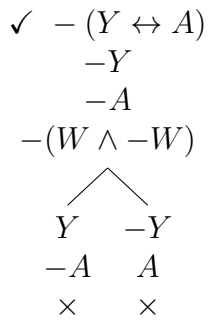
Solutions for Problems 8.4.1

1.



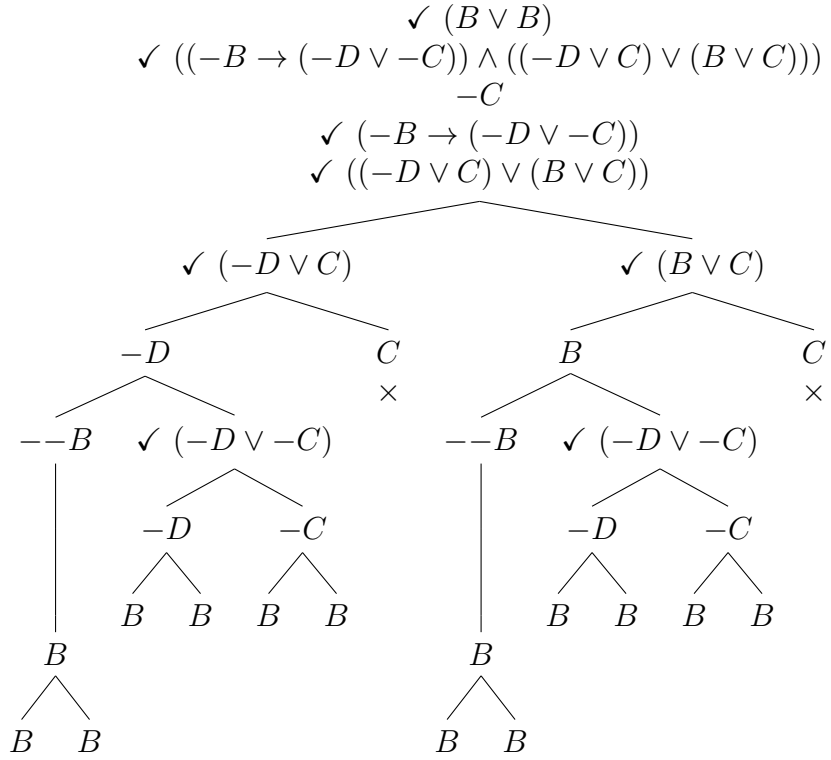
The tree is open; the argument is invalid. A counterexamples are given when A and B are both true, and C is either true or false.

2.



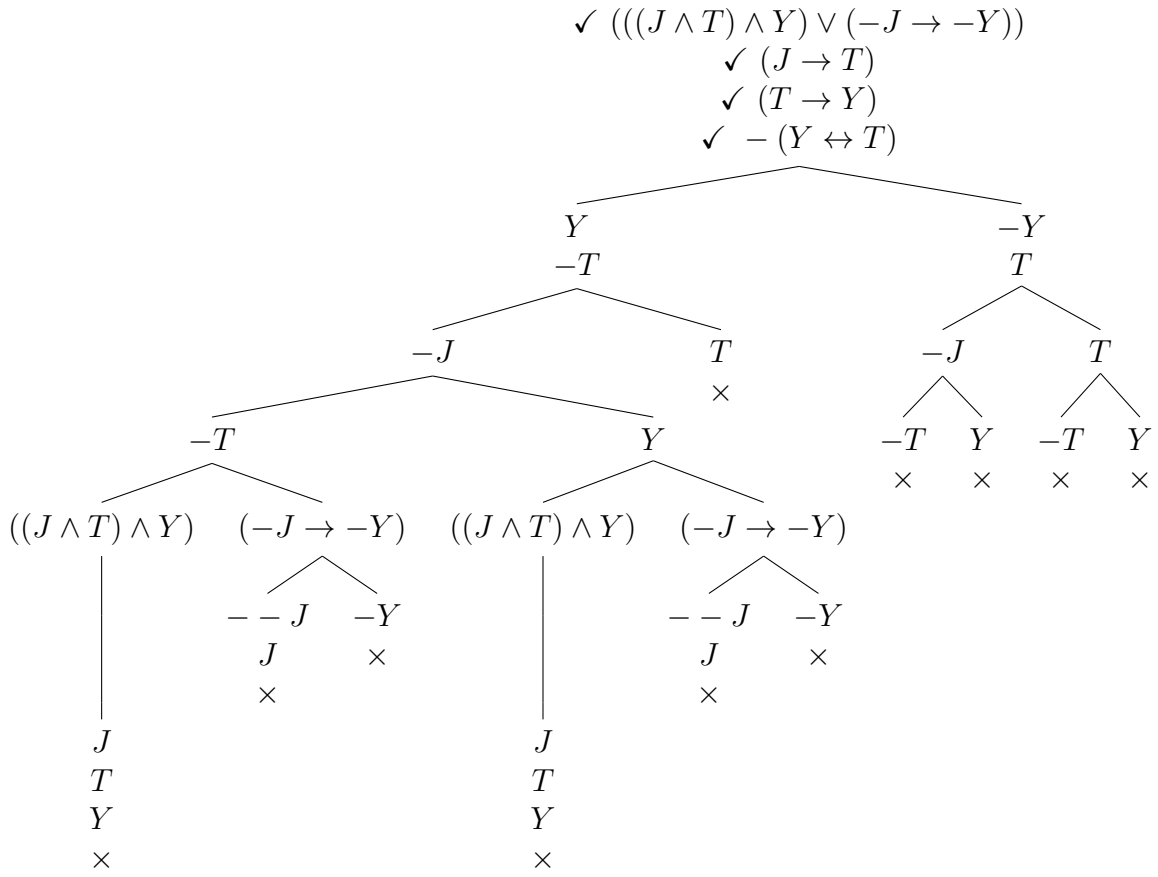
The tree is closed; the argument is valid.

3.



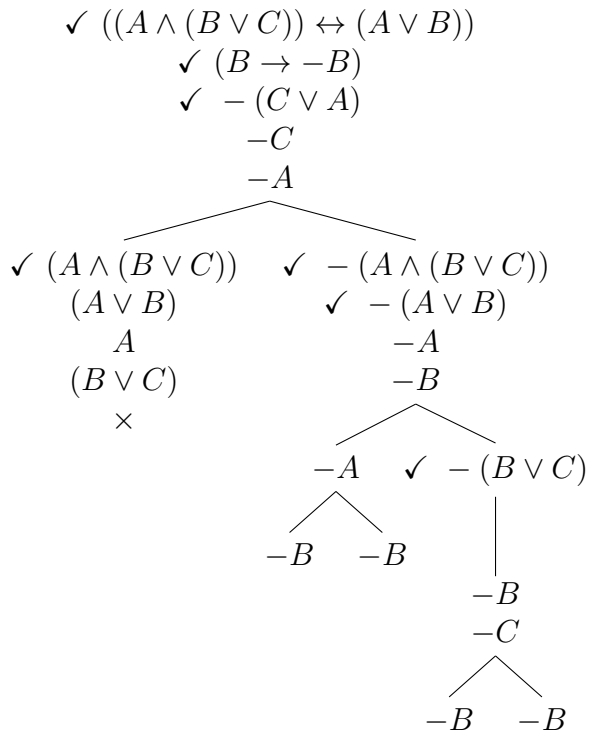
The tree is open; the argument is invalid. A counterexample is given when B is true and C and D are both false.

4.



The tree is closed; the argument is valid.

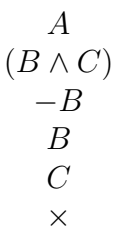
5.



The tree is open; the argument is invalid. A counterexample is given when A, B, C are all false.

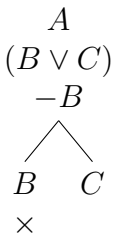
Solutions for Problems 8.5.1

1.



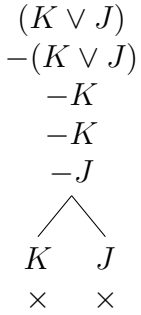
The tree is closed; the implication holds.

2.



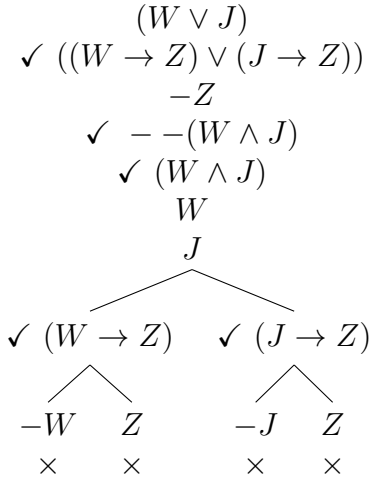
The tree is open; the implication fails. The counterexample is given when A, C are true and B is false.

3.



The tree is closed; the implication holds.

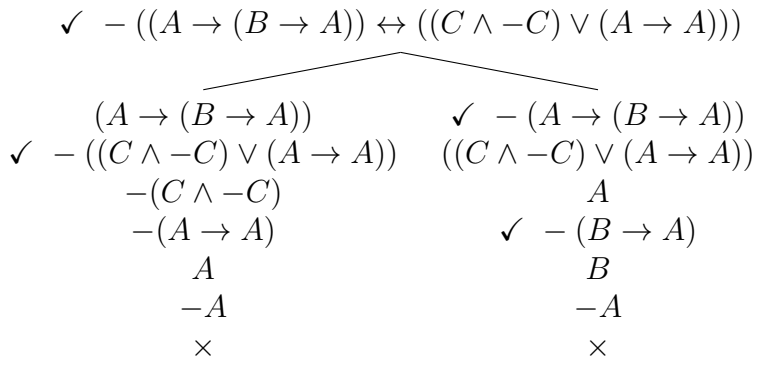
4.



The tree is closed; the implication holds.

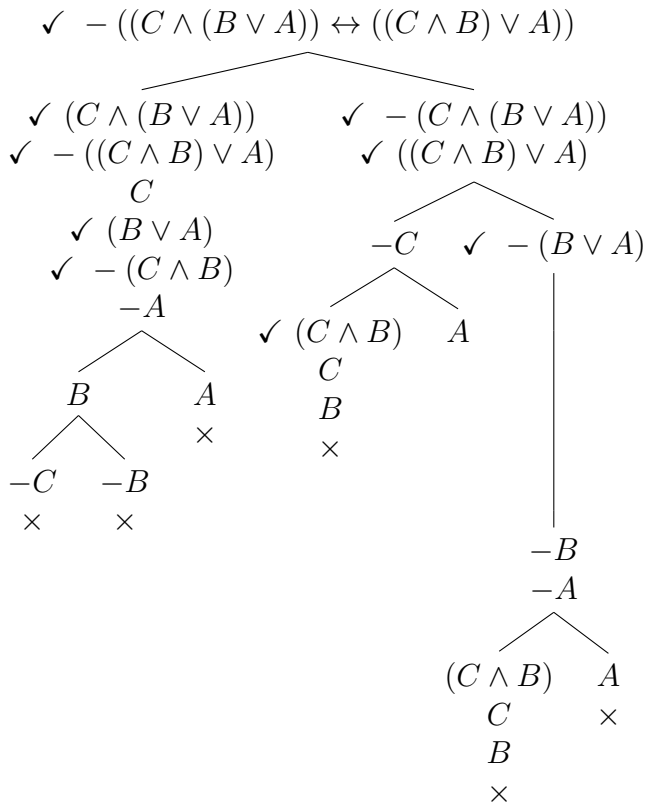
Solutions for Problems 8.6.1

1.



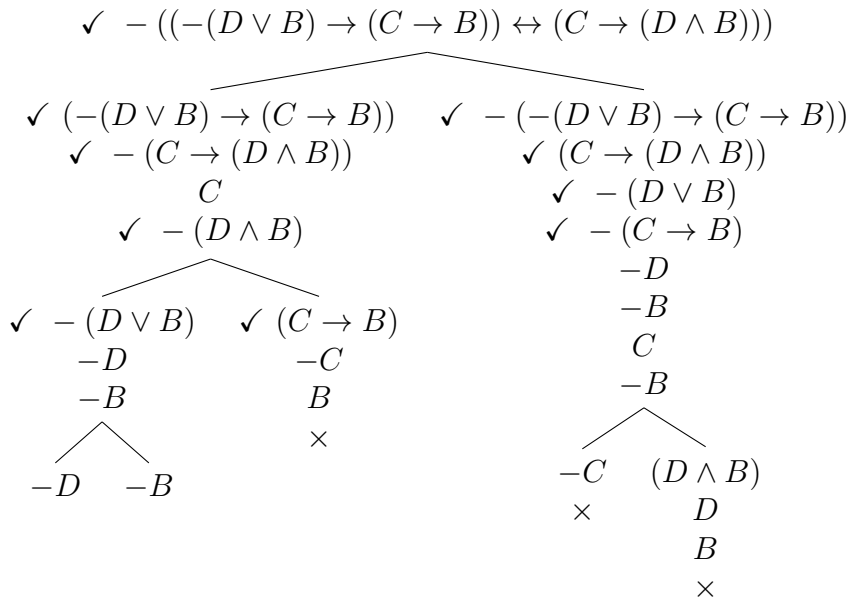
The tree is closed; the equivalence holds.

2.



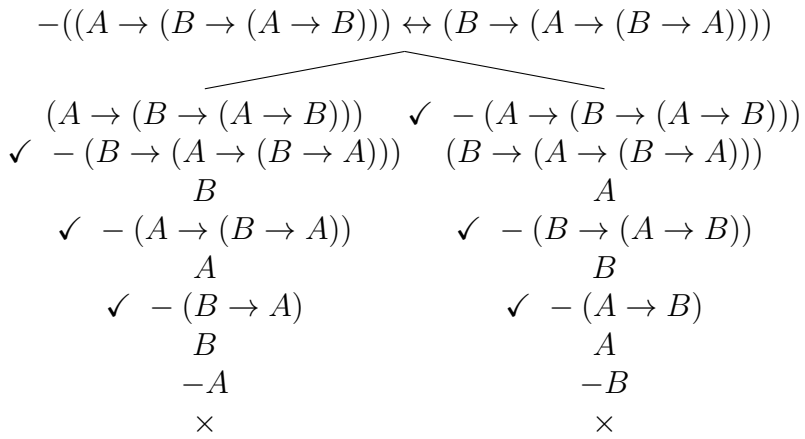
The tree is open; the equivalence fails.

3.



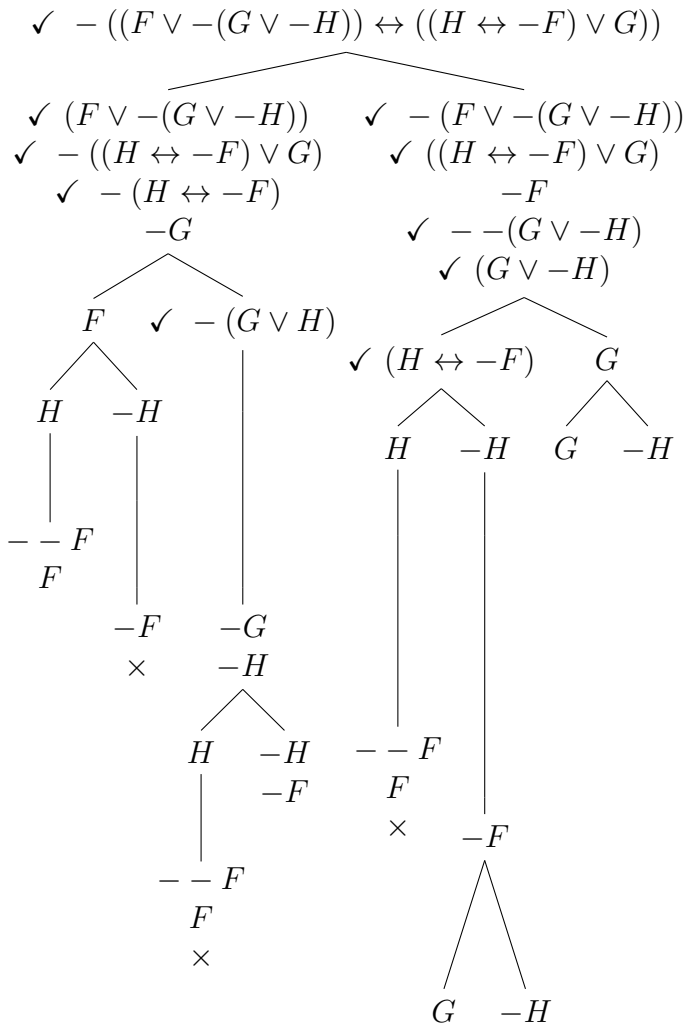
The tree is open; the equivalence fails.

4.



The tree is closed; the equivalence holds.

5.



The tree is open; the equivalence fails.