Tableau Proofs

Robert Keller

January 2009

Meaning

- “Tableau” (French for “table”)
- “Tableaux” plural
- As used here, more of a tree than a table. Several variations exist.
- Variant terminology:
  - Analytic tableau
  - Semantic tableau

Synatctic or Semantic

- This is a syntactic proof method, with obvious connections to semantics.

Proof by Refutation

- Recall:

  A is valid iff ¬A is unsatisfiable.

- A refutation proof proves unsatisfiability.
- Thus it indirectly proves validity of the negation (assuming RAA, therefore this is assumes classical, rather than constructive logic). However, constructive variants are possible.

Algorithmic

- For propositional logic, this method is algorithmic: guaranteed to find a proof.

- (For predicate logic, only semi-algorithmic).

Method

- Negate the formula to be proved (for which validity is to be checked).
- Then undertake the tree (tableau) construction to be described. Each step breaks down a working set of formulas into a possibly-larger set of **simpler** formulas.
- Note: This method can be used to **check** validity, even if some **other** proof method, such as Natural Deduction, is used to **present** a proof.

Why Not Used Universally?

- Although the tableau proof method is very effective, proofs are not usually **presented** this way informally. It is not as “natural”.

- Also, our tableau proofs are not constructive, in that they have the LEM built-in, in effect.

Tableau Construction

- Start with the formula to be checked for satisfiability as the root.
- Successively “replace” formulas by new ones derived from sub-formulas.
- In some cases, the tree branches into two.
- In all cases, the construction will eventually stop.
- At the stopping point, satisfiability is readily determined “by inspection”.

Formulas are classified in one of four ways:

- α (Conjunctive): The form is one of:

  $A \wedge B$

  $\neg(A \vee B)$

  $\neg(A \to B)$

- β (Disjunctive): The form is one of:

  $A \vee B$

  $\neg(A \wedge B)$

  $A \to B$

- $\neg\neg A$: Replace with A. This is sometimes regarded as an α.

- Other: Either a proposition symbol or its negation. Leave as is.

α: A ∧ B, ¬(A ∨ B), ¬(A → B)

- The formula is checked off (removed from further consideration).
- The derived formulas are "**stacked**" in its place on the tree, as follows:

A ∧ B	stack A and B
¬(A ∨ B)	stack ¬A and ¬B
¬(A → B)	stack A and ¬B

β: A ∨ B, A → B, ¬(A ∧ B)

- The formula is checked off (removed from further consideration).
- The sub-formulas are "**split**" creating a branch of the tree for each, as follows:

A ∨ B	split to A and B
¬(A ∧ B)	split to ¬A and ¬B
A → B	split to ¬A and B

Completion

- A tableau construction is **complete** if no further rule applications are possible.

Examples

- Check validity of: $p \rightarrow (q \rightarrow p)$
- Negate:

$$\neg(p \rightarrow (q \rightarrow p))$$

Tableau

1. $\neg(p \rightarrow (q \rightarrow p))$

Tableau

1. $\neg(p \rightarrow (q \rightarrow p))$ ✓ (stack)
2. p
3. $\neg(q \rightarrow p)$

Tableau

1. $\neg(p \rightarrow (q \rightarrow p))$ ✓ (stack)
2. p
3. $\neg(q \rightarrow p)$ ✓ (stack)
4. q
5. $\neg p$

At this point, the tableau is complete.

Closed and Open Tableau

- A **path** from root to leaf in a tableau is **closed** if there is **a formula and its negation** appearing on the path.
- We show closed paths by placing an X at the leaf.
- A **tableau** is closed if all paths from root to leaves are **closed**.
- A tableau is **open** iff it is not closed.

Main Result

- The root formula is unsatisfiable iff the complete tableau is closed.

Tableau

1. $\neg(p \rightarrow (q \rightarrow p))$ ✓ (stack)
2. p
3. $\neg(q \rightarrow p)$ ✓ (stack)
4. q
5. $\neg p$

X(4, 5) closed by p, $\neg p$

At this point, the tableau is complete.
There is only one path from root to leaf, and it is closed.
Therefore, the root formula is unsatisfiable.
(The original formula is thus valid.)

Tableau Example

1. $\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))$

1. $\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))$ ✓

2. $(p \rightarrow q)$

3. $\neg(\neg q \rightarrow \neg p)$

1. $\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))$ ✓

2. $(p \rightarrow q)$

3. $\neg(\neg q \rightarrow \neg p)$ ✓

4. $\neg q$

5. $\neg\neg p$

1. $\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))$ ✓

2. $(p \rightarrow q)$

3. $\neg(\neg q \rightarrow \neg p)$ ✓

4. $\neg q$

5. $\neg\neg p$ ✓

6. p

1. $\neg((p \to q) \to (\neg q \to \neg p))$ ✓

2. $(p \to q)$ ✓

3. $\neg(\neg q \to \neg p)$ ✓

4. $\neg q$

5. $\neg\neg p$ ✓

6. p

Split from line 2.

7. $\neg p$

X(6, 7)

8. q

X(4, 8)

Satisfiable Root

- If the root formula is **satisfiable**, there is at least **one open path** in the complete tableau.

- That **path** can determine an **assignment** that satisfies the formula:
  - If a proposition symbol occurs **bare**, then assign **true**.
  - If a proposition symbol occurs **negated**, then assign **false**.
  - If a proposition symbol does not occur, then its value does not matter in the assignment.

Example

1. $\neg((p \rightarrow q) \wedge (q \rightarrow p))$ ✓

2. $\neg(p \rightarrow q)$ ✓	3. $\neg(q \rightarrow p)$ ✓
4. p	6. q
5. $\neg q$	7. $\neg p$

The tableau is complete.
Both paths are open.

One satisfying assignment is $\nu(p) = T$, $\nu(q) = F$.
Another is $\nu(p) = F$, $\nu(q) = T$.

Why the Tableau Works

- Each move adds to a path, but preserves satisfiabilty along *some* path Γ.

- α **move**, such as $A \wedge B$:
  $\Gamma \cup \{A \wedge B\}$ is satisfiable iff $\Gamma \cup \{A, B\}$ is satisfiable.

- β **move**, such as $A \vee B$:
  $\Gamma \cup \{A \vee B\}$ is satisfiable iff $\Gamma \cup \{A\}$ is satisfiable **or** $\Gamma \cup \{B\}$ is satisfiable.

- $\Gamma \cup \{\neg\neg A\}$ is satisfiable iff $\Gamma \cup \{A\}$ is.

- $\Gamma \cup \{p, \neg p\}$ is not satisfiable (path closes).

$\Gamma \cup \{A \wedge B\}$ is satisfiable
iff $\Gamma \cup \{A, B\}$ is satisfiable

The following are equivalent:

- ν satisfies $\Gamma \cup \{A \wedge B\}$.
- ν satisfies Γ and ν satisfies $A \wedge B$.
- ν satisfies Γ and ν satisfies A and ν satisfies B.
- ν satisfies $\Gamma \cup \{A, B\}$.

$\Gamma \cup \{A \vee B\}$ is satisfiable
iff $\Gamma \cup \{A\}$ is satisfiable
or $\Gamma \cup \{B\}$ is satisfiable

The following are equivalent:

- ν satisfies $\Gamma \cup \{A \vee B\}$.
- ν satisfies Γ and ν satisfies $A \vee B$.
- ν satisfies Γ and (ν satisfies A or ν satisfies B).
- ν satisfies $\Gamma \cup \{A\}$ or ν satisfies $\Gamma \cup \{B\}$.

Why the Tableau Works

$A \wedge B$ satisfiable

A satisfiable
B satisfiable

$A \vee B$ satisfiable

A satisfiable **or** B satisfiable

$A \wedge B$ unsatisfiable

A unsatisfiable
B or unsatisfiable

$A \vee B$ unsatisfiable

A unsatisfiable **and** B unsatisfiable

Our Tableau vs. Ben Ari's

- Instead of using the tree formation, Ben-Ari carries **each path** of un-checked formulas as a **set**.

- (These are called "Block tableaux" in Smullyan's terminology.)

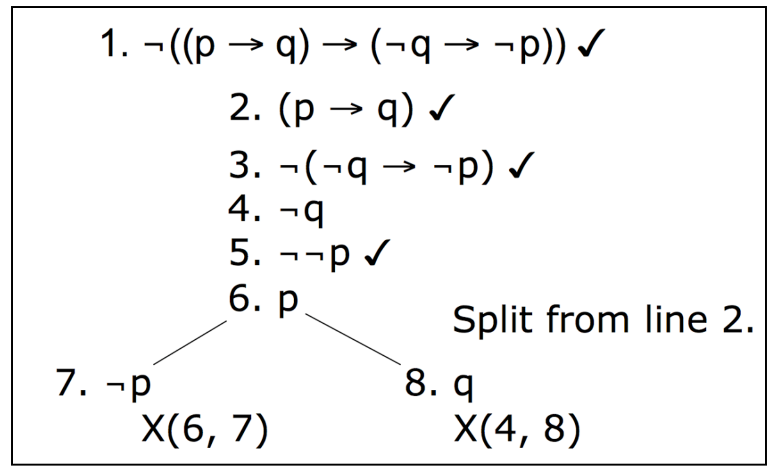
- When an α rule is used, the formula is replaced **within the set** with the two constituent formulas.

- When a β rule is used, the **set splits**. The formula is replaced with one of the two constituent formulas in each set.

Ben-Ari version vs. Ours

- $\{\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))\}$
- $\{p \rightarrow q, \neg(\neg q \rightarrow \neg p)\}$
- $\{p \rightarrow q, \neg q, \neg\neg p\}$
- $\{p \rightarrow q, \neg q, p\}$
  (split)
- $\{\neg p, \neg q, p\}$, $\{q, \neg q, p\}$

  X X



Example: Ben-Ari, p. 31

$(p \vee q) \wedge (\neg p \wedge \neg q)$ set of 1

$\downarrow$ α

$p \vee q, \neg p \wedge \neg q$ set of 2

$\downarrow$ α

$p \vee q, \neg p, \neg q$ set of 3

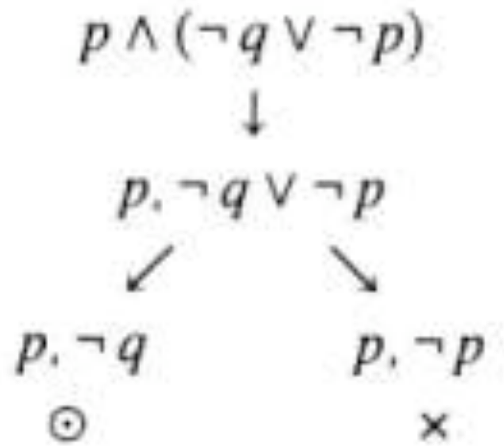
$\swarrow \quad \searrow$ β

$p, \neg p, \neg q$ $\quad$ $q, \neg p, \neg q$ 2 sets of 3

$\times$ $\quad$ $\times$

closed $\quad$ closed

Example: Ben-Ari, p. 31



α

β

open

closed

Ben-Ari's Tableau Rules

α	α_1	α_2
$\neg\neg A_1$	A_1	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \uparrow A_2)$	A_1	A_2
$A_1 \downarrow A_2$	$\neg A_1$	$\neg A_2$
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$
$\neg(A_1 \oplus A_2)$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

β	β_1	β_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \vee B_2$	B_1	B_2
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \uparrow B_2$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \downarrow B_2)$	B_1	B_2
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$
$B_1 \oplus B_2$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

Why does the tableau construction terminate?

- Each move deconstructs a formula on a path.
- We can create a **metric** W for a set that:

  has a positive integer value,

  thus cannot decrease indefinitely

  always decrease with each move

W = 3*#binary operators + # negations

Metric Example

- $W(\{\neg((p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p))\}) =$
  $3*3+3*1 = 12$

- $W(\{ (p \rightarrow q), \neg(\neg q \rightarrow \neg p)\} =$

  $2*3+3*1 = 9$

Ben-Ari, p 33

Proof: Let $\mathcal{T}$ be the tableau for formula A at any stage of its construction and let us assume for now that $\leftrightarrow$ and $\oplus$ do not occur in the formula A. For any leaf $l \in \mathcal{T}$, let $b(l)$ be the number of binary operators in formulas in $U(l)$ and let $n(l)$ be the number of negations in $U(l)$. Define

$$W(l) = 3b(l) + n(l).$$

We claim that any step of the construction creates a new node l' or nodes l', l'' such that $W(l) > W(l')$ and $W(l) > W(l'')$. For example, if we apply the α rule to $\neg(A_1 \vee A_2)$ to obtain $\neg A_1$ and $\neg A_2$, then

$$W(l) = k + 3 \cdot 1 + 1 \; > \; k + 3 \cdot 0 + 2 = W(l'),$$

where k is the sum of the number of operators in A_1 and A_2. Obviously, $W(l) \geq 0$, so no branch of $\mathcal{T}$ can be extended indefinitely. We leave it to the reader to check the correctness of the claim for the other rules and to modify the definition of $W(l)$ in the case that A contains $\leftrightarrow$ or $\oplus$. ∎

Gentzen System G

- These are rules that turn out to be a certain sense of “**dual”** of the tableau rules.
- They are expressed with **sets** of derived formulas carried in the antecedents and consequents of rules.
- CAUTION: Note that these sets represent **disjunctions** rather than conjunctions!!!

Examples for “System G”

$$\frac{\vdash U_1 \cup \{B_1\} \qquad \vdash U_2 \cup \{B_2\}}{\vdash U_1 \cup U_2 \cup \{B_1 \wedge B_2\}}$$

(representing disjunctions)

$$\frac{\vdash U_1 \cup \{A_1, A_2\}}{\vdash U_1 \cup \{A_1 \vee A_2\}}$$

All Rules for G

$$\frac{\vdash U_1 \cup \{\alpha_1, \alpha_2\}}{\vdash U_1 \cup \{\alpha\}} \qquad \frac{\vdash U_1 \cup \{\beta_1\} \qquad \vdash U_2 \cup \{\beta_2\}}{\vdash U_1 \cup U_2 \cup \{\beta\}},$$

where the classification into α- and β-formulas is the dual of the classification for semantic tableaux.

α	α_1	α_2
A	$\neg\neg A$	
$\neg(A_1 \wedge A_2)$	$\neg A_1$	$\neg A_2$
$A_1 \vee A_2$	A_1	A_2
$A_1 \rightarrow A_2$	$\neg A_1$	A_2
$A_1 \uparrow A_2$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \downarrow A_2)$	A_1	A_2
$\neg(A_1 \leftrightarrow A_2)$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$
$A_1 \oplus A_2$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$

β	β_1	β_2
$B_1 \wedge B_2$	B_1	B_2
$\neg(B_1 \vee B_2)$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \rightarrow B_2)$	B_1	$\neg B_2$
$\neg(B_1 \uparrow B_2)$	B_1	B_2
$B_1 \downarrow B_2$	$\neg B_1$	$\neg B_2$
$B_1 \leftrightarrow B_2$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$
$\neg(B_1 \oplus B_2)$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$

Axioms for System G

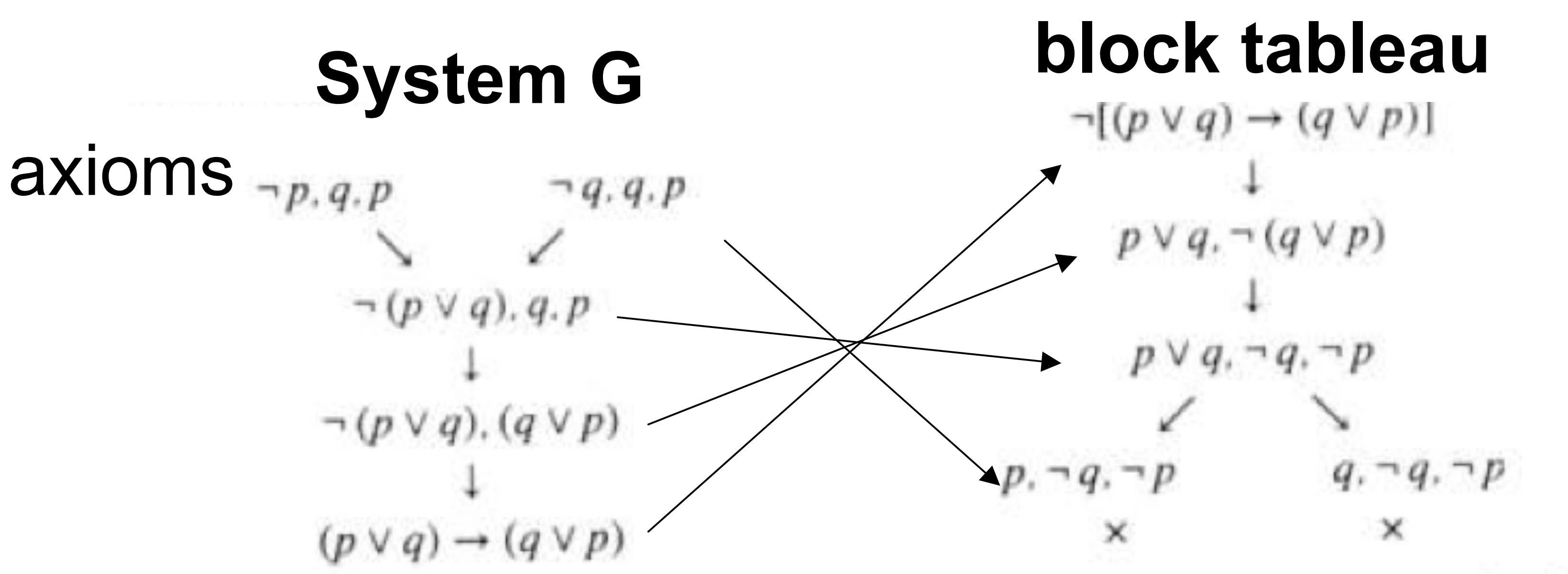
- An **axiom** in G is *any* set that includes a proposition symbol and its negation.
- Proof example:

Example 3.3 $\vdash (p \vee q) \rightarrow (q \vee p)$. The proof is:

1.	$\vdash \neg p, q, p$	Axiom
2.	$\vdash \neg q, q, p$	Axiom
3.	$\vdash \neg (p \vee q), q, p$	$\beta \vee$, 1, 2
4.	$\vdash \neg (p \vee q), (q \vee p)$	$\alpha \vee$, 3
5.	$\vdash (p \vee q) \rightarrow (q \vee p)$	$\alpha \rightarrow$, 4

Dual of the Tableau

- As noted in Ben-Ari, system G is the "**dual**" (?) of the tableau system, in that:
- Negating each formula in the proof and **inverting** the system G proof tree is equivalent to the tableau tree.



“Dual”

- The preceding usage of Ben-Ari does not seem to be the standard definition of “dual”.
- The standard dual of a formula is the formula obtained by interchanging $\wedge$ and $\vee$, and by interchanging $\bot$ and $\top$.
- For $\rightarrow$, we would need to interpret $A \rightarrow B$ as $\neg A \vee B$, giving $\neg A \wedge B$, i.e. $\neg(B \rightarrow A)$ as the dual.
- The standard purpose of introducing duals is intellectual economy: the dual of any equivalence is also an equivalence, the negation of the dual of a tautology is a tautology, etc.

From “The Dictionary of Philosophy”, by Dagobert D. Runes

The *dual of a formula* C of the propositional calculus is obtained by interchanging conjunction and disjunction throughout the formula, i.e., by replacing AB everywhere by A v B, and A v B by AB. Thus, e.g., the dual of the formula $\sim[pq \vee \sim r]$ is the formula $\sim[[p \vee q] \sim r]$. In forming the dual of a formula which is expressed with the aid of the defined contives, $|$, $\supset$, $\equiv$, $+$, it is convenient to remember that the effect of interchanging conjunction and (inclusive) disjunction is to replace $A|B$ by $\sim A \sim B$, to replace $A \supset B$ by $\sim A\, B$, and to interchange $\equiv$ and $+$.

It can be shown that the following *principles of duality* hold in the propositional calculus (where A^* and B^* denote the duals of the formulas A and B respectively): (1) if A is a theorem, then $\sim A^*$ is a theorem; (2) if $A \supset B$ is a theorem, then $B^* \supset A^*$ is a theorem; (3) if $A \equiv B$ is a theorem, then $A^* \equiv B^*$ is a theorem.

Tableau to System G

- A System G proof can be constructed by first constructing a tableau proof (easier to remember than G, perhaps), then transcribing it.
- Replace each formula with its negation.
- Turn the tree upside-down.

G Rules vs. Tableau Rules

System G

$$\frac{\vdash U_1 \cup \{\alpha_1, \alpha_2\}}{\vdash U_1 \cup \{\alpha\}}$$

$$\frac{\vdash U_1 \cup \{\beta_1\} \qquad \vdash U_2 \cup \{\beta_2\}}{\vdash U_1 \cup U_2 \cup \{\beta\}}$$

α	α_1	α_2
A	$\neg\neg A$	
$\neg(A_1 \wedge A_2)$	$\neg A_1$	$\neg A_2$
$A_1 \vee A_2$	A_1	A_2
$A_1 \rightarrow A_2$	$\neg A_1$	A_2
$A_1 \uparrow A_2$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \downarrow A_2)$	A_1	A_2
$\neg(A_1 \leftrightarrow A_2)$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$
$A_1 \oplus A_2$	$\neg(A_1 \rightarrow A_2)$	$\neg(A_2 \rightarrow A_1)$

β	β_1	β_2
$B_1 \wedge B_2$	B_1	B_2
$\neg(B_1 \vee B_2)$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \rightarrow B_2)$	B_1	$\neg B_2$
$\neg(B_1 \uparrow B_2)$	B_1	B_2
$B_1 \downarrow B_2$	$\neg B_1$	$\neg B_2$
$B_1 \leftrightarrow B_2$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$
$\neg(B_1 \oplus B_2)$	$B_1 \rightarrow B_2$	$B_2 \rightarrow B_1$

Tableau

$n: \{A_1 \wedge A_2\} \cup U_0$

$n': \{A_1, A_2\} \cup U_0$

$n: \{B_1 \vee B_2\} \cup U_0$

$n': \{B_1\} \cup U_0 \qquad n'': \{B_2\} \cup U_0$

α	α_1	α_2
$\neg\neg A_1$	A_1	
$A_1 \wedge A_2$	A_1	A_2
$\neg(A_1 \vee A_2)$	$\neg A_1$	$\neg A_2$
$\neg(A_1 \rightarrow A_2)$	A_1	$\neg A_2$
$\neg(A_1 \uparrow A_2)$	A_1	A_2
$A_1 \downarrow A_2$	$\neg A_1$	$\neg A_2$
$A_1 \leftrightarrow A_2$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$
$\neg(A_1 \oplus A_2)$	$A_1 \rightarrow A_2$	$A_2 \rightarrow A_1$

β	β_1	β_2
$\neg(B_1 \wedge B_2)$	$\neg B_1$	$\neg B_2$
$B_1 \vee B_2$	B_1	B_2
$B_1 \rightarrow B_2$	$\neg B_1$	B_2
$B_1 \uparrow B_2$	$\neg B_1$	$\neg B_2$
$\neg(B_1 \downarrow B_2)$	B_1	B_2
$\neg(B_1 \leftrightarrow B_2)$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$
$B_1 \oplus B_2$	$\neg(B_1 \rightarrow B_2)$	$\neg(B_2 \rightarrow B_1)$

Sequent Calculus (Gentzen System S)

- The sequent calculus generalizes system G by allowing **sets** of formulas on **both** sides of |–. It was used by Gentzen to derive results about Natural Deduction.

- The intuitive meaning is:

$\{A_1, \ldots, A_m\} \mid\!- \{B_1, \ldots, B_n\}$ is like

$(A_1 \wedge \ldots \wedge A_m) \rightarrow (B_1 \vee \ldots \vee B_n)$

So one needs to pay careful attention to left vs. right.

The case of n = 1 is of most interest in the last step.

Ben-Ari’s Presentation of S

- Ben-Ari uses $\Rightarrow$ instead of $\vdash$
  to emphasize that in S, $\Rightarrow$ is part of the object language, rather than the meta-language.

- Some other authors and JAPE use $\vdash$ anyway.

- Smullyan uses $\rightarrow$ instead of $\vdash$, and uses $\supset$ for implication.

Sequent Calculus Rules

Definition 3.46 Axioms in the Gentzen sequent system $\mathcal{S}$ are sequents of the form: $U \cup \{A\} \Rightarrow V \cup \{A\}$. The rules of inference are:

op	Introduction into consequent	Introduction into antecedent
$\wedge$	$\dfrac{U \Rightarrow V \cup \{A\} \qquad U \Rightarrow V \cup \{B\}}{U \Rightarrow V \cup \{A \wedge B\}}$	$\dfrac{U \cup \{A, B\} \Rightarrow V}{U \cup \{A \wedge B\} \Rightarrow V}$
$\vee$	$\dfrac{U \Rightarrow V \cup \{A, B\}}{U \Rightarrow V \cup \{A \vee B\}}$	$\dfrac{U \cup \{A\} \Rightarrow V \qquad U \cup \{B\} \Rightarrow V}{U \cup \{A \vee B\} \Rightarrow V}$
$\rightarrow$	$\dfrac{U \cup \{A\} \Rightarrow V \cup \{B\}}{U \Rightarrow V \cup \{A \rightarrow B\}}$	$\dfrac{U \Rightarrow V \cup \{A\} \qquad U \cup \{B\} \Rightarrow V}{U \cup \{A \rightarrow B\} \Rightarrow V}$
$\neg$	$\dfrac{U \cup \{A\} \Rightarrow V}{U \Rightarrow V \cup \{\neg A\}}$	$\dfrac{U \Rightarrow V \cup \{A\}}{U \cup \{\neg A\} \Rightarrow V}$

Sequent Calculus vs. ND

Definition 3.46 Axioms in the Gentzen sequent system S are sequents of the form: $U \cup \{A\} \Rightarrow V \cup \{A\}$. Axioms generalize LEM (thus giving RAA, etc.)

op	Introduction into consequent	Introduction into antecedent
$\wedge$	generalizes ∧I $\dfrac{U \Rightarrow V \cup \{A\} \qquad U \Rightarrow V \cup \{B\}}{U \Rightarrow V \cup \{A \wedge B\}}$	generalizes ∧E $\dfrac{U \cup \{A, B\} \Rightarrow V}{U \cup \{A \wedge B\} \Rightarrow V}$
$\vee$	generalizes ∨I $\dfrac{U \Rightarrow V \cup \{A, B\}}{U \Rightarrow V \cup \{A \vee B\}}$	generalizes ∨E $\dfrac{U \cup \{A\} \Rightarrow V \qquad U \cup \{B\} \Rightarrow V}{U \cup \{A \vee B\} \Rightarrow V}$
$\rightarrow$	generalizes →I $\dfrac{U \cup \{A\} \Rightarrow V \cup \{B\}}{U \Rightarrow V \cup \{A \rightarrow B\}}$	generalizes →E $\dfrac{U \Rightarrow V \cup \{A\} \qquad U \cup \{B\} \Rightarrow V}{U \cup \{A \rightarrow B\} \Rightarrow V}$
$\neg$	generalizes ¬I $\dfrac{U \cup \{A\} \Rightarrow V}{U \Rightarrow V \cup \{\neg A\}}$	generalizes ¬E $\dfrac{U \Rightarrow V \cup \{A\}}{U \cup \{\neg A\} \Rightarrow V}$

Example Sequent Calculus Proof

Constructed from bottom to top:

$$\dfrac{\dfrac{\dfrac{}{P \Rightarrow P, Q}\ \text{Axiom}}{P, \neg P \Rightarrow Q}\ \neg\ \text{to antecedent} \qquad \dfrac{}{Q, \neg P \Rightarrow Q}\ \text{Axiom}}{P \vee Q, \neg P \Rightarrow Q}\ \vee\ \text{to antecedent}$$