Definition 2.1. The *basic symbols* of SL are of three kinds:

1. Logical Connectives: $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
2. Punctuation Symbols: (,)
3. Sentence Letters: $A, B, C, \ldots, S, T, A_1, B_1, C_1, \ldots, S_1, T_1, A_2, \ldots$

Definition 2.1. The *basic symbols* of SL are of three kinds:

1. Logical Connectives: $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
2. Punctuation Symbols: (,)
3. Sentence Letters: $A, B, C, \ldots, S, T, A_1, B_1, C_1, \ldots, S_1, T_1, A_2, \ldots$

Definition 2.2. The *sentences of SL* are given by the following recursive definition:

Base Clause: Every sentence letter is a sentence.

Generating Clauses:

1. If ϕ is a sentence, then so is $\sim\phi$.
2. If ϕ and θ are sentences, then so are both $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
3. If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are sentences (the list must include at least two sentences and be finite), then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.

Closure Clause: A sequence of symbols is an SL sentence iff its being a sentence follows from the previous two clauses.

Definition 2.1. The *basic symbols* of SL are of three kinds:

1. Logical Connectives: $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
2. Punctuation Symbols: (,)
3. Sentence Letters: $A, B, C, \dots, S, T, A_1, B_1, C_1, \dots, S_1, T_1, A_2, \dots$

Definition 2.2. The *sentences of SL* are given by the following recursive definition:

Base Clause: Every sentence letter is a sentence.

Generating Clauses:

1. If ϕ is a sentence, then so is $\sim\phi$.
2. If ϕ and θ are sentences, then so are both $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
3. If all of $\phi_1, \phi_2, \phi_3, \phi_4, \dots, \phi_n$ are sentences (the list must include at least two sentences and be finite), then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \dots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \dots \vee \phi_n)$.

Closure Clause: A sequence of symbols is an SL sentence iff its being a sentence follows from the previous two clauses.

Definition 2.2 defines the **official sentences** of SL.

Definition 2.4 A string of symbols is an **unofficial sentence** of SL iff we can obtain it from an official sentence by

1. deleting outer parentheses, or
2. replacing one or more pairs of official round parentheses ( ) with square brackets [ ] or curly brackets { }.

Definition 2.5. The following clauses define when one sentence is a **subsentence** of another:

1. Every sentence is a subsentence of itself.
2. ϕ is a subsentence of $\sim\phi$.
3. ϕ and θ are subsentences of $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
4. All of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are subsentences of $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.
5. (Transitivity) If ϕ is a subsentence of θ and θ is a subsentence of ψ, then ϕ is a subsentence of ψ.
6. That's all.

A sentence ϕ is a **proper subsentence** of ψ iff ϕ is a subsentence of, but isn't identical to ψ. So, while each sentence is a subsentence of itself, no sentence is a proper subsentence of itself.

Definition 2.8 The following clauses define the **order** of every SL sentence. Let ORDϕ be the order of ϕ. Then:

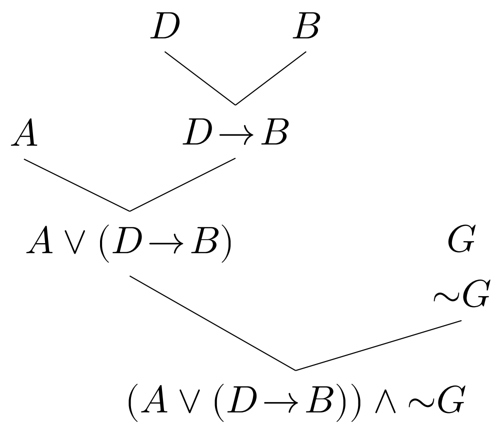
1. If ϕ is an atomic sentence (a sentence letter), then ORD$\phi = 1$.
2. For any sentence ϕ, ORD$\sim\phi$ = ORD$\phi + 1$.
3. For any sentences ϕ and θ, ORD$(\phi \rightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ. Likewise, ORD$(\phi \leftrightarrow \theta)$ is one greater than the max of ORDϕ and ORDθ.
4. For any sentences $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \wedge \ldots \wedge \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.
5. For any sentences $\phi_1, \ldots, \phi_n$, ORD$(\phi_1 \vee \ldots \vee \phi_n)$ is one greater than the max of ORDϕ_1, ..., ORDϕ_n.
6. That's all.

Definition 2.10. The **main connective** is the connective token (or tokens) that occur(s) in the sentence but in no proper subsentence.

Definition 2.12 The **construction tree** for a sentence is a diagram of how the sentence is generated through the recursive clauses of the definition of SL sentences. We put atomic sentences as leaves at the top, and the generating clauses specify how we can join nodes of the tree together (starting with the leaves at the top) into new nodes. The complete sentence is the node at the base of the tree.

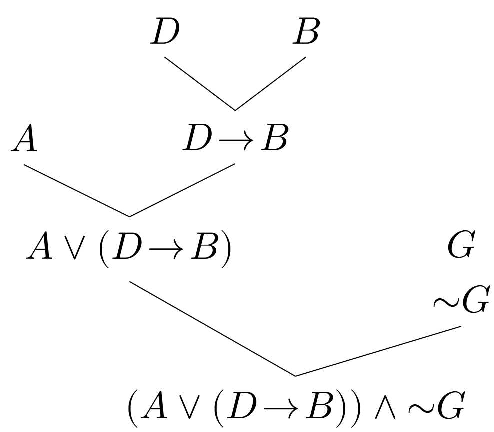
Definition 2.12 The **construction tree** for a sentence is a diagram of how the sentence is generated through the recursive clauses of the definition of SL sentences. We put atomic sentences as leaves at the top, and the generating clauses specify how we can join nodes of the tree together (starting with the leaves at the top) into new nodes. The complete sentence is the node at the base of the tree.

Example: $(A \vee (D \rightarrow B)) \wedge {\sim}G$.



Definition 2.12 The **construction tree** for a sentence is a diagram of how the sentence is generated through the recursive clauses of the definition of SL sentences. We put atomic sentences as leaves at the top, and the generating clauses specify how we can join nodes of the tree together (starting with the leaves at the top) into new nodes. The complete sentence is the node at the base of the tree.

Example: $(A \vee (D \rightarrow B)) \wedge {\sim}G$.

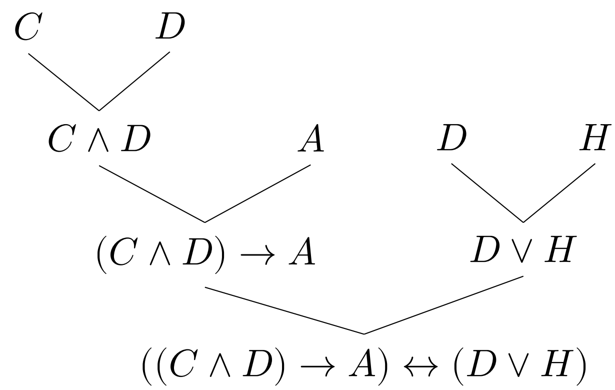


NB:

- The subsentences of a sentence are the nodes in the sentence's construction tree.
- The order of a sentence is the number of nodes of its longest branch.
- The main connective of a sentence is the connective added last (at the bottom) of the construction tree.

Definition 2.12 The **construction tree** for a sentence is a diagram of how the sentence is generated through the recursive clauses of the definition of SL sentences. We put atomic sentences as leaves at the top, and the generating clauses specify how we can join nodes of the tree together (starting with the leaves at the top) into new nodes. The complete sentence is the node at the base of the tree.

Example: $((C \wedge D) \to A) \leftrightarrow (D \vee H)$.



NB:

- The subsentences of a sentence are the nodes in the sentence's construction tree.
- The order of a sentence is the number of nodes of its longest branch.
- The main connective of a sentence is the connective added last (at the bottom) of the construction tree. f

Truth in a Model

Model of single sentence:

Definition 2.17 Given that ϕ is a sentence of SL, a *model for* ϕ is an assignment of a truth value, either true or false, to each sentence letter in ϕ.

notation:

if a model $\mathfrak{m}$ assigns a sentence letter ψ the value true, we will write $\mathfrak{m}(\psi) = \mathsf{T}$

If $\mathfrak{m}$ assigns a sentence letter ψ the value false, we will write $\mathfrak{m}(\psi) = \mathsf{F}$

Truth in a Model

Model of single sentence:

Definition 2.17 Given that ϕ is a sentence of SL, a *model for* ϕ is an assignment of a truth value, either true or false, to each sentence letter in ϕ.

notation:

- if a model $\mathfrak{m}$ assigns a sentence letter ψ the value true, we will write $\mathfrak{m}(\psi) = \mathsf{T}$
- If $\mathfrak{m}$ assigns a sentence letter ψ the value false, we will write $\mathfrak{m}(\psi) = \mathsf{F}$.

Model of a set of sentences:

Definition 2.18 Given that Δ is a set of SL sentences, $\mathfrak{m}$ is a *model for* Δ iff $\mathfrak{m}$ is a model for each sentence in Δ [i.e. iff $\mathfrak{m}$ is a model for each sentence letter in each sentence in Δ (by Def 2.17)]

Truth in a Model

Model of single sentence:

Definition 2.17 Given that ϕ is a sentence of SL, a *model for* ϕ is an assignment of a truth value, either true or false, to each sentence letter in ϕ.

notation:

- if a model $\mathfrak{m}$ assigns a sentence letter ψ the value true, we will write $\mathfrak{m}(\psi) = \mathsf{T}$
- If $\mathfrak{m}$ assigns a sentence letter ψ the value false, we will write $\mathfrak{m}(\psi) = \mathsf{F}$.

Model of a set of sentences:

Definition 2.18 Given that Δ is a set of SL sentences, $\mathfrak{m}$ is a *model for* Δ iff $\mathfrak{m}$ is a model for each sentence in Δ [i.e. iff $\mathfrak{m}$ is a model for each sentence letter in each sentence in Δ (by Def 2.17)]

Model for SL

Definition 2.19 $\mathfrak{m}$ is a *model for SL* iff $\mathfrak{m}$ is a model for every sentence of SL.

Sentences more complex than single sentence letters are not directly assigned truth values by models:

Model: $\mathfrak{m}(A) = \mathsf{T}, \mathfrak{m}(B) = \mathsf{F}, \mathfrak{m}(D) = \mathsf{F}, \mathfrak{m}(G) = \mathsf{T}$

sentence: $(A \vee (D \rightarrow B)) \wedge {\sim}G$

$\mathfrak{m}[(A \vee (D \rightarrow B)) \wedge {\sim}G] = ???$

Truth Functions

Definition 2.20 The following clauses define when an SL sentence θ is *true* (or *false*) on a model $\mathfrak{m}$ for θ:

1. A sentence letter ϕ is true on $\mathfrak{m}$ iff $\mathfrak{m}$ assigns true to it, i.e. iff $\mathfrak{m}(\phi) = \mathsf{T}$.
2. A negation $\sim\phi$ is true on $\mathfrak{m}$ iff the sentence ϕ is false on $\mathfrak{m}$.
3. A conjunction $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n$ is true on $\mathfrak{m}$ iff all of the conjuncts are true on $\mathfrak{m}$.
4. A disjunction $\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n$ is true on $\mathfrak{m}$ iff at least one of the disjuncts is true on $\mathfrak{m}$.
5. A conditional $\phi \rightarrow \psi$ is true on $\mathfrak{m}$ iff the LHS is false or the RHS is true on $\mathfrak{m}$.
6. A biconditional $\phi \leftrightarrow \psi$ is true on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

Truth Functions

Definition 2.20 The following clauses define when an SL sentence θ is *true* (or *false*) on a model $\mathfrak{m}$ for θ:

1. A sentence letter ϕ is true on $\mathfrak{m}$ iff $\mathfrak{m}$ assigns true to it, i.e. iff $\mathfrak{m}(\phi) = \mathsf{T}$.
2. A negation $\sim\phi$ is true on $\mathfrak{m}$ iff the sentence ϕ is false on $\mathfrak{m}$.
3. A conjunction $\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n$ is true on $\mathfrak{m}$ iff all of the conjuncts are true on $\mathfrak{m}$.
4. A disjunction $\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n$ is true on $\mathfrak{m}$ iff at least one of the disjuncts is true on $\mathfrak{m}$.
5. A conditional $\phi \rightarrow \psi$ is true on $\mathfrak{m}$ iff the LHS is false or the RHS is true on $\mathfrak{m}$.
6. A biconditional $\phi \leftrightarrow \psi$ is true on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

Important: truth value assignments vary from model to model, truth-funcitonal definitions of connectives do not.

Computing truth values of complex sentences

Method 1: compute $\mathfrak{m}(\phi)$ by computing truth in $\mathfrak{m}$ for each of ϕ's subsentences, using Definition 2.20

Model: $\mathfrak{m}(A) = \mathsf{T}, \mathfrak{m}(B) = \mathsf{F}, \mathfrak{m}(D) = \mathsf{F}, \mathfrak{m}(G) = \mathsf{T}$

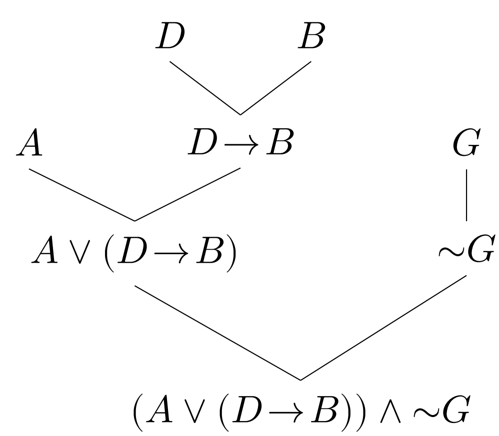
1. $\mathfrak{m}(A \wedge {\sim}B) =$
2. $\mathfrak{m}((A \vee (D \rightarrow B)) \wedge {\sim}G)) =$

Computing truth values of complex sentences

Method 2: construction tree method

1. **construct the tree** ϕ
2. $\mathfrak{m}$ and Clause 1 of Def. 2.20 establish the truth values of the sentence letters at the top of the tree
3. work you way down, using the relevant clause from Def 2.20 to determine the truth value of more complex subsentences of ϕ

Example: $(A \vee (D \rightarrow B)) \wedge {\sim}G$.

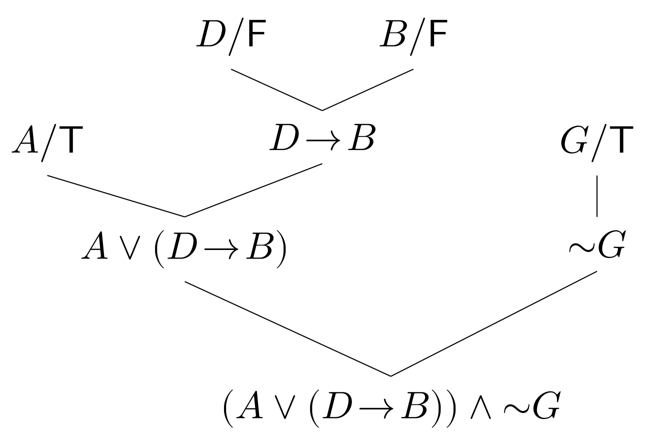


Computing truth values of complex sentences

Method 2: construction tree method

1. **construct the tree ϕ**
2. **$\mathfrak{m}$ and Clause 1 of Def. 2.20 establish the truth values of the sentence letters at the top of the tree**
3. work you way down, using the relevant clause from Def 2.20 to determine the truth value of more complex subsentences of ϕ

Example: $(A \vee (D \rightarrow B)) \wedge {\sim}G$.

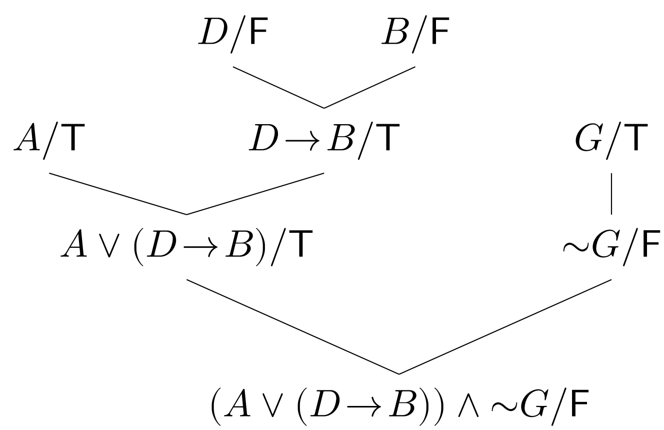


Computing truth values of complex sentences

Method 2: construction tree method

1. **construct the tree** ϕ
2. $\mathfrak{m}$ **and Clause 1 of Def. 2.20 establish the truth values of the sentence letters at the top of the tree**
3. **work you way down, using the relevant clause from Def 2.20 to determine the truth value of more complex sub-sentences of** ϕ

Example: $(A \vee (D \rightarrow B)) \wedge {\sim}G$.



Practice sentences

Calculate the truth values of $\mathfrak{m}$ using Definition 2.20, justifying your answers. Then, verify your answers by constructing a truth tree for each sentence

Model: $\mathfrak{m}(A) = \mathsf{T}, \mathfrak{m}(B) = \mathsf{F}, \mathfrak{m}(D) = \mathsf{F}, \mathfrak{m}(G) = \mathsf{T}$

1. $\mathfrak{m}(A \wedge {\sim}A) =$
2. $\mathfrak{m}({\sim}A \rightarrow B) =$
3. $\mathfrak{m}(A \rightarrow (A \leftrightarrow {\sim}B) =$
4. $\mathfrak{m}(B \vee (A \rightarrow B)) =$

Truth Tables and Truth Functional Connectives

$\sim$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are *truth functions:* functions from truth values to truth values, often given in *truth tables*.

Example:

ϕ	ψ	$\phi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

Recall:

Definition 2.20(5) The following clauses define when an SL sentence θ is *true* (or *false*) on a model $\mathfrak{m}$ for θ:

5. A conditional $\phi \rightarrow \psi$ is true on $\mathfrak{m}$ iff the LHS is false or the RHS is true on $\mathfrak{m}$.

Truth Tables and Truth Functional Connectives

$\sim$, $\wedge$, $\vee$, $\rightarrow$ and $\leftrightarrow$ are *truth functions:* functions from truth values to truth values, often given in *truth tables*.

Example:

ϕ	ψ	$\phi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

Recall:

Definition 2.20(5) The following clauses define when an SL sentence θ is *true* (or *false*) on a model $\mathfrak{m}$ for θ:

5. A conditional $\phi \rightarrow \psi$ is true on $\mathfrak{m}$ iff the LHS is false or the RHS is true on $\mathfrak{m}$.

What are the truth tables for the other truth functional connectives?

Logical Truth

Distinguish three types of sentences of SL:

Definition 2.26: A sentence ϕ of SL is *truth functionally true* (TFT) iff it is true on all models for ϕ.

- Examples of TFT's: $A \vee \sim A$, $A \leftrightarrow A$

Definition 2.27: A sentence ϕ of SL is *truth functionally false* (TFF) iff it is false on all models for ϕ.

- Examples of TFF's: $A \wedge \sim A, A \leftrightarrow \sim A$

Definition 2.28: A sentence ϕ of SL is *truth functionally contingent* (TFC) iff it is true on some models for ϕ and false on others.

- Examples of TFC's: A, $A \vee B$

TFT, TFF, or TFC and truth tables

example: $(A \vee B) \rightarrow A$

A	B	$(A \vee B) \rightarrow A$
T	T	
T	F	
F	T	
F	F	

TFT, TFF, or TFC and truth tables

example: $(A \vee B) \to A$

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

TFT, TFF, or TFC and truth tables

example: $(A \wedge B) \to A$

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

TFT, TFF, or TFC and truth tables

example: $(A \wedge B) \to A$

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

TFT, TFF, or TFC and truth tables

example: $((A \vee (D \rightarrow B)) \wedge {\sim}G)$

A	B	D	G	$((A \vee (D \rightarrow B)) \wedge {\sim}G)$
T	T	T	T	F
T	T	T	F	T
T	T	F	T	F
T	T	F	F	T
T	F	T	T	F
T	F	T	F	T
T	F	F	T	F
T	F	F	F	T
F	T	T	T	F
F	T	T	F	T
F	T	F	T	F
F	T	F	F	T
F	F	T	T	F
F	F	T	F	F
F	F	F	T	F
F	F	F	F	T

Comments on truth tables

1. they get big, fast
   - with 2 truth values and n sentence letters, there are 2^n lines on a truth table
2. though helpful in studying simple languages such as SL, they're less helpful in studying more sophisticated languages

TFT, TFF, TFC and Informal Proofs

Example: Prove whether $(A \wedge B) \rightarrow A$ is TFT, TFF, or TFC

TFT, TFF, TFC and Informal Proofs

Example: Prove whether $(A \wedge B) \rightarrow A$ is TFT, TFF, or TFC

Proof 1:

1. Let $\mathfrak{m}$ be an arbitrary model of SL at which A is true
2. So, $\mathfrak{m}(A \wedge B) \rightarrow A = \text{T}$ (from 1, Def. 2.20(5))
3. $\mathfrak{m}$ was arbitrarily selected, so the point generalizes: $(A \wedge B) \rightarrow A$ is true at every SL model at which A is true. (from 1,2)
4. Let $\mathfrak{m}'$ be an arbitrary model of SL at which A is false
5. So, $\mathfrak{m}'(A \wedge B) = \text{F}$ (from 4, Def. 2.20(3))
6. So, $\mathfrak{m}'(A \wedge B) \rightarrow A = \text{T}$ (from 5, Def. 2.20(5))
7. $\mathfrak{m}'$ was arbitrarily selected, so the point generalizes: $(A \wedge B) \rightarrow A$ is true at every SL model at which A is false.
8. At every SL model: either A is true, or A is false (Def. 2.20(7))
9. So, $(A \wedge B) \rightarrow A$ is true at every SL model (from 3,7.8)
10. So, $(A \wedge B) \rightarrow A$ is TFT (from 9, Def. 2.26)

For reference: Def. 2.20 [shorter]:

1. $\mathfrak{m}(\phi) = \text{T}$ iff $\mathfrak{m}$ assigns true to it.
2. $\mathfrak{m}(\sim\phi) = \text{T}$ iff $\mathfrak{m}(\phi) = \text{F}$.
3. $\mathfrak{m}(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n) = \text{T}$ iff all conjuncts are T on $\mathfrak{m}$.
4. $\mathfrak{m}(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n) = \text{T}$ iff at least one disjunct is T on $\mathfrak{m}$.
5. $\mathfrak{m}(\phi \rightarrow \psi) = \text{T}$ iff the LHS is F or the RHS is T on $\mathfrak{m}$.
6. $\mathfrak{m})(\phi \leftrightarrow \psi)$ is T on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

TFT, TFF, TFC and Informal Proofs

Example: Prove whether $(A \wedge B) \rightarrow A$ is TFT, TFF, or TFC

TFT, TFF, TFC and Informal Proofs

Example: Prove whether $(A \wedge B) \rightarrow A$ is TFT, TFF, or TFC

Proof 2:

1. Suppose for *reductio* that there exists a model $\mathfrak{m}$ at which $\mathfrak{m}((A \wedge B) \rightarrow A) = \text{F}$
2. So, $\mathfrak{m}((A \wedge B) = \text{T}$ and $\mathfrak{m}(A) = \text{F}$ (from 1, Def. 2.20(5))
3. So, $\mathfrak{m}(A) = \text{T}$ (from 2, Def. 2.20(3))
4. So, $\mathfrak{m}(A) = \text{T}$ and $\mathfrak{m}(A) = \text{F}$ (From 2,3)
5. From the supposition at (1) we've derived a contraction at (5), so the supposition at (1) must be false: there is no model at which $\mathfrak{m}((A \wedge B) \rightarrow A) = \text{F}$ (from 1, 4)
6. So, $(A \wedge B) \rightarrow A$ is TFT (from 5, Def. 2.26)

For reference: Def. 2.20 [shorter]:

1. $\mathfrak{m}(\phi) = \text{T}$ iff $\mathfrak{m}$ assigns true to it.
2. $\mathfrak{m}({\sim}\phi) = \text{T}$ iff $\mathfrak{m}(\phi) = \text{F}$.
3. $\mathfrak{m}(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n) = \text{T}$ iff all conjuncts are T on $\mathfrak{m}$.
4. $\mathfrak{m}(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n) = \text{T}$ iff at least one disjunct is T on $\mathfrak{m}$.
5. $\mathfrak{m}(\phi \rightarrow \psi) = \text{T}$ iff the LHS is F or the RHS is T on $\mathfrak{m}$.
6. $\mathfrak{m})(\phi \leftrightarrow \psi)$ is T on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

TFT, TFF, TFC and Informal Proofs

Prove whether $(A \lor B) \to (B \lor C)$ is TFT, TFF, or TFC

TFT, TFF, TFC and Informal Proofs

Prove whether $(A \vee B) \rightarrow (B \vee C)$ is TFT, TFF, or TFC

Proof:

1. Let $\mathfrak{m}$ be a model that makes C true
2. So, $\mathfrak{m}(B \vee C) = \text{T}$ (From 1, Def. 2.20.(4))
3. So, $\mathfrak{m}((A \vee B) \rightarrow (B \vee C)) = \text{T}$ (from 2, Def. 2.20(5))
4. So, there's at least one model that makes $(A \vee B) \rightarrow (B \vee C)$ true (from 3)
5. Let $\mathfrak{m}'$ be a model that makes A true, and makes B and C false
6. So, $\mathfrak{m}'(A \vee B) = \text{T}$ (from 5, Def. 2.20(4))
7. So, $\mathfrak{m}'(B \vee C) = \text{F}$ (from 5, Def. 2.20(4))
8. So, $\mathfrak{m}'((A \vee B) \rightarrow (B \vee C)) = \text{F}$ (from 6,7, Def. 2.20(5))
9. So, there's at least one model that makes $(A \vee B) \rightarrow (B \vee C)$ false (from 9)
10. So, $(A \vee B) \rightarrow (B \vee C)$ is TFC (from 4,10, Def. 2.28)

For reference: Def. 2.20 [shorter]:

1. $\mathfrak{m}(\phi) = \text{T}$ iff $\mathfrak{m}$ assigns true to it.
2. $\mathfrak{m}(\sim\phi) = \text{T}$ iff $\mathfrak{m}(\phi) = \text{F}$.
3. $\mathfrak{m}(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n) = \text{T}$ iff all conjuncts are T on $\mathfrak{m}$.
4. $\mathfrak{m}(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n) = \text{T}$ iff at least one disjunct is T on $\mathfrak{m}$.
5. $\mathfrak{m}(\phi \rightarrow \psi) = \text{T}$ iff the LHS is F or the RHS is T on $\mathfrak{m}$.
6. $\mathfrak{m})(\phi \leftrightarrow \psi)$ is T on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

TFT, TFF, TFC and Informal Proofs

Prove whether $A \leftrightarrow \sim A$ is TFT, TFF, or TFC

TFT, TFF, TFC and Informal Proofs

Prove whether $A \leftrightarrow \sim A$ is TFT, TFF, or TFC

Proof 1:

1. Let $\mathfrak{m}$ be an arbitrary model of SL; either $\mathfrak{m}(A) = \text{T}$, or $\mathfrak{m}(A) = \text{F}$ (Def. 2.20(7))
2. Suppose $\mathfrak{m}(A) = \text{T}$
3. So, $\mathfrak{m}(\sim A) = \text{F}$ (from 2, Def. 2.20(2))
4. So, $\mathfrak{m}(A \leftrightarrow \sim A) = \text{F}$ (from 2,3, Def. 2.20(6))
5. So, if $\mathfrak{m}(A) = \text{T}$, then $\mathfrak{m}(A \leftrightarrow \sim A) = \text{F}$ (from 2-4)
6. Now suppose that $\mathfrak{m}(A) = \text{F}$
7. So, $\mathfrak{m}(\sim A) = \text{T}$ (from 6, Def. 2.20(2))
8. So, $\mathfrak{m}(A \leftrightarrow \sim A) = \text{F}$ (from 8, Def. 2.20(6))
9. So, if $\mathfrak{m}(A) = \text{F}$, then $\mathfrak{m}(A \leftrightarrow \sim A) = \text{F}$ (from 6-8)
10. So, $\mathfrak{m}(A \leftrightarrow \sim A) = \text{F}$ (from 1, 2-5, 6-9)
11. $\mathfrak{m}$ was arbitrarily chosen, so the point generalizes: at every SL model, $A \leftrightarrow \sim A$ is false (from 1-10)
12. So, $A \leftrightarrow \sim A$ is TFF (from 11, Def. 2.27)

For reference: Def. 2.20 [shorter]:

1. $\mathfrak{m}(\phi) = \text{T}$ iff $\mathfrak{m}$ assigns true to it.
2. $\mathfrak{m}(\sim\phi) = \text{T}$ iff $\mathfrak{m}(\phi) = \text{F}$.
3. $\mathfrak{m}(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge ... \wedge \phi_n) = \text{T}$ iff all conjuncts are T on $\mathfrak{m}$.
4. $\mathfrak{m}(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee ... \vee \phi_n) = \text{T}$ iff at least one disjunct is T on $\mathfrak{m}$.
5. $\mathfrak{m}(\phi \rightarrow \psi) = \text{T}$ iff the LHS is F or the RHS is T on $\mathfrak{m}$.
6. $\mathfrak{m})(\phi \leftrightarrow \psi)$ is T on $\mathfrak{m}$ iff both ϕ and ψ have the same truth value on $\mathfrak{m}$.
7. If a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$.

TFT, TFF, TFC and Informal Proofs

Informal proofs are sometimes easier to construct that truth tables:

Prove whether:

$(A \vee B \vee C \vee D \vee E) \rightarrow [(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)]$

is TFT, TFF, or TFC

NB: The truth table would have $2^6 = 64$ lines!

TFT, TFF, TFC and Informal Proofs

Informal proofs are sometimes easier to construct that truth tables:

Prove whether:
$(A \vee B \vee C \vee D \vee E) \rightarrow [(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)]$
is TFT, TFF, or TFC

NB: The truth table would have $2^6 = 64$ lines!

Proof:

1. Let $\mathfrak{m}$ be an arbitrary model that makes $A \vee B \vee C \vee D \vee E$ true.
2. So, $\mathfrak{m}$ makes true at least one disjunct of $A \vee B \vee C \vee D \vee E$; let ϕ be one of those disjuncts true at $\mathfrak{m}$. (from 1, Def. 2.20(4)).
3. So, any disjunction with ϕ as a disjunct is true at $\mathfrak{m}$; let θ be one such disjunction (from 2, Def. 2.20(4)).
4. So, any disjunction with θ as a disjunct is true at $\mathfrak{m}$ (from 3, Def. 2.20(4)).
5. So, $(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)$ is true at $\mathfrak{m}$ (from 1-4).
6. So, $(A \vee B \vee C \vee D \vee E) \rightarrow [(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)]$ is true at $\mathfrak{m}$ (from 5, Def. 2.20(5)).
7. $\mathfrak{m}$ was arbitrarily chosen, so the point generalizes: $(A \vee B \vee C \vee D \vee E) \rightarrow [(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)]$ is true at every SL model (from 1-6)
8. So, $(A \vee B \vee C \vee D \vee E) \rightarrow [(A \vee F) \vee (B \vee F) \vee (C \vee F) \vee (D \vee F) \vee (E \vee F)]$ is TFT (from 7, Def. 2.26)

Entailment

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Testing entailment with truth tables

prove: $(A \wedge B) \models B$

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

Testing entailment with truth tables

prove: $(A \wedge B) \models B$

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

Testing entailment with truth tables

prove: $(A \wedge B) \models B$

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

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.

2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Testing entailment with truth tables

prove: $A \vee B, \sim C \rightarrow \sim A, B \rightarrow C \models C$

A	B	C	$A \vee B$	$\sim C \rightarrow \sim A$	$B \rightarrow C$	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				

Testing entailment with truth tables

prove: $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C \models C$

A	B	C	$A \vee B$	${\sim}C \rightarrow {\sim}A$	$B \rightarrow C$	C
T	T	T	T	T	T	T
T	T	F	T	F	F	F
T	F	T	T	T	T	T
T	F	F	T	F	T	F
F	T	T	T	T	T	T
F	T	F	T	T	F	F
F	F	T	F	T	T	T
F	F	F	F	T	T	F

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Testing entailment with informal proofs

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Proof 1: $(A \wedge B) \vDash B$

1. Let $\mathfrak{m}$ be an arbitrary model on which $A \wedge B$ is true
2. So, $\mathfrak{m}(B) = \mathrm{T}$ (from 1, Def. 2.20(3))
3. $\mathfrak{m}$ was arbitrarily chosen, so the point generalizes: every model that makes $A \wedge B$ true also makes B true (from 1,2)
4. So, $(A \wedge B) \vDash B$ (from 3, Def. 2.38)

Testing entailment with informal proofs

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Proof 2: $(A \land B) \vDash B$

1. Suppose for *reductio* that there exists a model $\mathfrak{m}$ that makes $A \land B$ true and B false
2. So, $\mathfrak{m}(A \land B) = \text{F}$ (from 1, Def. 2.20)
3. So, $\mathfrak{m}(A \land B) = \text{T}$ and $\mathfrak{m}(A \land B) = \text{F}$ (from 1,2)
4. But that's impossible: a sentence is false on $\mathfrak{m}$ iff it's not true on $\mathfrak{m}$ (Def. 2.20(7))
5. So, reject supposition 1: there is no model that makes $A \land B$ true and B false
6. So, $(A \land B) \vDash B$ (from 5, Def. 2.38)

Testing entailment with informal proofs

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Proof: $A, A \to B \vDash B$

1. Let $\mathfrak{m}$ be an arbitrary model that makes true both A and $A \to B$
2. $\mathfrak{m}(A \to B) = \text{T}$ iff either: $\mathfrak{m}(A) = \text{F}$, or $\mathfrak{m}(B) = \text{T}$. (Def. 2.20(5)).
3. $\mathfrak{m}(A) = \text{T}$ (from 1), and A can't be both true and false at $\mathfrak{m}$ (from Def. 2.20(7))
4. So, $\mathfrak{m}(B) = \text{T}$
5. $\mathfrak{m}$ was arbitrarily chosen, so the point generalizes: every model that makes both A and $A \to B$ true also makes B true (from 1-4)
6. So, $A, A \to B \vDash B$ (from 5, Def. 2.38)

Testing entailment with informal proofs

Definition 2.38. If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.

2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

prove: $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C \models C$

Proof:

1. Suppose for *reductio* that there exists a model $\mathfrak{m}$ that makes $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C$ all true, and C is false.

2. Since $\mathfrak{m}(B \rightarrow C) = \text{T}$ and $\mathfrak{m}(C) = \text{F}$, $\mathfrak{m}(B) = \text{F}$. (1, Def. 2.20)

3. Since $\mathfrak{m}(A \vee B) = \text{T}$ and $\mathfrak{m}(B) = \text{F}$, $\mathfrak{m}(A) = \text{T}$ (1,2, Def. 2.20)

4. Since $\mathfrak{m}(C) = \text{F}$, $\mathfrak{m}({\sim}C) = \text{T}$ (from 1, Def. 2.20)

5. Since $\mathfrak{m}({\sim}C) = \text{T}$ and $\mathfrak{m}({\sim}C \rightarrow {\sim}A) = \text{T}$, $\mathfrak{m}({\sim}A) = \text{T}$ (1,4, Def. 2.20)

6. So, $\mathfrak{m}(A) = \text{T}$ and $\mathfrak{m}({\sim}A) = \text{F}$ (from 3,5)

7. But no model makes a sentence both true and false, so that's impossible (from Def. 2.20)

8. So, reject the supposition at (1): there does not exist a model that makes

   $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C$ all true, and C is false. (1-7)

9. So, $A \vee B, {\sim}C \rightarrow {\sim}A, B \rightarrow C \vDash C$

Your turn

Prove, then check your answer with a truth table:

1. $A \models A$
2. $A \models A \vee B$
3. $\sim A \models A \rightarrow B$
4. $A \leftrightarrow B \models A \rightarrow B$
5. $A, A \rightarrow B \models B$
6. $\models A \vee \sim A$
7. $A \wedge \sim A \models B$

SL Exportation Theorem: For all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models \phi \rightarrow \theta$

SL Exportation Theorem: For all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models \phi \rightarrow \theta$

How to prove a biconditional (e.g. ϕ iff ψ):

- ϕ iff ψ is equivalent to: (if ϕ then ψ) and (if ψ then ϕ)
- we prove the biconditional by proving the two conditionals

SL Exportation Theorem: For all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models \phi \rightarrow \theta$

How to prove a biconditional (e.g. ϕ iff ψ):

- ϕ iff ψ is equivalent to: (if ϕ then ψ) and (if ψ then ϕ)
- we prove the biconditional by proving the two conditionals

How to prove a conditional (e.g. if ϕ then ψ):

1. assume that the LHS (ϕ) of the conditional is true
2. on the basis of that information, derive the RHS (ψ) of the conditional
3. conclude that 'if ϕ then ψ'

SL Exportation Theorem: For all SL sentences ϕ and θ, $\phi \models \theta$ iff $\models \phi \rightarrow \theta$

Proof:

First step: prove that if $\phi \models \theta$ then $\models \phi \rightarrow \theta$

Second step: prove that if $\models \phi \rightarrow \theta$ then $\phi \models \theta$

Prove:

Theorem 2.49.(4) If ϕ and ψ are SL sentences, then $\phi \models \psi$ iff $\phi, \sim\psi \models (A \wedge \sim A)$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally equivalent: Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff all models for θ and ϕ assign them the same truth value, which is the same as saying they entail each other: both $\theta \models \phi$ and $\phi \models \theta$.

example: $A, (A \vee B) \wedge {\sim}B$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally equivalent: Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff all models for θ and ϕ assign them the same truth value, which is the same as saying they entail each other: both $\theta \models \phi$ and $\phi \models \theta$.

example: $A, (A \vee B) \wedge \sim B$

Truth Functionally Equivalent sentences and Truth Tables:

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

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally contradictory: Two sentences θ and ϕ are *truth functionally contradictory* iff all models for θ and ϕ assign them opposite truth values, which is the same as saying that each sentence is TFE to the negation of the other.

example: $(A \leftrightarrow \sim B), ((A \wedge B) \vee (\sim A \wedge \sim B))$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally contradictory: Two sentences θ and ϕ are *truth functionally contradictory* iff all models for θ and ϕ assign them opposite truth values, which is the same as saying that each sentence is TFE to the negation of the other.

example: $(A \leftrightarrow \sim B), ((A \wedge B) \vee (\sim A \wedge \sim B))$

Truth Functionally Contradictory sentences and Truth Tables:

A	B	$(A \leftrightarrow \sim B)$	$((A \wedge B) \vee (\sim A \wedge \sim B))$
T	T	F	T
T	F	T	F
F	T	T	F
F	F	F	T

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally contrary: Two sentences θ and ϕ are *truth functionally contrary* iff they cannot both be true in the same model $\mathfrak{m}$. (This is the same as saying that each entails the negation of the other.)

example: $(A \wedge B), (A \wedge {\sim}B)$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally contrary: Two sentences θ and ϕ are *truth functionally contrary* iff they cannot both be true in the same model $\mathfrak{m}$. (This is the same as saying that each entails the negation of the other.)

example: $(A \wedge B), (A \wedge \sim B)$

Truth Functionally Contrary sentences and Truth Tables:

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

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally subcontrary: Two sentences θ and ϕ are *truth functionally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$. (This is the same as saying that the negation of each entails the other.)

example: $(A \to B), (A \to {\sim}B)$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally subcontrary: Two sentences θ and ϕ are *truth functionally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$. (This is the same as saying that the negation of each entails the other.)

example: $(A \rightarrow B), (A \rightarrow \sim B)$

Truth Functionally Subcontrary sentences and Truth Tables:

A	B	$(A \rightarrow B)$	$(A \rightarrow \sim B)$
T	T	T	F
T	F	F	T
F	T	T	T
F	F	T	T

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally independent: Two sentences θ and ϕ are *truth functionally independent* iff none of the above hold (including entailments), i.e. iff there are four models:

1. A model in which both θ and ϕ are true;
2. A model in which both θ and ϕ are false;
3. A model in which θ is true and ϕ is false; and
4. A model in which θ is false and ϕ is true.

example: $(A \vee B), (A \vee C)$

OTHER RELATIONS BETWEEN SL SENTENCES

Truth functionally independent: Two sentences θ and ϕ are *truth functionally independent* iff none of the above hold (including entailments), i.e. iff there are four models:

1. A model in which both θ and ϕ are true;
2. A model in which both θ and ϕ are false;
3. A model in which θ is true and ϕ is false; and
4. A model in which θ is false and ϕ is true.

example: $(A \vee B), (A \vee C)$

Truth Functionally Independent sentences and Truth Tables:

A	B	C	$(A \vee B)$	$(A \vee C)$
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	T	T
F	T	T	T	T
F	T	F	T	F
F	F	T	F	T
F	F	F	F	F

Truth functionally equivalent: Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff all models for θ and ϕ assign them the same truth value, which is the same as saying they entail each other: both $\theta \models \phi$ and $\phi \models \theta$.

Truth functionally contradictory: Two sentences θ and ϕ are *truth functionally contradictory* iff all models for θ and ϕ assign them opposite truth values, which is the same as saying that each sentence is TFE to the negation of the other.

Truth functionally contrary: Two sentences θ and ϕ are *truth functionally contrary* iff they cannot both be true in the same model $\mathfrak{m}$. (This is the same as saying that each entails the negation of the other.)

Truth functionally subcontrary: Two sentences θ and ϕ are *truth functionally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$. (This is the same as saying that the negation of each entails the other.)

Truth functionally independent: Two sentences θ and ϕ are *truth functionally independent* iff none of the above hold (including entailments), i.e. iff there are four models:

1. A model in which both θ and ϕ are true;
2. A model in which both θ and ϕ are false;
3. A model in which θ is true and ϕ is false; and
4. A model in which θ is false and ϕ is true.

Question: how are these relations related?

Truth functionally equivalent: Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff all models for θ and ϕ assign them the same truth value, which is the same as saying they entail each other: both $\theta \models \phi$ and $\phi \models \theta$.

Truth functionally contradictory: Two sentences θ and ϕ are *truth functionally contradictory* iff all models for θ and ϕ assign them opposite truth values, which is the same as saying that each sentence is TFE to the negation of the other.

Truth functionally contrary: Two sentences θ and ϕ are *truth functionally contrary* iff they cannot both be true in the same model $\mathfrak{m}$. (This is the same as saying that each entails the negation of the other.)

Truth functionally subcontrary: Two sentences θ and ϕ are *truth functionally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$. (This is the same as saying that the negation of each entails the other.)

Truth functionally independent: Two sentences θ and ϕ are *truth functionally independent* iff none of the above hold (including entailments), i.e. iff there are four models:

1. A model in which both θ and ϕ are true;
2. A model in which both θ and ϕ are false;
3. A model in which θ is true and ϕ is false; and
4. A model in which θ is false and ϕ is true.

Question: What relations hold among these sentences?

1. A
2. $A \wedge B$
3. $\sim A \wedge B$
4. $A \rightarrow C$
5. $A \rightarrow \sim C$
6. $(A \wedge B) \vee C$
7. $D \wedge \sim D$

Relations between SL sentences:

Entailment: If Δ is a set of SL sentences and θ is an SL sentence, then the following are equivalent ways to define when Δ entails θ:

1. $\Delta \models \theta$ iff every model for Δ and θ that makes all sentences in Δ true also makes θ true.
2. $\Delta \models \theta$ iff every model for Δ and θ either makes at least one sentence in Δ false or makes θ true.

Truth functionally equivalent: Two sentences θ and ϕ are *truth functionally equivalent* (TFE) iff all models for θ and ϕ assign them the same truth value, which is the same as saying they entail each other: both $\theta \models \phi$ and $\phi \models \theta$.

Truth functionally contradictory: Two sentences θ and ϕ are *truth functionally contradictory* iff all models for θ and ϕ assign them opposite truth values, which is the same as saying that each sentence is TFE to the negation of the other.

Truth functionally contrary: Two sentences θ and ϕ are *truth functionally contrary* iff they cannot both be true in the same model $\mathfrak{m}$. (This is the same as saying that each entails the negation of the other.)

Truth functionally subcontrary: Two sentences θ and ϕ are *truth functionally subcontrary* iff they cannot both be false in the same model $\mathfrak{m}$. (This is the same as saying that the negation of each entails the other.)

Truth functionally independent: Two sentences θ and ϕ are *truth functionally independent* iff none of the above hold (including entailments), i.e. iff there are four models:

1. A model in which both θ and ϕ are true;
2. A model in which both θ and ϕ are false;
3. A model in which θ is true and ϕ is false; and
4. A model in which θ is false and ϕ is true.

Recursive Definitions and Recursive Proofs

Definition 1.1. A *recursive definition* has three clauses:

1. the *base clause(s)*, which specifies a set of objects which unqualifiedly count as meeting the definition,
2. the *generating clause(s)*, which specifies one or more ways of generating (or finding) new objects that meet the definition, and
3. the *closure clause*, which specifies that something meets the definition only if it can be shown to meet the definition by applications of the first two clauses.

Recursive Definitions and Recursive Proofs

Definition 1.1. A *recursive definition* has three clauses:

1. the *base clause(s)*, which specifies a set of objects which unqualifiedly count as meeting the definition,
2. the *generating clause(s)*, which specifies one or more ways of generating (or finding) new objects that meet the definition, and
3. the *closure clause*, which specifies that something meets the definition only if it can be shown to meet the definition by applications of the first two clauses.

Definition 2.2. The *sentences of SL* are given by the following recursive definition:

Base Clause: Every sentence letter is a sentence.

Generating Clauses:

1. If ϕ is a sentence, then so is $\sim\phi$.
2. If ϕ and θ are sentences, then so are both $(\phi \rightarrow \theta)$ and $(\phi \leftrightarrow \theta)$.
3. If all of $\phi_1, \phi_2, \phi_3, \phi_4, \ldots, \phi_n$ are sentences (the list must include at least two sentences and be finite), then so are $(\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \ldots \wedge \phi_n)$ and $(\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4 \vee \ldots \vee \phi_n)$.

Closure Clause: A sequence of symbols is an SL sentence iff its being a sentence follows from the previous two clauses.

Recursive Definitions and Recursive Proofs

Definition 1.2. Let Δ be some set whose members are defined recursively. To prove that all members of Δ have some property ϕ we use a *recursive proof*, which works as follows:

Base Step: Show that everything identified by the base clause of the recursive definition has ϕ.

Inheritance Step: Show that ϕ is inherited; i.e., show that if the previous objects from which new objects are generated (or found) by the generating clause have ϕ then the new ones have ϕ too.

Closure Step: Finally, show that completing the base and inheritance steps is sufficient to show that all members of Δ have ϕ.

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Closure Step:

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \dots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less.

[show: given RA, any sentence of SL generated from $\theta, \theta_1, \theta_2, \dots, \theta_n$ – any sentence of order $k + 1$ – must have exactly as many left parentheses as right]

Closure Step:

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \ldots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less.

[show: given RA, any sentence of SL generated from $\theta, \theta_1, \theta_2, \ldots, \theta_n$ – any sentence of order $k + 1$ – must have exactly as many left parentheses as right]

Closure Step: [conclude that the Base and Inheritance steps are sufficient to establish our target theorem]

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \ldots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less. We want to show that sentences of order $k + 1$ have the same property.

Negation: [show: given RA, $\sim\theta$ must have exactly as many left parentheses as right]

Conditional: [show: given RA, $(\phi \rightarrow \theta)$ must have exactly as many left parentheses as right]

Biconditional: [show: given RA, $(\phi \leftrightarrow \theta)$ must have exactly as many left parentheses as right]

Disjunction: [show: given RA, $(\theta_1 \vee \theta_2 \vee \ldots \vee \theta_n)$ must have exactly as many left parentheses as right]

Conjunction: [show: given RA, $(\theta_1 \wedge \theta_2 \wedge \ldots \wedge \theta_n)$ must have exactly as many left parentheses as right]

Closure Step:

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \dots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less. We want to show that sentences of order $k + 1$ have the same property.

Negation: [show: given RA, $\sim\theta$ must have exactly as many left parentheses as right]

Conditional: [show: given RA, $(\phi \to \theta)$ must have exactly as many left parentheses as right]

Biconditional: [show: given RA, $(\phi \leftrightarrow \theta)$ must have exactly as many left parentheses as right]

Disjunction: [show: given RA, $(\theta_1 \vee \theta_2 \vee \dots \vee \theta_n)$ must have exactly as many left parentheses as right]

Conjunction: [show: given RA, $(\theta_1 \wedge \theta_2 \wedge \dots \wedge \theta_n)$ must have exactly as many left parentheses as right]

Closure Step: Because the inheritance step covers all the ways of generating an SL sentence, for all SL sentences the number of left parentheses is the same as the number of right parentheses.

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \dots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less. We want to show that sentences of order $k+1$ have the same property.

Negation: New sentence $\sim\phi$ of order $k+1$ includes all and only the parentheses in sentence ϕ of order k. By our Recursive Assumption, are equal in number ϕ has the same number of left and right parentheses.

Conditional: [show: given RA, $(\phi \rightarrow \theta)$ must have exactly as many left parentheses as right]

Biconditional: [show: given RA, $(\phi \leftrightarrow \theta)$ must have exactly as many left parentheses as right]

Disjunction: [show: given RA, $(\theta_1 \vee \theta_2 \vee \dots \vee \theta_n)$ must have exactly as many left parentheses as right]

Conjunction: [show: given RA, $(\theta_1 \wedge \theta_2 \wedge \dots \wedge \theta_n)$ must have exactly as many left parentheses as right]

Closure Step: Because the inheritance step covers all the ways of generating an SL sentence, for all SL sentences the number of left parentheses is the same as the number of right parentheses.

Recursive Proof (example 1)

Prove: every (official) sentence of SL has exactly as many left parentheses as right.

Base Step: All atomic sentences (i.e., sentence letters) of SL have zero left parentheses and zero right parentheses. And, of course, $0 = 0$.

Inheritance Step:

Recursive Assumption: Suppose ϕ and $\theta, \theta_1, \theta_2, \ldots, \theta_n$ each have exactly as many left parentheses as right, and are of order k or less. We want to show that sentences of order $k + 1$ have the same property.

Negation: New sentence $\sim\phi$ of order $k + 1$ includes all and only the parentheses in sentence ϕ of order k. By our Recursive Assumption, are equal in number ϕ has the same number of left and right parentheses.

Conditional: New sentence $(\phi \rightarrow \theta)$ of order $k+1$ includes all and only the parentheses in sentences ϕ and θ (of order k or less) together with one new pair of parentheses. By our recursive assumption, ϕ and θ each have the same number of left and right parentheses, so $(\phi \rightarrow \theta)$ has the same number of left and right parentheses.

Biconditional: New sentence $(\phi \leftrightarrow \theta)$ of order $k+1$ includes all and only the parentheses in sentences ϕ and θ (of order k or less) together with one new pair of parentheses. By our recursive assumption, ϕ and θ each have the same number of left and right parentheses, so $(\phi \leftrightarrow \theta)$ has the same number of left and right parentheses.

Disjunction: New sentence $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$ of order $k+1$ includes all and only the parentheses in sentences $\theta_1, \theta_2, ..., \theta_n$ (of order k or less) together with one new pair of parentheses. By our recursive assumption, $\theta_1, \theta_2, ..., \theta_n$ each have the same number of left and right parentheses, so $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$ has the same number of left and right parentheses.

Conjunction: New sentence $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ of order $k+1$ includes all and only the parentheses in sentences $\theta_1, \theta_2, ..., \theta_n$ (of order k or less) together with one new pair of parentheses. By our recursive assumption, $\theta_1, \theta_2, ..., \theta_n$ each have the same number of left and right parentheses, so $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ has the same number of left and right parentheses.

Closure Step: Because the inheritance step covers all the ways of generating an SL sentence, for all SL sentences the number of left parentheses is the same as the number of right parentheses.

Recursive Proof (example 2)

Prove: In every SL sentence there is a subsentence which is TFC.

Base step:

Inheritance step:

- **Recursive Assumption:**
- **Negation:**
- **Conditional:**
- **Biconditional:**
- **Disjunction:**
- **Conjunction:**

Closure step:

Recursive Proof (example 2)

Prove: In every SL sentence there is a subsentence which is TFC.

Base step: In the base case ϕ is atomic, and all atomic sentences are TFC.

Inheritance step:

Recursive Assumption:

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 2)

Prove: In every SL sentence there is a subsentence which is TFC.

Base step: In the base case ϕ is atomic, and all atomic sentences are TFC.

Inheritance step:

Recursive Assumption: Assume that $\phi, \phi_1, \phi_2, ..., \phi_n$ are sentences of SL of order K or less, each of which contains a subsentence which is TFC.

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 2)

Prove: In every SL sentence there is a subsentence which is TFC.

Base step: In the base case ϕ is atomic, and all atomic sentences are TFC.

Inheritance step:

- **Recursive Assumption:** Assume that $\phi, \phi_1, \phi_2, ..., \phi_n$ are sentences of SL of order K or less, each of which contains a subsentence which is TFC.
- **Negation:** $\sim\phi$ is a sentence of order $k+1$ that contains ϕ, which by assumption contains a TFC subsentence. So by the transitivity of the subsentence relation, $\sim\phi$ contains a TFC as a subsentence as well.
- **Conditional:** $(\phi_1 \rightarrow \phi_2)$ is a sentence of order $k+1$ that contains ϕ_1 and ϕ_2, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \rightarrow \phi_2)$ contains a TFC as a subsentence as well.
- **Biconditional:** $(\phi_1 \leftrightarrow \phi_2)$ is a sentence of order $k+1$ that contains ϕ_1 and ϕ_2, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \leftrightarrow \phi_2)$ contains a TFC as a subsentence as well.
- **Disjunction:** $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ is a sentence of order $k+1$ that contains $\phi_1, \phi_2, ..., \phi_n$, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ contains a TFC as a subsentence as well.
- **Conjunction:** $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ is a sentence of order $k+1$ that contains $\phi_1, \phi_2, ..., \phi_n$, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ contains a TFC as a subsentence as well.

Closure step:

Recursive Proof (example 2)

Prove: In every SL sentence there is a subsentence which is TFC.

Base step: In the base case ϕ is atomic, and all atomic sentences are TFC.

Inheritance step:

Recursive Assumption: Assume that $\phi, \phi_1, \phi_2, ..., \phi_n$ are sentences of SL of order K or less, each of which contains a subsentence which is TFC.

Negation: $\sim\phi$ is a sentence of order $k+1$ that contains ϕ, which by assumption contains a TFC subsentence. So by the transitivity of the subsentence relation, $\sim\phi$ contains a TFC as a subsentence as well.

Conditional: $(\phi_1 \rightarrow \phi_2)$ is a sentence of order $k+1$ that contains ϕ_1 and ϕ_2, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \rightarrow \phi_2)$ contains a TFC as a subsentence as well.

Biconditional: $(\phi_1 \leftrightarrow \phi_2)$ is a sentence of order $k+1$ that contains ϕ_1 and ϕ_2, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \leftrightarrow \phi_2)$ contains a TFC as a subsentence as well.

Disjunction: $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ is a sentence of order $k+1$ that contains $\phi_1, \phi_2, ..., \phi_n$, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ contains a TFC as a subsentence as well.

Conjunction: $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ is a sentence of order $k+1$ that contains $\phi_1, \phi_2, ..., \phi_n$, which by assumption each contain a TFC subsentence. So by the transitivity of the subsentence relation, $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ contains a TFC as a subsentence as well.

Closure step: These are all the ways of constructing SL sentences of order $k+1$ out of sentences of order k or less, and in each case if if the order

k or less sentences each have a subsentence which is TFC, then so does the new order $k + 1$ sentence. So, in every SL sentence there is a TFC subsentence.

Recursive Proof (example 3)

Prove: If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ assign the same truth values to each of the sentence letters of ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$, i.e. $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters in θ.

Base step:

Inheritance step:

Recursive Assumption:

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 3)

Prove: If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ assign the same truth values to each of the sentence letters in ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$, i.e. $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters in ϕ.

Base step: In the base case ϕ is an atomic sentence letter. By assumption $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters of SL. Thus, they agree on ϕ. So, ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$.

Inheritance step:

Recursive Assumption:

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 3)

Prove: If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ assign the same truth values to each of the sentence letters in ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$, i.e. $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters in ϕ.

Base step: In the base case ϕ is an atomic sentence letter. By assumption $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters of SL. Thus, they agree on ϕ. So, ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$.

Inheritance step:

Recursive Assumption: Suppose that $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on each sentence $\phi_1, \phi_2, ..., \phi_n$

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 3)

Prove: If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ assign the same truth values to each of the sentence letters in ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$, i.e. $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters in ϕ.

Base step: In the base case ϕ is an atomic sentence letter. By assumption $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters of SL. Thus, they agree on ϕ. So, ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$.

Inheritance step:

Recursive Assumption: Suppose that $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on each sentence $\phi_1, \phi_2, ..., \phi_n$

Negation: $\sim\phi_1$ is true on $\mathfrak{m}$ iff ϕ_1 is false on $\mathfrak{m}$ (by Def of Truth) iff ϕ_1 is false on $\mathfrak{m}^*$ (by RA) iff $\sim\phi_1$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Conditional:

Biconditional:

Disjunction:

Conjunction:

Closure step:

Recursive Proof (example 3)

Prove: If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ assign the same truth values to each of the sentence letters in ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$, i.e. $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters in ϕ.

Base step: In the base case ϕ is an atomic sentence letter. By assumption $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all the sentence letters of SL. Thus, they agree on ϕ. So, ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$.

Inheritance step:

Recursive Assumption: Suppose that $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on each sentence $\phi_1, \phi_2, ..., \phi_n$

Negation: $\sim\phi_1$ is true on $\mathfrak{m}$ iff ϕ_1 is false on $\mathfrak{m}$ (by Def of Truth) iff ϕ_1 is false on $\mathfrak{m}^*$ (by RA) iff $\sim\phi_1$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Conditional: $(\phi_1 \rightarrow \phi_2)$ is true on $\mathfrak{m}$ iff ϕ_1 is false on $\mathfrak{m}$ or ϕ_2 is true on $\mathfrak{m}$ (by Def of Truth), iff ϕ_1 is false on $\mathfrak{m}^*$ or ϕ_2 is true on $\mathfrak{m}^*$ (by RA), iff $(\phi_1 \rightarrow \phi_2)$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Biconditional: $(\phi_1 \leftrightarrow \phi_2)$ is true on $\mathfrak{m}$ iff either both ϕ_1 and ϕ_2 are true on $\mathfrak{m}$, or both are false on $\mathfrak{m}$ (by Def of Truth), iff either both ϕ_1 and ϕ_2 are true on $\mathfrak{m}^*$, or both are false on $\mathfrak{m}^*$ (by RA); iff $(\phi_1 \leftrightarrow \phi_2)$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Disjunction: $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ is true on $\mathfrak{m}$ iff at least one of $\phi_1, \phi_2, ..., \phi_n$ is true on $\mathfrak{m}$ (by Def of Truth), iff at least one of $\phi_1, \phi_2, ..., \phi_n$ is true on $\mathfrak{m}^*$ (by RA); iff $(\phi_1 \vee \phi_2 \vee ... \vee \phi_n)$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Conjunction: $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ is true on $\mathfrak{m}$ iff all of $\phi_1, \phi_2, ..., \phi_n$ are true on $\mathfrak{m}$ (by Def of Truth), iff all of $\phi_1, \phi_2, ..., \phi_n$ are true on $\mathfrak{m}^*$ (by RA); iff $(\phi_1 \wedge \phi_2 \wedge ... \wedge \phi_n)$ is true on $\mathfrak{m}^*$ (by Def of Truth).

Closure step: These are the only ways SL sentences can be generated, and in every case If two models $\mathfrak{m}$ and $\mathfrak{m}^*$ agree on all of the sentence letters of ϕ, then ϕ is true in $\mathfrak{m}$ iff ϕ is true in $\mathfrak{m}^*$

Recursive Proof (example 4)

Prove: no sentence of SL that contains no negations is TFF, i.e. show that every TFF sentence of SL contains at least one '$\sim$'.

Recursive Proof (example 4)

Prove: no sentence of SL that contains no negations is TFF, i.e. show that every TFF sentence of SL contains at least one '$\sim$'.

Base step: If ϕ is an atomic sentence, then $\mathfrak{m}^+$ makes it true (by Def of $\mathfrak{m}^+$).

Inheritance step: Recursive Assumption: Suppose the sentences θ_1, θ_2, ... θ_n are true in the model $\mathfrak{m}^+$

Recursive Proof (example 4)

Prove: no sentence of SL that contains no negations is TFF, i.e. show that every TFF sentence of SL contains at least one '$\sim$'.

Base step: If ϕ is an atomic sentence, then $\mathfrak{m}^+$ makes it true (by Def of $\mathfrak{m}^+$).

Inheritance step: Recursive Assumption: Suppose the sentences θ_1, θ_2, ... θ_n are true in the model $\mathfrak{m}^+$

It follows immediately from the recursive assumption and the definition of Truth in a model that $(\theta_1 \rightarrow \theta_2)$, $(\theta_1 \leftrightarrow \theta_2)$, $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$, and $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ are all true on $\mathfrak{m}^+$.

Recursive Proof (example 4)

Prove: no sentence of SL that contains no negations is TFF, i.e. show that every TFF sentence of SL contains at least one '$\sim$'.

Base step: If ϕ is an atomic sentence, then $\mathfrak{m}^+$ makes it true (by Def of $\mathfrak{m}^+$).

Inheritance step: Recursive Assumption: Suppose the sentences θ_1, θ_2, ... θ_n are true in the model $\mathfrak{m}^+$

It follows immediately from the recursive assumption and the definition of Truth in a model that $(\theta_1 \rightarrow \theta_2)$, $(\theta_1 \leftrightarrow \theta_2)$, $(\theta_1 \vee \theta_2 \vee ... \vee \theta_n)$, and $(\theta_1 \wedge \theta_2 \wedge ... \wedge \theta_n)$ are all true on $\mathfrak{m}^+$.

Conclusion: All relevant sentences of SL (i.e., those without negations) are either atomic or generated by means covered in the inheritance clause. Thus it has been shown that every sentence without negations is true on $\mathfrak{m}^+$, and thus is not TFF. Equivalently, every TFF sentence contains at least one negation.

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ.

Base step:

Inheritance step:

Conclusion:

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ.

Base step: In the base case, ϕ is a sentence letter of SL, in which case ϕ contains one subsentence (ϕ itself), which is equal to its number of negations (zero) in ϕ plus its number of sentence letter tokens in ϕ (1: ϕ itself) plus the number of left parentheses in ϕ (zero)

Inheritance step:

Conclusion:

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ. Shorter: every sentence letter in SL has property (shorter: has property SS=NT+SLT+LPT)

Base step: In the base case, ϕ is a sentence letter of SL, in which case ϕ contains one subsentence (ϕ itself), which is equal to its number of negations (zero) in ϕ plus its number of sentence letter tokens in ϕ (1: ϕ itself) plus the number of left parentheses in ϕ (zero). So, ϕ has property SS=NT+SLT+LPT.

Inheritance step:

Recursive assumption: Suppose the sentences ϕ_1, ϕ_2, ... ϕ_n each has property SS=NT+SLT+LPT.

Negation:

Conditional:

Biconditional:

Disjunction:

Conjunction:

Conclusion:

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ. Shorter: every sentence letter in SL has SS=NT+SLT+LPT.

Base step: In the base case, ϕ is a sentence letter of SL, in which case ϕ contains one subsentence (ϕ itself), which is equal to its number of negations (zero) in ϕ plus its number of sentence letter tokens in ϕ (1: ϕ itself) plus the number of left parentheses in ϕ (zero). So, ϕ has property SS=NT+SLT+LPT.

Inheritance step:

- **Recursive assumption:** Suppose the sentences ϕ_1, ϕ_2, ... ϕ_n each has property SS=NT+SLT+LPT.
- **Negation:** Compared to ϕ, $\sim\phi$ has 1 additional subsentence ($\sim\phi$ itself; by Def of Subsentence)), 1 additional token of '$\sim$', 0 additional sentence letter tokens, and 0 additional left parentheses (by Def of SL sentence). So, $\sim\phi$ has SS=NT+SLT+LPT.
- **Conditional:**
- **Biconditional:**
- **Disjunction:**
- **Conjunction:**

Conclusion:

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ. Shorter: every sentence letter in SL has property (shorter: has property SS=NT+SLT+LPT)

Base step: In the base case, ϕ is a sentence letter of SL, in which case ϕ contains one subsentence (ϕ itself), which is equal to its number of negations (zero) in ϕ plus its number of sentence letter tokens in ϕ (1: ϕ itself) plus the number of left parentheses in ϕ (zero). So, ϕ has property SS=NT+SLT+LPT.

Inheritance step:

Recursive assumption: Suppose the sentences ϕ_1, ϕ_2, ... ϕ_n of order k or less each has property SS=NT+SLT+LPT.

Negation: Compared to ϕ, $\sim\phi$ has 1 additional subsentence ($\sim\phi$ itself; by Def of Subsentence)), 1 additional token of '$\sim$', 0 additional sentence letter tokens, and 0 additional left parentheses (by Def of SL sentence). So, $\sim\phi$ has SS=NT+SLT+LPT.

Conditional: Compared to ϕ_1, ϕ_2, $(\phi_1 \rightarrow \phi_2)$ has 1 additional subsentence ($(\phi_1 \rightarrow \phi_2)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \rightarrow \phi_2)$ has property SS=NT+SLT+LPT.

Biconditional: Compared to ϕ_1, ϕ_2, $(\phi_1 \leftrightarrow \phi_2)$ has 1 additional subsentence ($(\phi_1 \leftrightarrow \phi_2)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \leftrightarrow \phi_2)$ has property SS=NT+SLT+LPT.

Disjunction: Compared to $\phi_1, \phi_2, ..., \phi_n$, $(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ has 1 additional subsentence ($(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ has property SS=NT+SLT+LPT.

Conjunction: Compared to $\phi_1, \phi_2, ..., \phi_n$, $(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ has 1 additional subsentence ($(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ has property SS=NT+SLT+LPT.

Conclusion:

Recursive Proof (example 5)

Prove: for any official SL sentence ϕ, the number of subsentences in ϕ is equal to: the number of tokens of sentence letters in ϕ plus the number of tokens of negation in ϕ plus the number of tokens of left parentheses in ϕ. Shorter: every sentence letter in SL has property (shorter: has property SS=NT+SLT+LPT)

Base step: In the base case, ϕ is a sentence letter of SL, in which case ϕ contains one subsentence (ϕ itself), which is equal to its number of negations (zero) in ϕ plus its number of sentence letter tokens in ϕ (1: ϕ itself) plus the number of left parentheses in ϕ (zero). So, ϕ has property SS=NT+SLT+LPT.

Inheritance step:

Recursive assumption: Suppose the sentences ϕ_1, ϕ_2, ... ϕ_n of order k or less each has property SS=NT+SLT+LPT.

Negation: Compared to ϕ, $\sim\phi$ has 1 additional subsentence ($\sim\phi$ itself; by Def of Subsentence)), 1 additional token of '$\sim$', 0 additional sentence letter tokens, and 0 additional left parentheses (by Def of SL sentence). So, $\sim\phi$ has SS=NT+SLT+LPT.

Conditional: Compared to ϕ_1, ϕ_2, $(\phi_1 \rightarrow \phi_2)$ has 1 additional subsentence ($(\phi_1 \rightarrow \phi_2)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \rightarrow \phi_2)$ has property SS=NT+SLT+LPT.

Biconditional: Compared to ϕ_1, ϕ_2, $(\phi_1 \leftrightarrow \phi_2)$ has 1 additional subsentence ($(\phi_1 \leftrightarrow \phi_2)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \leftrightarrow \phi_2)$ has property SS=NT+SLT+LPT.

Disjunction: Compared to $\phi_1, \phi_2, ..., \phi_n$, $(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ has 1 additional subsentence ($(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \vee \phi_2, \vee, ..., \vee\phi_n)$ has property SS=NT+SLT+LPT.

Conjunction: Compared to $\phi_1, \phi_2, ..., \phi_n$, $(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ has 1 additional subsentence ($(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ itself; by Def of Subsentence)), 0 additional tokens of '$\sim$', 0 additional sentence letter tokens, and 1 additional left parentheses (by Def of SL sentence). So, $(\phi_1 \wedge \phi_2, \wedge, ..., \wedge\phi_n)$ has property SS=NT+SLT+LPT.

Conclusion: These are all the ways to generate a SL sentence of order $k+1$ from SL sentences ϕ_1, ϕ_2, ... ϕ_n of order k or less, and in every case property SS=NT+SLT+LPT is inherited. So, every sentence of SL has property SS=NT+SLT+LPT.

Disjunctive Normal Form

Theorem 2.67. Truth Functional Equivalence Replacement: Let ϕ be a subsentence of θ. If ϕ and ϕ^* are truth functionally equivalent, and θ^* is the result of replacing one occurrence of ϕ by ϕ^* in θ, then θ and θ^* are truth functionally equivalent.

Base Step: Suppose that θ is an atomic sentence. In this case θ and ϕ are the same. So, θ^* and ϕ^* are the same. Thus, by hypothesis (the hypothesis being that ϕ and ϕ^* are truth functionally equivalent), θ^* and θ are truth functionally equivalent.

Inheritance Step: For this proof we must consider each connective separately.

Negation: Suppose θ is a negation $\sim\psi$, for some sentence ψ which either is identical to ϕ, or of which ϕ is a subsentence.

Assume that ψ and ψ^* (ψ^* the result of replacing at least one occurrence of ϕ with ϕ^* in ψ) are truth functionally equivalent, i.e. have the same truth value on every model, and are of order k or less. (This is our recursive assumption.) Now, by supposition θ^* is the same sentence as $(\sim\psi)^*$, which with a little thought one can see is the same sentence as $\sim(\psi^*)$.

Next, note that by the definition of true in a model, $\sim(\psi^*)$ is true in a model $\mathfrak{m}$ iff ψ^* is false in $\mathfrak{m}$ iff ψ is false in $\mathfrak{m}$ iff $\sim\psi$ is true in $\mathfrak{m}$. So, θ^* is true in $\mathfrak{m}$ iff $\sim\psi$ is true in $\mathfrak{m}$.

So, θ^* is true in $\mathfrak{m}$ iff θ is true in $\mathfrak{m}$, so by definition θ and θ^* are truth functionally equivalent.

Conditional: Suppose θ is a conditional $(\psi_1 \rightarrow \psi_2)$, were at least one of ψ_1 and ψ_2 has ϕ as a subsentence, or is identical to ϕ. So θ^* is $(\psi_1 \rightarrow \psi_2)^*$, which with some thought one can see is either (case 1) $({\psi_1}^* \rightarrow \psi_2)$ or (case 2) $(\psi_1 \rightarrow {\psi_2}^*)$.

Suppose it is the first case, and assume that ψ_1 is truth functionally equivalent to ${\psi_1}^*$, and that both are of order k or less. (This is our recursive assumption.)

We know that $({\psi_1}^* \to \psi_2)$ is true in a model $\mathfrak{m}$ iff ${\psi_1}^*$ is false in $\mathfrak{m}$ or ψ_2 is true in $\mathfrak{m}$. But that holds iff ψ_1 is false in $\mathfrak{m}$ or ψ_2 is true in $\mathfrak{m}$, and that holds iff $(\psi_1 \to \psi_2)$ is true in $\mathfrak{m}$.

In the first case, θ^* is true in $\mathfrak{m}$ iff θ is true in $\mathfrak{m}$, so by definition θ and θ^* are truth functionally equivalent.

Showing that θ and θ^* are truth functionally equivalent in case 2 is left to the reader.

Biconditional: Showing that θ and θ^* are truth functionally equivalent when θ is a biconditional of the form $(\psi_1 \leftrightarrow \psi_2)$ is left to the reader.

Conjunction: Suppose that θ is a conjunction $(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \ldots \wedge \psi_n)$, where at least one of ψ_1, ψ_2, ψ_3, ... ψ_n has ϕ as a subsentence, or is identical to ϕ.

So θ^* is $(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \ldots \wedge \psi_n)^*$. Since θ^* is the result of replacing <u>one</u> occurrence of ϕ in θ with ϕ^*, it's not hard to see that $(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \ldots \wedge \psi_n)^*$ can <u>only</u> one of either $(\psi_1^* \wedge \psi_2 \wedge \ldots \wedge \psi_n)$ or $(\psi_1 \wedge \psi_2^* \wedge \ldots \wedge \psi_n)$ or ... or $(\psi_1 \wedge \psi_2 \wedge \ldots \wedge \psi_n^*)$. But, clearly, there's no difference which it is. So, without loss of generality say that θ^* is $(\psi_1 \wedge \psi_2^* \wedge \psi_3 \wedge \ldots \wedge \psi_n)$.

As before, assume that ψ_2 and ${\psi_2}^*$ are truth functionally equivalent, and are of order k or less. (This is our recursive assumption.) So, for any model $\mathfrak{m}$, θ is true in $\mathfrak{m}$ iff $(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \ldots \wedge \psi_n)$ is true in $\mathfrak{m}$ iff $(\psi_1 \wedge \psi_2^* \wedge \psi_3 \wedge \ldots \wedge \psi_n)$ is is true in $\mathfrak{m}$ iff $(\psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \ldots \wedge \psi_n)^*$ is true in $\mathfrak{m}$ iff θ^* is true in $\mathfrak{m}$.

Therefore θ and θ^* are truth functionally equivalent.

Disjunction: The disjunction case is similar and it is left to the reader.

Closure Step: Those are the only ways SL sentences can be formed; hence the theorem is proved.

Disjunctive Normal Form

Observation: sometimes a sentence of SL is hard for us to read and to process, even though a TFE sentence is easy to read and process

Examples:

1. $\sim[A \rightarrow \sim(C \wedge B)] \rightarrow (A \rightarrow C)$
2. $A \wedge R \wedge A \wedge \sim R$
3. $\sim(\sim[A \rightarrow \sim R] \rightarrow (A \rightarrow R))$
4. $[\sim A \vee \sim C \vee \sim B] \vee [\sim A \vee C]$

Disjunctive Normal Form

Observation: sometimes a sentence of SL is hard for us to read and to process, even though a TFE sentence is easy to read and process

Examples:

1. $\sim[A \rightarrow \sim(C \wedge B)] \rightarrow (A \rightarrow C)$
2. $A \wedge R \wedge A \wedge \sim R$
3. $\sim(\sim[A \rightarrow \sim R] \rightarrow (A \rightarrow R))$
4. $[\sim A \vee \sim C \vee \sim B] \vee [\sim A \vee C]$

NB:

- 1. and 4. are TFE
- 2. and 3. are TFE

2. and 4. are in *Disjunctive Normal Form*:

A SL sentence is in *disjunctive normal form* (DNF) iff

1. it contains no conditional ($\rightarrow$) or biconditional ($\leftrightarrow$),
2. negations ($\sim$) only govern sentence letters, and
3. no conjunction ($\wedge$) contains a disjunction ($\vee$) as a subsentence.

Disjunctive Normal Form

Recipe for translating sentence ϕ into DNF:

Step A: If a subsentence of ϕ is of the form $(\psi \rightarrow \theta)$ or $(\theta \leftrightarrow \psi)$, replace the subsentence by $(\sim\psi \vee \theta)$ or $(\psi \wedge \theta) \vee (\sim\psi \wedge \sim\theta)$ respectively. Repeat as necessary to obtain a sentence ϕ' without $\rightarrow$'s or $\leftrightarrow$'s.

Step B:

1. Replace any subsentence of the form $\sim\sim\psi$ in ϕ' with ψ.
2. Replace any subsentence of the form $\sim(\psi \wedge \theta)$ in ϕ' with $(\sim\psi \vee \sim\theta)$.
3. Replace $\sim(\psi \vee \theta)$ in ϕ' with $(\sim\psi \wedge \sim\theta)$.

Repeat as necessary to obtain ϕ'' in which negations govern nothing but sentence letters.

Step C: The only thing that could prevent ϕ'' from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence $\theta \wedge (\psi_1 \vee \psi_2 \vee ... \vee \psi_n)$, or the reverse $(\psi_1 \vee \psi_2 \vee ... \vee \psi_n) \wedge \theta$. Those subsentences can be replaced by the equivalent $(\psi_1 \wedge \theta) \vee (\psi_2 \wedge \theta) \vee ... \vee (\psi_n \wedge \theta)$. Repeat as necessary.

Disjunctive Normal Form

Recipe for translating sentence ϕ into DNF:

Step A: If a subsentence of ϕ is of the form $(\psi \rightarrow \theta)$ or $(\theta \leftrightarrow \psi)$, replace the subsentence by $(\sim\psi \vee \theta)$ or $(\psi \wedge \theta) \vee (\sim\psi \wedge \sim\theta)$ respectively. Repeat as necessary to obtain a sentence $\phi^{'}$ without $\rightarrow$'s or $\leftrightarrow$'s.

Example: $\sim[\sim(A \rightarrow \sim(C \wedge B)) \rightarrow (A \rightarrow C))]$

1. $\sim[\sim(A \rightarrow \sim(C \wedge B)) \rightarrow (A \rightarrow C)]$
2. $\sim[\sim(A \rightarrow \sim(C \wedge B)) \rightarrow (\sim A \vee C)]$
3. $\sim[\sim(\sim A \vee (\sim(C \wedge B))) \rightarrow (\sim A \vee C)]$
4. $\sim[\sim\sim(\sim A \vee (\sim(C \wedge B))) \vee (\sim A \vee C)]$

Disjunctive Normal Form

Recipe for translating sentence ϕ into DNF:

Step B:

1. Replace any subsentence of the form $\sim\sim\psi$ in ϕ' with ψ.
2. Replace any subsentence of the form $\sim(\psi \wedge \theta)$ in ϕ' with $(\sim\psi \vee \sim\theta)$.
3. Replace $\sim(\psi \vee \theta)$ in ϕ' with $(\sim\psi \wedge \sim\theta)$.

Repeat as necessary to obtain ϕ'' in which negations govern nothing but sentence letters.

4. $\sim[\sim\sim(\sim A \vee (\sim(C \wedge B))) \vee (\sim A \vee C)]$
5. $\sim[(\sim A \vee (\sim(C \wedge B))) \vee (\sim A \vee C)]$
6. $\sim[(\sim A \vee (\sim C \vee \sim B)) \vee (\sim A \vee C)]$
7. $\sim(\sim A \vee (\sim C \vee \sim B)) \wedge \sim(\sim A \vee C)$
8. $\sim\sim A \wedge \sim(\sim C \vee \sim B)) \wedge \sim(\sim A \vee C)$
9. $\sim\sim A \wedge \sim(\sim C \vee \sim B)) \wedge \sim\sim A \wedge \sim C$
10. $\sim\sim A \wedge (\sim\sim C \wedge \sim\sim B)) \wedge \sim\sim A \wedge \sim C$
11. $A \wedge (C \wedge B)) \wedge A \wedge \sim C$

Disjunctive Normal Form

Recipe for translating sentence ϕ into DNF:

Step C: The only thing that could prevent ϕ'' from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence $\theta \wedge (\psi_1 \vee \psi_2 \vee ... \vee \psi_n)$, or the reverse $(\psi_1 \vee \psi_2 \vee ... \vee \psi_n) \wedge \theta$. Those subsentences can be replaced by the equivalent $(\psi_1 \wedge \theta) \vee (\psi_2 \wedge \theta) \vee ... \vee (\psi_n \wedge \theta)$. Repeat as necessary.

11. $A \wedge (C \wedge B)) \wedge A \wedge {\sim}C$

Disjunctive Normal Form

Recipe for translating sentence ϕ into DNF:

Step C: The only thing that could prevent ϕ'' from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence $\theta \wedge (\psi_1 \vee \psi_2 \vee ... \vee \psi_n)$, or the reverse $(\psi_1 \vee \psi_2 \vee ... \vee \psi_n) \wedge \theta$. Those subsentences can be replaced by the equivalent $(\psi_1 \wedge \theta) \vee (\psi_2 \wedge \theta) \vee ... \vee (\psi_n \wedge \theta)$. Repeat as necessary.

11. $A \wedge (C \wedge B)) \wedge A \wedge {\sim}C$

11. is in *Disjunctive Normal Form*:

A SL sentence is in *disjunctive normal form* (DNF) iff

1. it contains no conditional ($\rightarrow$) or biconditional ($\leftrightarrow$),
2. negations (${\sim}$) only govern sentence letters, and
3. no conjunction ($\wedge$) contains a disjunction ($\vee$) as a subsentence.

Disjunctive Normal Form

We can prove that 1. and 11. are TFE with a truth table:

A	B	C	$\sim[\sim(A \to \sim(C \wedge B)) \to (A \to C)]$	$A \wedge (C \wedge B) \wedge A \wedge \sim C)$
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F

Disjunctive Normal Form

We can prove that 1. and 11. are TFE with a truth table:

A	B	C	$\sim[\sim(A \rightarrow \sim(C \wedge B)) \rightarrow (A \rightarrow C)]$	$A \wedge (C \wedge B) \wedge A \wedge \sim C)$
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F
F	F	F	F	F

But! We didn't need the truth table to see that 11. is TFF.

11. $A \wedge (C \wedge B) \wedge A \wedge \sim C)$

Checking for TFF with DNF: if a TFF sentence is in DNF, then *every* disjunct will contain the conjunction with both ϕ and $\sim\phi$ as conjuncts

Disjunctive Normal Form

Checking for TFT with DNF:

- ϕ is TFF iff $\sim\phi$ is TFT.
- So, we can check whether some sentence θ is true by
  1. consider θ's negation, $\sim\theta$
  2. convert $\sim\theta$ to DNF
  3. determine whether $\sim\theta$ is TFF.

Disjunctive Normal Form

Checking for TFT with DNF:

- ϕ is TFF iff $\sim\phi$ is TFT.
- So, we can check whether some sentence θ is true by
  1. consider θ's negation, $\sim\theta$
  2. convert $\sim\theta$ to DNF
  3. determine whether $\sim\theta$ is TFF.

Example: is $A \rightarrow (B \rightarrow A)$ TFT? Consider the DNF of it's negation, $\sim(A \rightarrow (B \rightarrow A))$:

1. $\sim(A \rightarrow (B \rightarrow A)$
2. $\sim(A \rightarrow (\sim B \vee A))$
3. $\sim(\sim A \vee (\sim B \vee A))$
4. $\sim\sim A \wedge \sim(\sim B \vee A)$
5. $\sim\sim A \wedge \sim\sim B \wedge \sim A$
6. $A \wedge B \wedge \sim A$

Lesson:

- sentence 6. is TFF
- 6. and 1. are TFE, so 1. is TFF as well
- $A \rightarrow (B \rightarrow A)$ is TFE to the negation of 1. (and also 6.), so $A \rightarrow (B \rightarrow A)$ it TFT

Disjunctive Normal Form

Step A: If a subsentence of ϕ is of the form $(\psi \rightarrow \theta)$ or $(\theta \leftrightarrow \psi)$, replace the subsentence by $(\sim\psi \vee \theta)$ or $(\psi \wedge \theta) \vee (\sim\psi \wedge \sim\theta)$ respectively. Repeat as necessary to obtain a sentence $\phi^{'}$ without $\rightarrow$'s or $\leftrightarrow$'s.

Step B:

1. Replace any subsentence of the form $\sim\sim\psi$ in $\phi^{'}$ with ψ.
2. Replace any subsentence of the form $\sim(\psi \wedge \theta)$ in $\phi^{'}$ with $(\sim\psi \vee \sim\theta)$.
3. Replace $\sim(\psi \vee \theta)$ in $\phi^{'}$ with $(\sim\psi \wedge \sim\theta)$.

Repeat as necessary to obtain $\phi^{''}$ in which negations govern nothing but sentence letters.

Step C: The only thing that could prevent $\phi^{''}$ from being in DNF is that some conjunctions govern some disjunctions, i.e., there is a subsentence $\theta \wedge (\psi_1 \vee \psi_2 \vee ... \vee \psi_n)$, or the reverse $(\psi_1 \vee \psi_2 \vee ... \vee \psi_n) \wedge \theta$. Those subsentences can be replaced by the equivalent $(\psi_1 \wedge \theta) \vee (\psi_2 \wedge \theta) \vee ... \vee (\psi_n \wedge \theta)$. Repeat as necessary.

Translate the following into DNF:

1. $\sim(Q \rightarrow R) \vee (Q \rightarrow \sim R)$
2. $O \wedge (O \rightarrow Q)$
3. $[(Q \rightarrow R) \rightarrow Q] \rightarrow Q$
4. $\sim(Q \rightarrow R) \wedge (O \vee P)$

Truth Functional Expressiveness

NB: there are 16 possible functions (f_n) from ordered pairs of truth values to truth values (assuming 2 truth values):[1]

ϕ	θ	$\phi(f_1)\theta$	$\phi(f_2)\theta$	$\phi(f_3)\theta$	$\phi(f_4)\theta$	$\phi(f_5)\theta$	$\phi(f_6)\theta$	$\phi(f_7)\theta$	$\phi(f_8)\theta$
T	T	F	T	F	F	F	T	F	F
T	F	F	F	T	F	F	T	T	F
F	T	F	F	F	T	F	F	T	T
F	F	F	F	F	F	T	F	F	T

ϕ	θ	$\phi(F_9)\theta$	$\phi(f_{10})\theta$	$\phi(f_{11})\theta$	$\phi(f_{12})\theta$	$\phi(f_{13})\theta$	$\phi(f_{14})\theta$	$\phi(f_{15})\theta$	$\phi(f_{16})\theta$
T	T	T	F	T	F	T	T	T	T
T	F	F	T	F	T	F	T	T	T
F	T	F	F	T	T	T	F	T	T
F	F	T	T	F	T	T	T	F	T

[1]NB I'm departing from our textbook here in treating conjunction and disjunction as functions from ordered *pairs* of truth values to truth values.

Truth Functional Expressiveness

NB: there are 16 possible functions (f_n) from ordered pairs of truth values to truth values (assuming 2 truth values):

ϕ	θ	$\phi(f_1)\theta$	$\phi \wedge \theta$	$\phi(f_3)\theta$	$\phi(f_4)\theta$	$\phi(f_5)\theta$	$\phi(f_6)\theta$	$\phi(f_7)\theta$	$\phi(f_8)\theta$
T	T	F	T	F	F	F	T	F	F
T	F	F	F	T	F	F	T	T	F
F	T	F	F	F	T	F	F	T	T
F	F	F	F	F	F	T	F	F	T

ϕ	θ	$\phi \leftrightarrow \theta$	$\phi(f_{10})\theta$	$\phi(f_{11})\theta$	$\phi(f_{12})\theta$	$\phi \rightarrow \theta$	$\phi(f_{14})\theta$	$\phi \vee \theta$	$\phi(f_{16})\theta$
T	T	T	F	T	F	T	T	T	T
T	F	F	T	F	T	F	T	T	T
F	T	F	F	T	T	T	F	T	T
F	F	T	T	F	T	T	T	F	T

Truth Functional Expressiveness

NB: there are 16 possible functions (f_n) from ordered pairs of truth values to truth values (assuming 2 truth values):

ϕ	θ	$\phi(f_1)\theta$	$\phi \wedge \theta$	$\phi(f_3)\theta$	$\phi(f_4)\theta$	$\phi \downarrow \theta$	$\phi(f_6)\theta$	$\phi(f_7)\theta$	$\phi(f_8)\theta$
T	T	F	T	F	F	F	T	F	F
T	F	F	F	T	F	F	T	T	F
F	T	F	F	F	T	F	F	T	T
F	F	F	F	F	F	T	F	F	T

ϕ	θ	$\phi \leftrightarrow \theta$	$\phi(f_{10})\theta$	$\phi(f_{11})\theta$	$\phi \vert \theta$	$\phi \rightarrow \theta$	$\phi(f_{14})\theta$	$\phi \vee \theta$	$\phi(f_{16})\theta$
T	T	T	F	T	F	T	T	T	T
T	F	F	T	F	T	F	T	T	T
F	T	F	F	T	T	T	F	T	T
F	F	T	T	F	T	T	T	F	T

NAND (‘|’): $\phi|\theta$ is true whenever at least one component is false

NOR (‘↓’): $\phi \downarrow \theta$ is true only when all components are false

Truth Functional Expressiveness

NB: every pair of sentences of SL of form $\phi \downarrow \theta$ and $\sim(\phi \vee \theta)$ are TFE:

ϕ	θ	$\phi \downarrow \theta$	$\sim(\phi \vee \theta)$
T	T	F	F
T	F	F	F
F	T	F	F
F	F	T	T

Truth Functional Expressiveness

NB: every pair of sentences of SL of form $\phi \downarrow \theta$ and $\sim(\phi \vee \theta)$ are TFE:

ϕ	θ	$\phi \downarrow \theta$	$\sim(\phi \vee \theta)$
T	T	F	F
T	F	F	F
F	T	F	F
F	F	T	T

Similarly, every pair of sentences of SL of form $\phi|\theta$ and $\sim(\phi \wedge \theta)$ are TFE:

ϕ	θ	$\phi|\theta$	$\sim(\phi \wedge \theta)$
T	T	F	F
T	F	T	T
F	T	T	T
F	F	T	T

Lesson:

- adding '|' or '↓' to SL do not increase its expressive power, i.e. allow SL to express any sentences that are not TFE to a sentence already expressible in SL

Truth Functional Expressiveness

There's nothing special about '$\wedge$' and '$\vee$':

ϕ	θ	$\phi \downarrow \theta$	$\sim(\phi \downarrow \theta)$	$\phi \vee \theta$
T	T	F	T	T
T	F	F	T	T
F	T	F	T	T
F	F	T	F	F

And:

ϕ	θ	$\phi \vert \theta$	$\sim(\phi \vert \theta)$	$\phi \wedge \theta$
T	T	F	T	T
T	F	T	F	F
F	T	T	F	F
F	F	T	F	F

Lesson:

- adding '$|$' or '$\downarrow$' *and removing* '$\wedge$' *and* '$\vee$' *from SL do not decrease its expressive power*