Sequent Systems for Propositional Logic

The resolution proof system has tremendous computational advantages over the Hilbert system. However, these advantages come at a price; namely:

(a) The computation is restricted to data which are clauses, or at least which are in CNF.
(b) The proof system is complete only for refutations. This requires that the negation of the goal be added to the hypotheses, resulting in somewhat "unnatural" proofs.

An important and interesting question is that which asks whether there is a proof system which has the "naturalness" of the Hilbert system, while at the same time lending itself to computational methods.

The answer is, to some degree, yes. In this set of notes, one such system will be developed.

Important: The sequent proof system developed in Section 3.3 of the textbook, while similar to the one developed in these slides, is not identical. Unfortunately, the system developed in the textbook has many of the flaws of the Hilbert system, so we choose to develop a better, alternate system in these slides.

Sequents:

Definition: A *sequent* is an ordered pair (Λ_1, Λ_2) of finite sequences of wff's.
Notational convention: Throughout this section of notes, unless stated otherwise, a finite propositional $\mathcal{L}$ will be fixed.

Sequences will be written between angular brackets. Thus, a sequent is of the form

$$(\langle\lambda_{11}, \lambda_{12}, .., \lambda_{1m}\rangle, \langle\lambda_{21}, \lambda_{22}, .., \lambda_{2n}\rangle)$$

with the λ_{ij}'s wff's. Either or both of the sequences may be empty.

There is a special notation which is often used for sequents. The alternate notation for

$$(\langle\lambda_{11}, \lambda_{12}, .., \lambda_{1m}\rangle, \langle\lambda_{21}, \lambda_{22}, .., \lambda_{2n}\rangle)$$

is

$$\lambda_{11}, \lambda_{12}, .., \lambda_{1m} \Rightarrow \lambda_{21}, \lambda_{22}, .., \lambda_{2n}$$

Example:

$(\langle A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E), \neg E, \neg C\rangle, \langle\ \rangle)$

and

$(\langle A, (\neg B \vee C \vee D), (\neg D \vee E)\rangle, \langle E, C, (A \wedge \neg B)\rangle)$

are each sequents. In the common sequence notation, they are written as

$A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E), \neg E, \neg C \Rightarrow$

and

$A, (\neg B \vee C \vee D), (\neg D \vee E) \Rightarrow E, C, (A \wedge \neg B)$

Although it is a bit more cryptic, the latter notation is very widely used, and so it will be emphasized within these slides.

Important: Do not confuse the symbol $\rightarrow$ with the symbol $\Rightarrow$.

- $\rightarrow$ is a logical connective in wff's.
- $\Rightarrow$ is a connector for defining sequents.

Definition: An interpretation v in L is a *model* of a sequent

$$\Lambda_1 \Rightarrow \Lambda_2 \;=\; \lambda_{11}, \lambda_{12}, .., \lambda_{1m} \Rightarrow \lambda_{21}, \lambda_{22}, .., \lambda_{2n}$$

precisely in the case that it is a model of

$$\neg\lambda_{11} \vee \neg\lambda_{12} \vee .. \vee \neg\lambda_{1m} \vee \lambda_{21} \vee \lambda_{22} \vee .. \vee \lambda_{2n}$$

Equivalently, it is a *model* of the sequent if it is a model of the *corresponding wff,* which is

$$(\lambda_{11} \wedge \lambda_{12} \wedge .. \wedge \lambda_{1m}) \rightarrow (\lambda_{21} \vee \lambda_{22} \vee .. \vee \lambda_{2n})$$

In this case, we write

$$v \models \Lambda_1 \Rightarrow \Lambda_2$$

Furthermore, the Mod(-) notation is extended to both single sequents and to sets of sequents, in the obvious fashion.

The full definition of $\models$ is also extended to sequents. That is, if Σ is a set of sequents, and if $\Lambda_1 \Rightarrow \Lambda_2$ is also a sequent, then

$$\Sigma \models \Lambda_1 \Rightarrow \Lambda_2$$

denotes that $\mathrm{Mod}(\Sigma) \subseteq \mathrm{Mod}(\Lambda_1 \Rightarrow \Lambda_2)$.

A sequent is *valid* (or a *sequent tautology)* if it is a model of every interpretation.

Proposition: A sequent is valid iff the corresponding wff is a tautology. □

Note: The examples on the previous slides are valid, although this requires some work to prove. (Establishing this validity computationally is the ultimate focus of this discussion.)

Definition: A sequent (Γ_1, Γ_2) is *atomic* if both Γ_1 and Γ_2 consist entirely of proposition names.

Example: A, B, C $\Rightarrow$ B, D, E and
A, B, C $\Rightarrow$ D, E are atomic sequents, as are
A, B, C $\Rightarrow$ and the empty sequent $\Rightarrow$.

Example: A, B, C $\Rightarrow$ B, $\neg$D, E and
$\neg$A, B, C $\Rightarrow$ are not atomic sequents, nor are the sequent examples on the previous slides.

It should be remarked that the order of the elements in a sequent has no particular significance. One could just as well work with sets of formulas.

The Gentzen system G′:

Definition: The axiom schema $Ax_{G'}$ is defined as follows:

$$\Gamma_1, \alpha, \Gamma_2 \Rightarrow \Gamma_3, \alpha, \Gamma_4$$

or, equivalently,

$$(\langle \Gamma_1, \alpha, \Gamma_2 \rangle, \langle \Gamma_3, \alpha, \Gamma_4 \rangle)$$

Here α is to be bound to a wff and the Γ_i's are to be bound to arbitrary sequences of wff's.

Any instance of this schema is called an *axiom* of G′. An *indecomposable axiom* is any sequent of the form

$$P_1, p, P_2 \Rightarrow P_3, p, P_4$$

with p a proposition names, and in which each of the P_i's is a sequence of proposition names. In other words, an indecomposable axiom is an atomic sequent in which the left and right sequences contain a common element.

Observation: Any instantiation of $Ax_{G'}$ is a valid sequent. □

Definition: The Gentzen system G′ is the proof system consisting of

(a) the axiom schema $Ax_{G'}$, and
(b) the ten proof rules Left-$\wedge$, Right-$\wedge$, Left-$\vee$, Right-$\vee$, Left-$\rightarrow$, Right-$\rightarrow$, Left-$\neg$, Right-$\neg$, Left-$\leftrightarrow$, and Right-$\leftrightarrow$.

The ten proof rules of the Gentzen system G′ in basic sequence format are given below. In each case, the α_i's are to be bound to wff's, and the Γ_i's are to be bound to sequences of wff's.

$\dfrac{(\langle \Gamma_1, \alpha_1, \alpha_2, \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}{(\langle \Gamma_1, (\alpha_1 \wedge \alpha_2), \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}$	Left-$\wedge$
$\dfrac{(\langle \Gamma_1 \rangle, \langle \Gamma_2, \alpha_1, \Gamma_3 \rangle) \quad (\langle \Gamma_1 \rangle, \langle \Gamma_2, \alpha_2, \Gamma_3 \rangle)}{(\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_1 \wedge \alpha_2), \Gamma_3 \rangle)}$	Right-$\wedge$
$\dfrac{(\langle \Gamma_1, \alpha_1, \Gamma_2 \rangle, \langle \Gamma_3 \rangle) \quad (\langle \Gamma_1, \alpha_2, \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}{(\langle \Gamma_1, (\alpha_1 \vee \alpha_2), \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}$	Left-$\vee$
$\dfrac{(\langle \Gamma_1 \rangle, \langle \Gamma_2, \alpha_1, \alpha_2, \Gamma_3 \rangle)}{(\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_1 \vee \alpha_2), \Gamma_3 \rangle)}$	Right-$\vee$
$\dfrac{(\langle \Gamma_1, \Gamma_2 \rangle, \langle \alpha_1, \Gamma_3 \rangle) \quad (\langle \alpha_2, \Gamma_1, \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}{(\langle \Gamma_1, (\alpha_1 \rightarrow \alpha_2), \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}$	Left-$\rightarrow$
$\dfrac{(\langle \alpha_1, \Gamma_1 \rangle, \langle \alpha_2, \Gamma_2, \Gamma_3 \rangle)}{(\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_1 \rightarrow \alpha_2), \Gamma_3 \rangle)}$	Right-$\rightarrow$
$\dfrac{(\langle \Gamma_1, \Gamma_2 \rangle, \langle \alpha, \Gamma_3 \rangle)}{(\langle \Gamma_1, \neg\alpha, \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}$	Left-$\neg$
$\dfrac{(\langle \alpha, \Gamma_1 \rangle, \langle \Gamma_2, \Gamma_3 \rangle)}{(\langle \Gamma_1 \rangle, \langle \Gamma_2, \neg\alpha, \Gamma_3 \rangle)}$	Right-$\neg$
$\dfrac{(\langle \Gamma_1, (\alpha_1 \rightarrow \alpha_2), (\alpha_2 \rightarrow \alpha_1), \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}{(\langle \Gamma_1, (\alpha_1 \leftrightarrow \alpha_2), \Gamma_2 \rangle, \langle \Gamma_3 \rangle)}$	Left-$\leftrightarrow$
$\dfrac{(\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_1 \rightarrow \alpha_2), \Gamma_3 \rangle,) \quad (\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_2 \rightarrow \alpha_1), \Gamma_3 \rangle)}{(\langle \Gamma_1 \rangle, \langle \Gamma_2, (\alpha_1 \leftrightarrow \alpha_2), \Gamma_3 \rangle)}$	Right-$\leftrightarrow$

The ten proof rules of the Gentzen system G′ in common format are given below. In each case, the α_i's are to be bound to wff's, and the Γ_i's are to be bound to sequences of wff's.

$\dfrac{\Gamma_1, \alpha_1, \alpha_2, \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1, (\alpha_1 \wedge \alpha_2), \Gamma_2 \Rightarrow \Gamma_3}$	Left-$\wedge$
$\dfrac{\Gamma_1 \Rightarrow \Gamma_2, \alpha_1, \Gamma_3 \qquad \Gamma_1 \Rightarrow \Gamma_2, \alpha_2, \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \wedge \alpha_2), \Gamma_3}$	Right-$\wedge$
$\dfrac{\Gamma_1, \alpha_1, \Gamma_2 \Rightarrow \Gamma_3 \qquad \Gamma_1, \alpha_2, \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1, (\alpha_1 \vee \alpha_2), \Gamma_2, \Rightarrow \Gamma_3}$	Left-$\vee$
$\dfrac{\Gamma_1 \Rightarrow \Gamma_2, \alpha_1, \alpha_2, \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \vee \alpha_2), \Gamma_3}$	Right-$\vee$
$\dfrac{\Gamma_1, \Gamma_2 \Rightarrow \alpha_1, \Gamma_3 \qquad \alpha_2, \Gamma_1, \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1, (\alpha_1 \rightarrow \alpha_2), \Gamma_2 \Rightarrow \Gamma_3}$	Left-$\rightarrow$
$\dfrac{\alpha_1, \Gamma_1 \Rightarrow \alpha_2, \Gamma_2, \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \rightarrow \alpha_2), \Gamma_3}$	Right-$\rightarrow$
$\dfrac{\Gamma_1, \Gamma_2 \Rightarrow \alpha, \Gamma_3}{\Gamma_1, \neg\alpha, \Gamma_2 \Rightarrow \Gamma_3}$	Left-$\neg$
$\dfrac{\alpha, \Gamma_1 \Rightarrow \Gamma_2, \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, \neg\alpha, \Gamma_3}$	Right-$\neg$
$\dfrac{\Gamma_1, (\alpha_1 \rightarrow \alpha_2), (\alpha_2 \rightarrow \alpha_1), \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1, (\alpha_1 \leftrightarrow \alpha_2), \Gamma_2 \Rightarrow \Gamma_3}$	Left-$\leftrightarrow$
$\dfrac{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \rightarrow \alpha_2), \Gamma_3 \qquad \Gamma_1 \Rightarrow \Gamma_2, (\alpha_2 \rightarrow \alpha_1), \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \leftrightarrow \alpha_2), \Gamma_3}$	Right-$\leftrightarrow$

At this point, it might appear that we are no better off with G′ than we are with the Hilbert system. We have an infinite number of axioms, and so a infinite number of proofs.

Q: To begin a proof, must we “divine” the correct set of starting axioms?

A: No! With G′, we will construct the proof backwards. That is, we will start with the statement which we wish to prove, and work backwards towards the axioms. There is an algorithm to construct such proofs.

Before developing formally notions of soundness and completeness, we will work out a variety of examples.

Examples:

First of all, we will provide a proof, within G′, that the wff

$$((A_1 \rightarrow A_2) \rightarrow (\neg A_2 \rightarrow \neg A_1))$$

is a tautology.

As with resolution and semantic tableaux, the proof is best represented as a directed graph. A solution is shown below on the next slide, in both notations.

Notice the following:

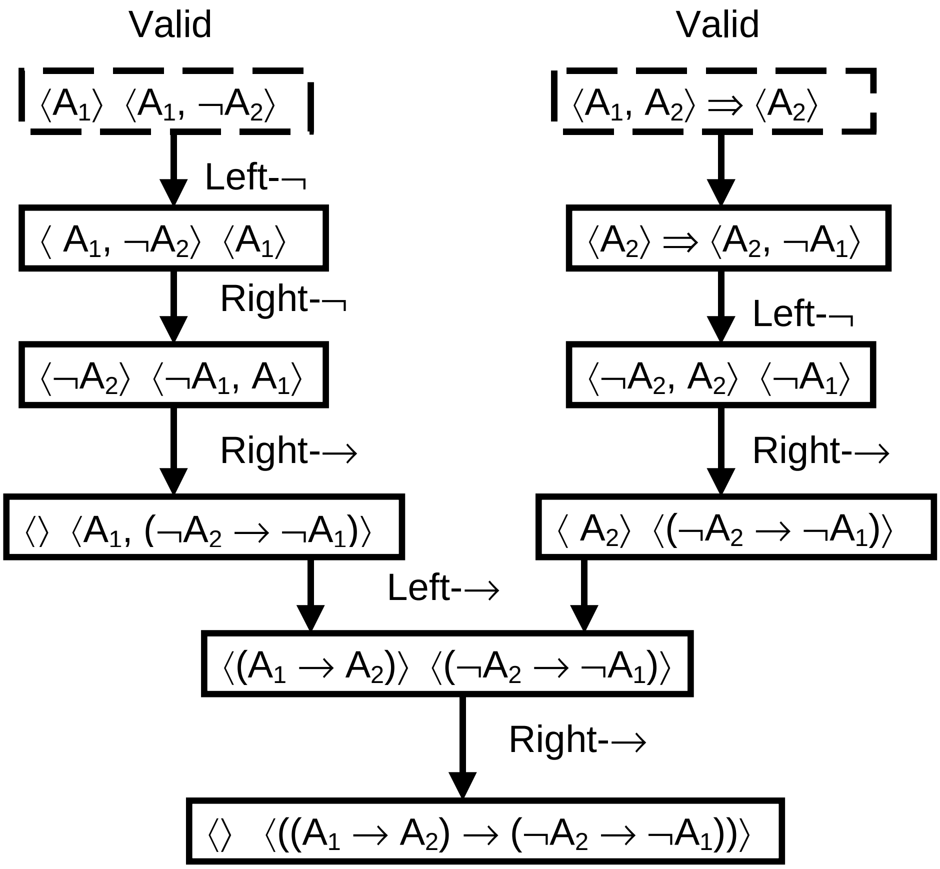
- In constructing this proof, we work backward from the conclusion, to reach valid atomic sequents.

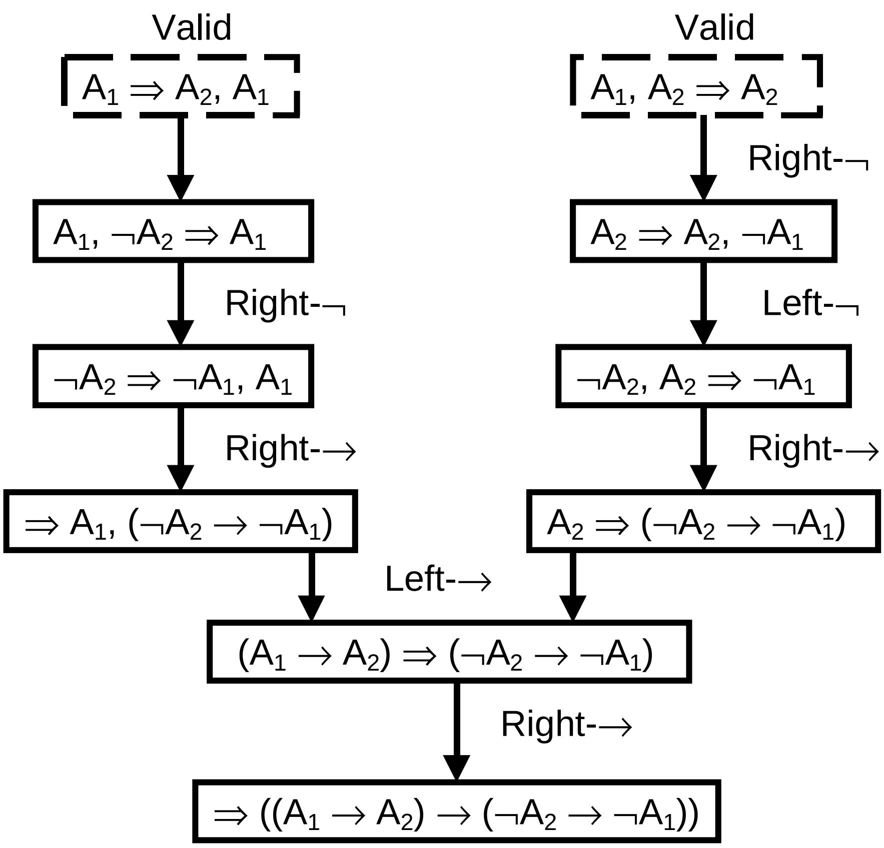
- The formula φ to be validated is represented as the sequent $\Rightarrow \varphi$.

- As we work backwards, each step up the tree produces a "simpler" formula. (This means that the tree cannot grow forever.)

- A path ends whenever an atomic sequent is encountered. Such a sequent cannot be reduced further.

- The original formula is valid iff each path ends with a valid atomic sequent.


Valid
$\langle A_1 \rangle \; \langle A_1, \neg A_2 \rangle$
Left-$\neg$
$\langle A_1, \neg A_2 \rangle \; \langle A_1 \rangle$
Right-$\neg$
$\langle \neg A_2 \rangle \; \langle \neg A_1, A_1 \rangle$
Right-$\rightarrow$
$\langle \rangle \; \langle A_1, (\neg A_2 \rightarrow \neg A_1) \rangle$
Valid
$\langle A_1, A_2 \rangle \Rightarrow \langle A_2 \rangle$
$\langle A_2 \rangle \Rightarrow \langle A_2, \neg A_1 \rangle$
Left-$\neg$
$\langle \neg A_2, A_2 \rangle \; \langle \neg A_1 \rangle$
Right-$\rightarrow$
$\langle A_2 \rangle \; \langle (\neg A_2 \rightarrow \neg A_1) \rangle$
Left-$\rightarrow$
$\langle (A_1 \rightarrow A_2) \rangle \; \langle (\neg A_2 \rightarrow \neg A_1) \rangle$
Right-$\rightarrow$
$\langle \rangle \; \langle ((A_1 \rightarrow A_2) \rightarrow (\neg A_2 \rightarrow \neg A_1)) \rangle$

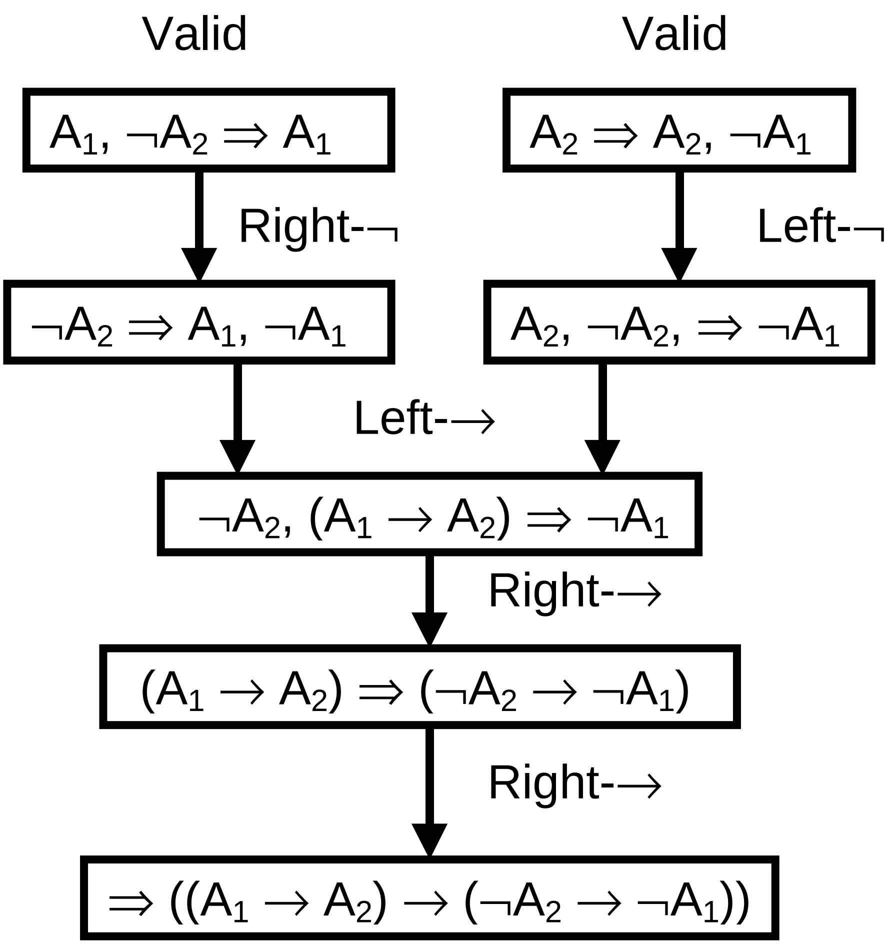

Valid
$A_1 \Rightarrow A_2, A_1$
$A_1, \neg A_2 \Rightarrow A_1$
Right-$\neg$
$\neg A_2 \Rightarrow \neg A_1, A_1$
Right-$\rightarrow$
$\Rightarrow A_1, (\neg A_2 \rightarrow \neg A_1)$
Valid
$A_1, A_2 \Rightarrow A_2$
Right-$\neg$
$A_2 \Rightarrow A_2, \neg A_1$
Left-$\neg$
$\neg A_2, A_2 \Rightarrow \neg A_1$
Right-$\rightarrow$
$A_2 \Rightarrow (\neg A_2 \rightarrow \neg A_1)$
Left-$\rightarrow$
$(A_1 \rightarrow A_2) \Rightarrow (\neg A_2 \rightarrow \neg A_1)$
Right-$\rightarrow$
$\Rightarrow ((A_1 \rightarrow A_2) \rightarrow (\neg A_2 \rightarrow \neg A_1))$

- In establishing validity it is not necessary to expand a node until it becomes an atomic sequent. For validity, it suffices that there be a common proposition name (as a wff; not as a component of one) in each sequence. Thus, the expansions in the previous example could actually have been halted without expanding the boxes which are outlined with dashes.

- It is in fact possible to halt the expansion of a node when there is a common *wff* in each sequence. However, testing for such common formulas at each step involves substantial overhead, and so the decision to do this should be weighed carefully. In these notes, only checking for matching proposition names will be performed.

- This shortcut only applies to establishing validity. For establishing invalidity, a full expansion to atomic sequents must be performed (or other, more complex shortcuts must be employed).

- At some steps, there may be a choice as to which rule to apply. Here is an alternate proof, also making use of the shortcut just described.



- In some sense, this proof is “better” than the first one, because it delays branching into two distinct paths as long as possible.

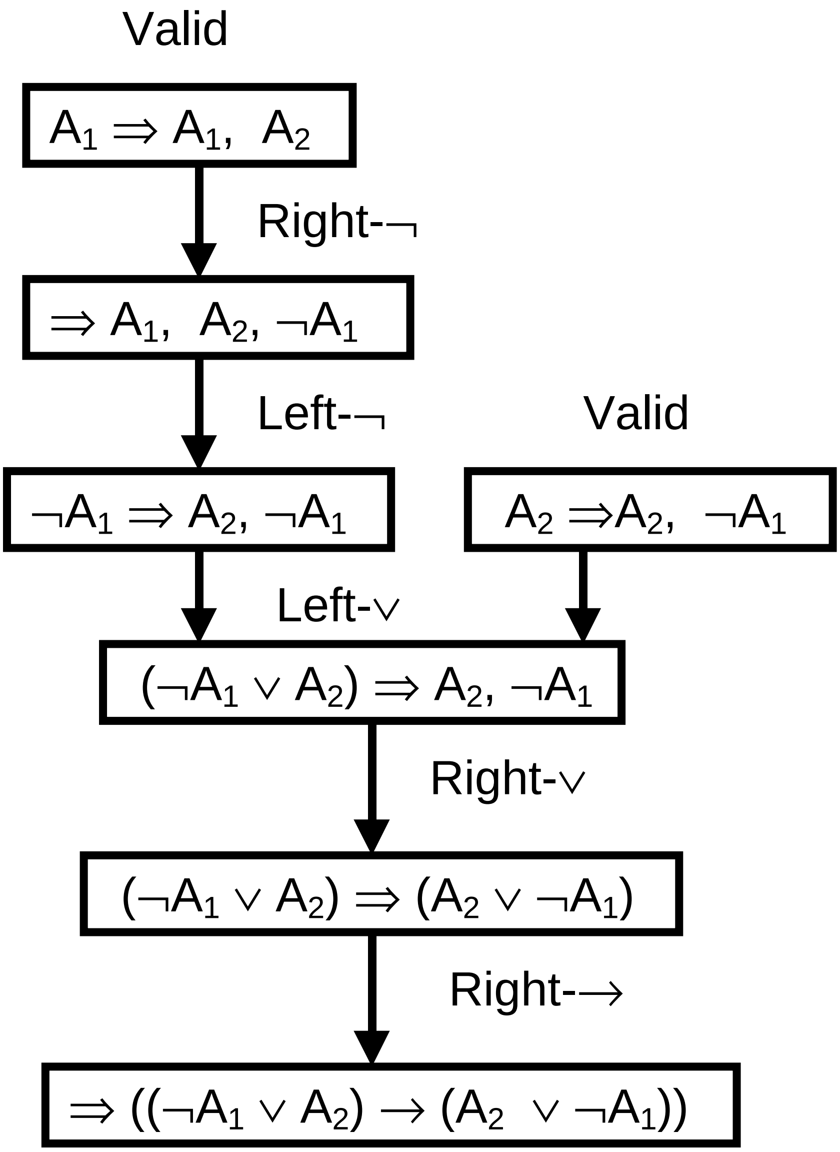
- In most logic books, including the course textbook, proofs are written using a sort of “stacked proof rule” syntax, as illustrated below.

- This syntax tends to become difficult to read as the size of the proof grows, and it makes it all but impossible to show explicitly the proof rules which were used. The explicit tree syntax is therefore preferable.

$$\dfrac{\dfrac{\dfrac{\dfrac{A_1, \neg A_2, \Rightarrow A_1}{\neg A_2, \Rightarrow \neg A_1, A_1}}{\Rightarrow A_1, (\neg A_2 \to \neg A_1)} \qquad \dfrac{\dfrac{A_2 \Rightarrow A_2, \neg A_1}{\neg A_2, A_2 \Rightarrow \neg A_1,}}{A_2 \Rightarrow (\neg A_2 \to \neg A_1)}}{(A_1 \to A_2) \Rightarrow (\neg A_2 \to \neg A_1)}}{\Rightarrow ((A_1 \to A_2) \to (\neg A_2 \to \neg A_1))}$$

Here is a proof of an equivalent formula, with the inner implications replaced by disjunctions.

Notice that the result is the same, but that the rules which are applied are different.

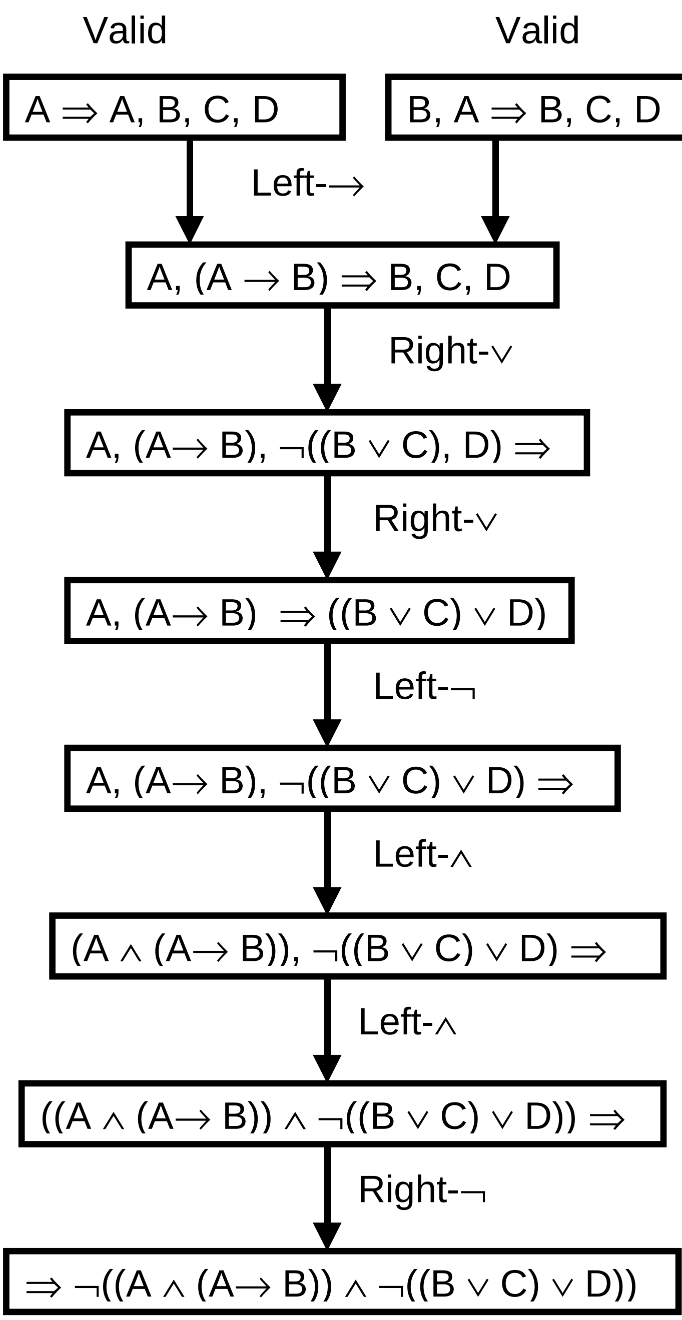


Example: Strictly speaking, a formula of the form

$$\neg(A \wedge (A \rightarrow B) \wedge \neg(B \vee C \vee D)$$

cannot be processed with the existing rules, since only binary conjunction and disjunction is supported. To handle such a formula, a grouping such as the following must first be imposed, either explicitly or implicitly.

$$\neg((A \wedge (A \rightarrow B)) \wedge \neg((B \vee C) \vee D))$$



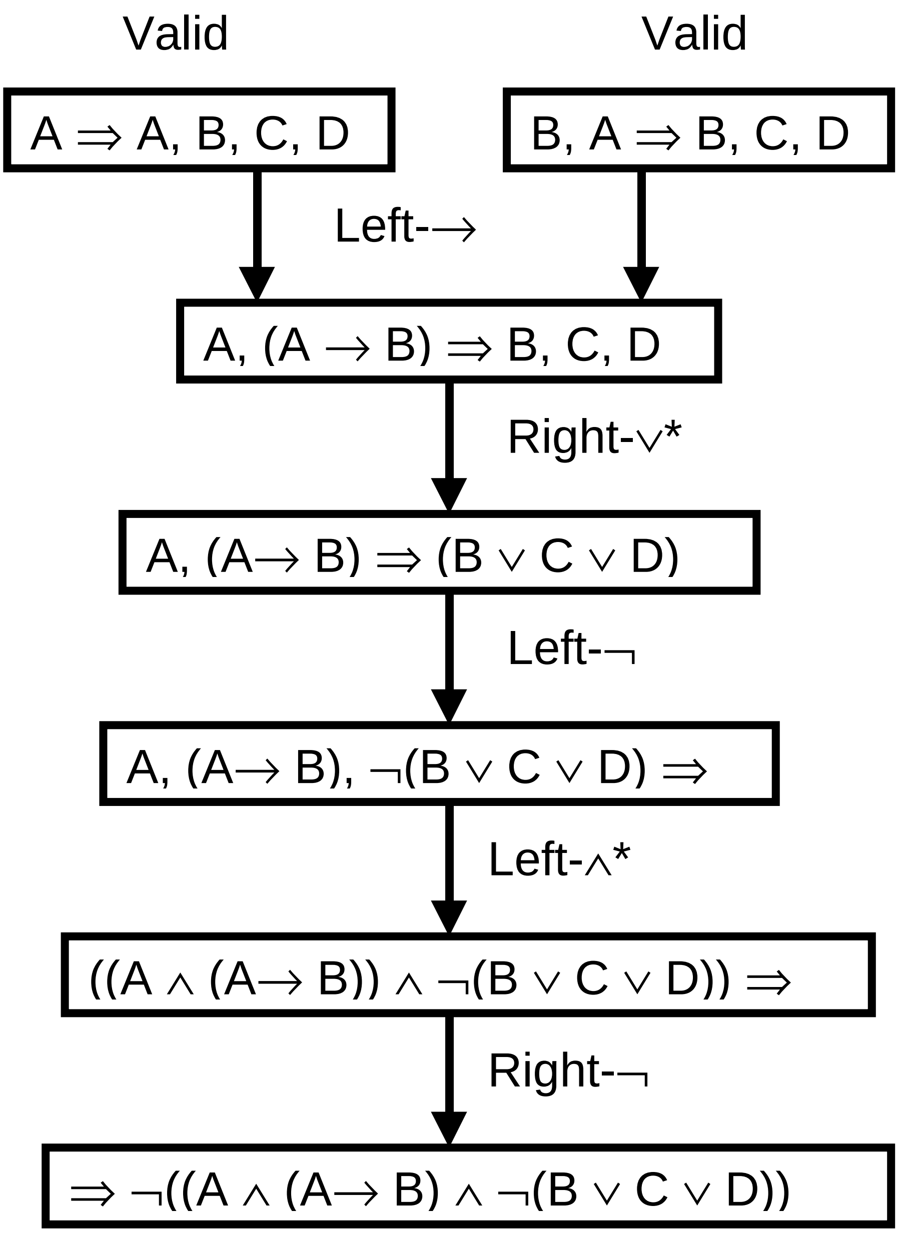
To handle such cases more cleanly, it is useful to introduce some extended rules. Essentially, all that they do is allow sequences of $\wedge$'s and $\vee$'s to be expanded in one step, rather than two at a time.

$\dfrac{\Gamma_1, \alpha_1, \alpha_2, .., \alpha_n, \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1, (\alpha_1 \wedge \alpha_2 \wedge .. \wedge \alpha_n), \Gamma_2 \Rightarrow \Gamma_3}$	Left-$\wedge$*
$\dfrac{\Gamma_1, \alpha_1, \Gamma_2 \Rightarrow \Gamma_3 \quad \Gamma_1, \alpha_2, \Gamma_2 \Rightarrow \Gamma_3 \quad .. \quad \Gamma_1, \alpha_n, \Gamma_2 \Rightarrow \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \wedge \alpha_2 \wedge .. \wedge \alpha_n), \Gamma_3}$	Right-$\wedge$*
$\dfrac{\Gamma_1 \Rightarrow \Gamma_2, \alpha_1, \Gamma_3 \quad \Gamma_1 \Rightarrow \Gamma_2, \alpha_2, \Gamma_3 \quad .. \quad \Gamma_1 \Rightarrow \Gamma_2, \alpha_n, \Gamma_3}{\Gamma_1, (\alpha_1 \vee \alpha_2 \vee .. \vee \alpha_n), \Gamma_2, \Rightarrow \Gamma_3}$	Left-$\vee$*
$\dfrac{\Gamma_1 \Rightarrow \Gamma_2, \alpha_1, \alpha_2, .., \alpha_n, \Gamma_3}{\Gamma_1 \Rightarrow \Gamma_2, (\alpha_1 \vee \alpha_2 \vee .. \vee \alpha_n), \Gamma_3}$	Right-$\vee$*

Here is a simpler proof of

$$\neg(A \wedge (A \rightarrow B) \wedge \neg(B \vee C \vee D))$$

Using these extended rules.



Now we look at the “people going to the party” example. There are two ways to formulate this problem in G′. First of all, things may be set up to refute the following set of clause, as was done in the resolution solution.

$$\psi = \{(J \lor Y), (\neg Y \lor S \lor C), (J \lor S), (\neg C)\}.$$

An alternative is to set things up as a direct implication; that is, to establish that

$$\{(J \lor Y), (\neg Y \lor S \lor C), (J \lor S)\} \models C$$

Both possibilities are expressible as a sequent.

In the first case, the refutation is expressed as

$$(J \lor Y), (\neg Y \lor S \lor C), (J \lor S), (\neg C) \Rightarrow$$

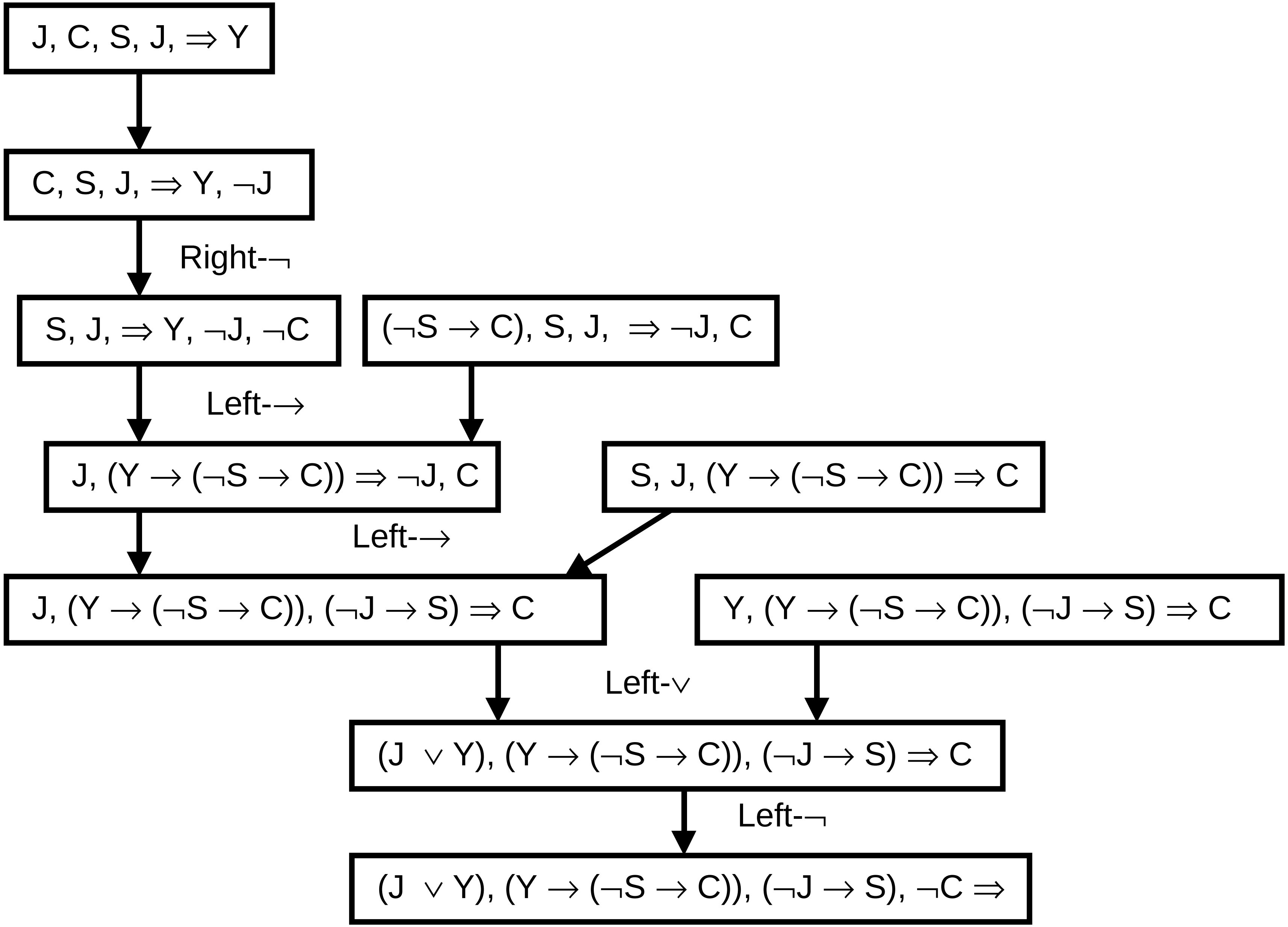
In the second case, the direct deduction is expressible as

$$(J \lor Y), (\neg Y \lor S \lor C), (J \lor S) \Rightarrow C$$

The proof graph on the next slide actually recaptures both cases. Note that the first rule applied (left-$\neg$ to $\neg C$) transforms the input sequent into the above representation of direct deduction.

Notice that an atomic sequent which is not valid is obtained, at which point further expansion is unnecessary, since the input sequent cannot be valid.

Not valid


J, C, S, J, ⇒ Y
C, S, J, ⇒ Y, ¬J
Right-¬
S, J, ⇒ Y, ¬J, ¬C
(¬S → C), S, J, ⇒ ¬J, C
Left-→
J, (Y → (¬S → C)) ⇒ ¬J, C
S, J, (Y → (¬S → C)) ⇒ C
Left-→
J, (Y → (¬S → C)), (¬J → S) ⇒ C
Y, (Y → (¬S → C)), (¬J → S) ⇒ C
Left-∨
(J ∨ Y), (Y → (¬S → C)), (¬J → S) ⇒ C
Left-¬
(J ∨ Y), (Y → (¬S → C)), (¬J → S), ¬C ⇒

On the next slide is shown a proof of the unsatisfiability of the following set of clauses:

$\{A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E), \neg E, \neg C\}$

This is equivalent to showing that

$\{A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E)\} \models E \vee C$

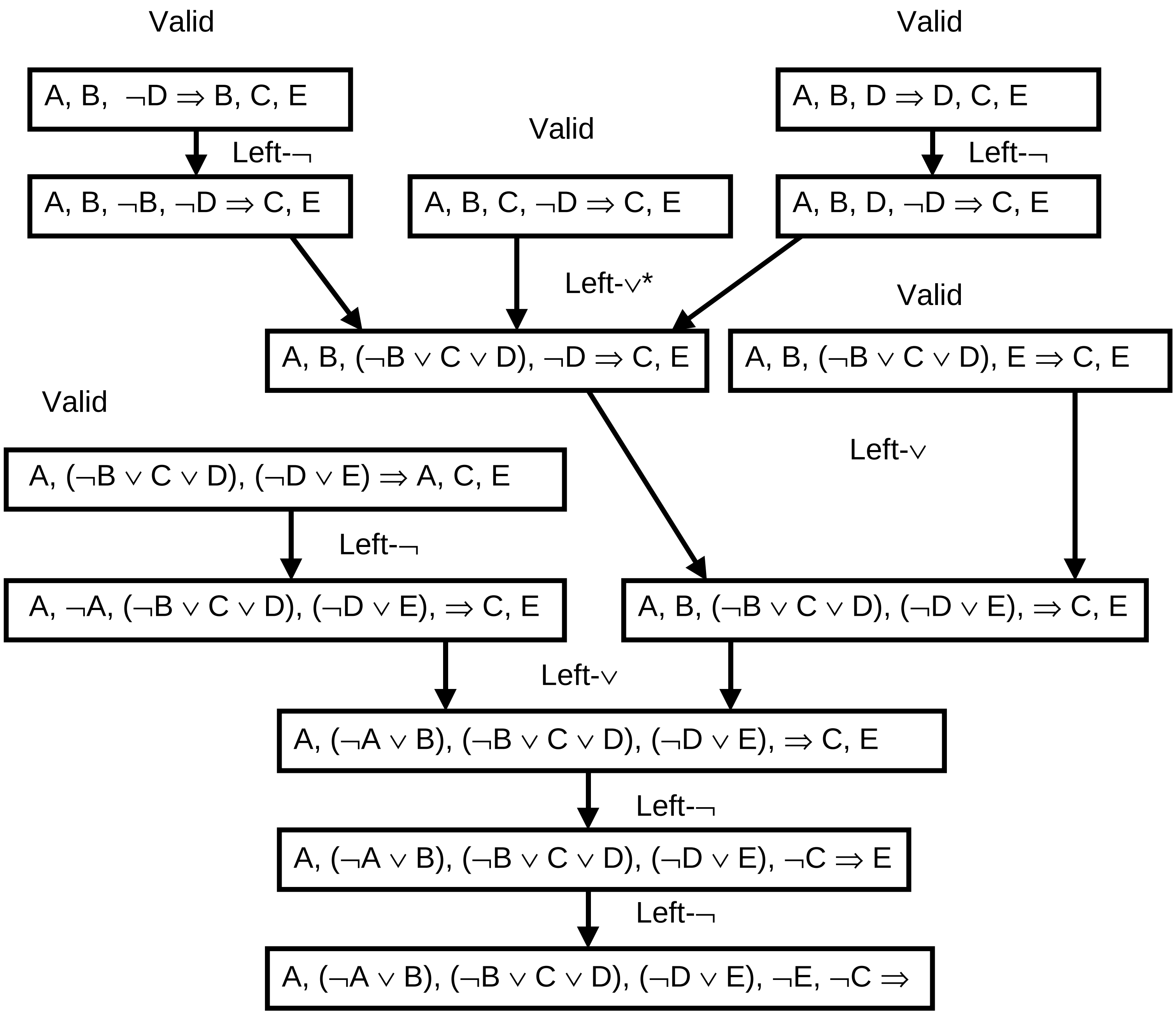
Note that within the sequent system, these two problems have almost the same representation and solution method. The sequent

$A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E) \Rightarrow E, C$

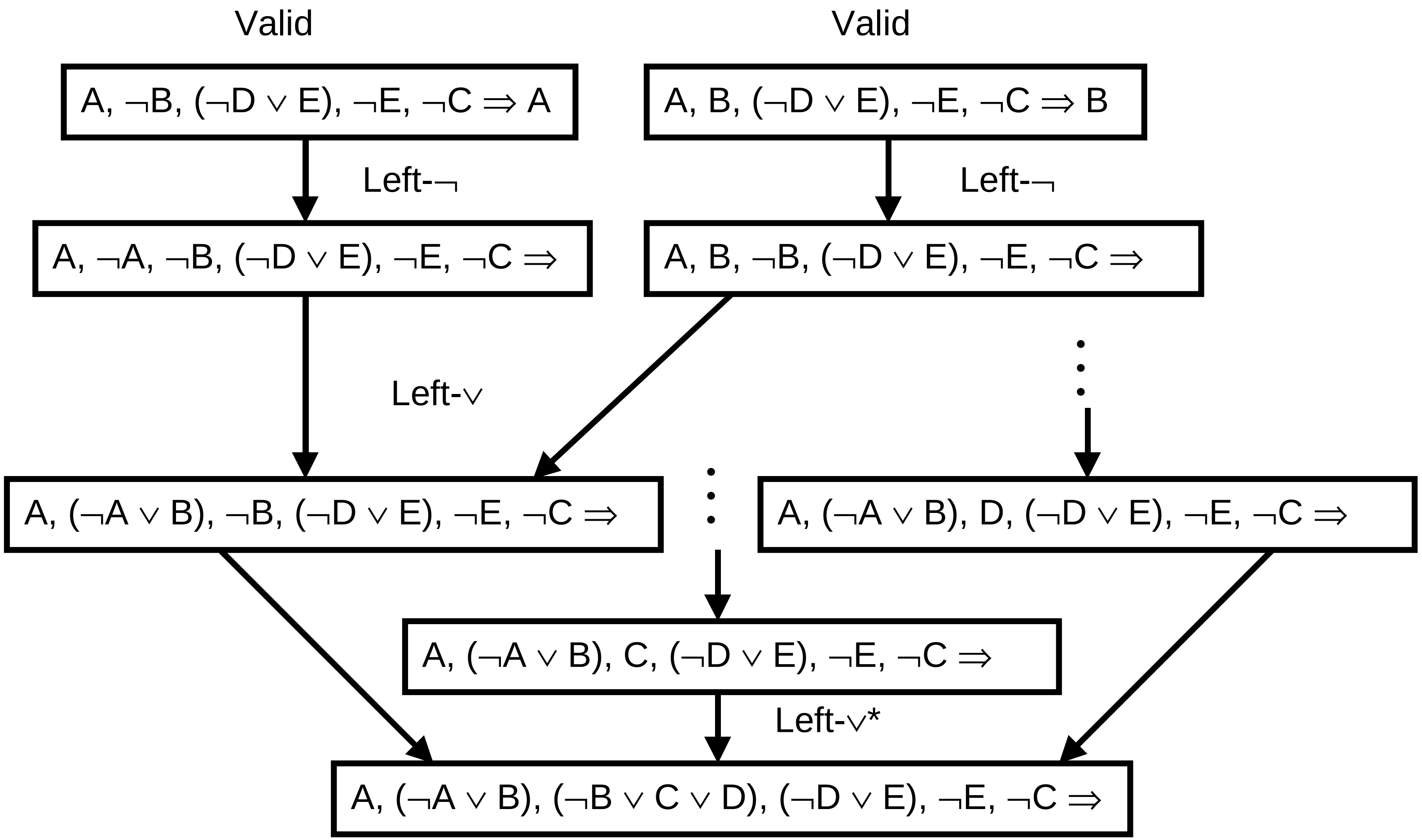
is obtained in a single step from the sequent

$A, (\neg A \vee B), (\neg B \vee C \vee D), (\neg D \vee E) \Rightarrow E \vee C$

by application of the Right-$\vee$ rule, so the proof on the following slide may be modified by replacing the first two steps with this single application of Right-$\vee$.


Valid
A, B, ¬D ⇒ B, C, E
Left-¬
A, B, ¬B, ¬D ⇒ C, E
Valid
A, B, C, ¬D ⇒ C, E
Valid
A, B, D ⇒ D, C, E
Left-¬
A, B, D, ¬D ⇒ C, E
Left-∨*
Valid
A, B, (¬B ∨ C ∨ D), ¬D ⇒ C, E
A, B, (¬B ∨ C ∨ D), E ⇒ C, E
Valid
A, (¬B ∨ C ∨ D), (¬D ∨ E) ⇒ A, C, E
Left-∨
Left-¬
A, ¬A, (¬B ∨ C ∨ D), (¬D ∨ E), ⇒ C, E
A, B, (¬B ∨ C ∨ D), (¬D ∨ E), ⇒ C, E
Left-∨
A, (¬A ∨ B), (¬B ∨ C ∨ D), (¬D ∨ E), ⇒ C, E
Left-¬
A, (¬A ∨ B), (¬B ∨ C ∨ D), (¬D ∨ E), ¬C ⇒ E
Left-¬
A, (¬A ∨ B), (¬B ∨ C ∨ D), (¬D ∨ E), ¬E, ¬C ⇒

Notice that the choice of which element to expand is important.   Here is an example of a poor choice. (There are several long branches left to be expanded, shown by three vertical dots.



Soundness and Completeness:

Definition: Let $\vdash_{G'}$ denote the inference relation for the proof system G′.

The soundness of G′ is an immediate consequence of the nature of the proof rules.

Theorem: G′ is a sound inference mechanism. That is,

$$\vdash_{G'} \Lambda_1 \Rightarrow \Lambda_2 \quad \text{implies} \quad \models \Lambda_1 \Rightarrow \Lambda_2. \ \square$$

Now let us turn to completeness.

Definition: Call a proof rule

$$\frac{\Omega_1 \quad \Omega_2 \quad .. \quad \Omega_n}{\Omega}$$

non-weakening if each of the "inverse" rules

$$\frac{\Omega}{\Omega_i}$$

is a sound proof rule. Otherwise, call the rule *weakening.* Intuitively, a non-weakening rule is one in which the premises are not too strong. There is cannot be information in the premises which is not passed on to the conclusion.

Proposition: Each of the ten proof rules of G′, as well as the four extended rules, are non-weakening.
Proof: This is a straightforward verification. It will be discussed in class, but not written out in these slides. □

Theorem: G′ is a complete inference mechanism. That is,

$$\models \Lambda_1 \Rightarrow \Lambda_2 \quad \text{implies} \quad \vdash_{G'} \Lambda_1 \Rightarrow \Lambda_2.$$

Proof: If a proof tree is found for which each leaf is a valid sequent, then the root must be a consequence of those axioms. On the other hand, if a proof tree is found for which one of the leaves is a non-valid atomic sequent, then that sequent must be a consequence of the root sequent, whence the root cannot be valid. □

Algorithmicity:

Are we done? Not quite. We are computer scientists, and the question of termination of this procedure is central to us.

How do we know that this process will terminate? Perhaps the process of expanding nodes of the tree can run forever. After all, there is an infinite number of axioms. The key to establishing that this cannot happen is to note that the formulas associated with the premises of the proof rules are simpler than those associated with the conclusion.

Definition: The *complexity of a wff* φ, denoted Complexity(φ), is defined according to the following table.

φ	Complexity(φ)
A	0
$(\neg\psi)$	1 + Complexity(ψ)
$(\psi_1 \vee \psi_2)$	1 + Complexity(ψ_1) + Complexity(ψ_2)
$(\psi_1 \wedge \psi_2)$	1 + Complexity(ψ_1) + Complexity(ψ_2)
$(\psi_1 \rightarrow \psi_2)$	1 + Complexity(ψ_1) + Complexity(ψ_2)
$(\psi_1 \leftrightarrow \psi_2)$	1 + Complexity($(\psi_1 \rightarrow \psi_2)$) + Complexity($(\psi_2 \rightarrow \psi_1)$)

The *complexity* of a sequent is the sum of the complexities of all of its members (in both sequences).

Note that the complexity of an atomic sequent is zero.

Definition: Call a proof rule *simplifying* if for any instantiation, the complexity of the conclusion is strictly greater than the complexity of each premise.

Proposition: The system G′ is simplifying.
Proof: This is easily verified from a case-by-case examination of the proof rules. □

Theorem: G′ admits an algorithmic deductive process.
Proof: Because the proof rules are simplyfing, the expansions cannot continue indefinitely. □

(Remember that proof construction proceeds from conclusion to hypotheses.)

The sequent system of the textbook:

The textbook presents a different sequent system than the one developed in these notes.

- Some of the proof rules of the system in the text are weakening. Examples include the rules L$\wedge$ (converted to our notation):

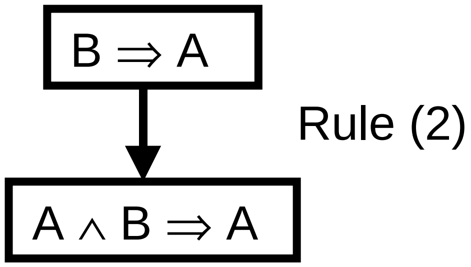
$$\frac{\Gamma_1, \alpha_1 \Rightarrow \Gamma_2}{\Gamma_1, (\alpha_1 \wedge \alpha_2) \Rightarrow \Gamma_2} \qquad \frac{\Gamma_1, \alpha_2 \Rightarrow \Gamma_2}{\Gamma_1, (\alpha_1 \wedge \alpha_2) \Rightarrow \Gamma_2}$$

As well as the rules LT and RT:

$$\frac{\Gamma_1 \Rightarrow \Gamma_2}{\Gamma_1 \Rightarrow \alpha, \Gamma_2} \qquad \frac{\Gamma_1 \Rightarrow \Gamma_2}{\Gamma_1, \alpha \Rightarrow \Gamma_2}$$

Because these rules are weakening, it is not guaranteed that leaf nodes are consequences of the root. Therefore, invalidity cannot be established except by exhaustive search.

Example: Below is shown a deduction of
$A \wedge B \Rightarrow A$



from $B \Rightarrow A$.

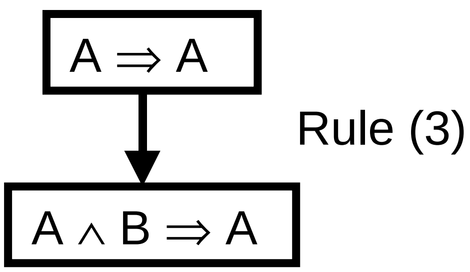
While the deduction is perfectly correct, it is not the case that the inverse deduction is correct. In other words, while

$$B \Rightarrow A \models A \wedge B \Rightarrow A$$

certainly holds, it is not the case that

$$A \wedge B \Rightarrow A \models B \Rightarrow A$$

does. Thus, it it is erroneous to conclude that $A \wedge B \Rightarrow A$ is not valid. Instead, it is necessary to backtrack and try other rules. The following does the trick.



However, this mandates the need for a much more complex search algorithm, capable of backtracking.

- The system in the textbook only uses the simpler axiom schema:

$$\alpha \Rightarrow \alpha$$

- While this schema is adequate, it does require some supporting effort, including replication and rearrangement axioms (RC, LC, RR, and LR).

$$\frac{\Gamma_1 \Rightarrow \alpha, \alpha, \Gamma_2}{\Gamma_1 \Rightarrow \alpha, \Gamma_2} \qquad \frac{\Gamma_1, \alpha, \alpha \Rightarrow \Gamma_2}{\Gamma_1, \alpha \Rightarrow \Gamma_2}$$

$$\frac{\Gamma_1 \Rightarrow \alpha_1, \alpha_2, \Gamma_2}{\Gamma_1 \Rightarrow \alpha_2, \alpha_1, \Gamma_2} \qquad \frac{\Gamma_1, \alpha_1, \alpha_2 \Rightarrow \Gamma_2}{\Gamma_1, \alpha_2, \alpha_1 \Rightarrow \Gamma_2}$$

Such axioms add to the search complexity. While they are non-weakening, they are not simplifying, and so may be applied and re-applied endlessly, unless a separate mechanism for avoiding this is added.

Conclusions:

- The system of the text is sound.

- It is probably complete, but the author has not proven this.

- It includes rules which are weakening and/or non-simplifying, so at best it is unclear how to devise an algorithm to use this system. At the very least, such an algorithm would require backtracking and intelligent search.

The cut rule:

No discussion of sequent calculi is complete without mentioning the cut rule. It is essentially a rule for chaining implications.

$$\frac{\Gamma_1 \Rightarrow \alpha, \Gamma_2 \qquad \Gamma_3, \alpha \Rightarrow \Gamma_4}{\Gamma_1, \Gamma_2 \Rightarrow \Gamma_3, \Gamma_4}$$

It is easy to see that this rule is valid. In some cases, it can simplify some proofs substantially.

Further information:

The most extensive reference on sequent systems, from a computer science perspective, is:

Gallier, Jean, H., *Logic for Computer Science: Foundations of Automatic Theorem Proving,* Harper and Row, 1986.

The sequent system described in these notes is also discussed there.