ELEMENTS OF DEDUCTIVE LOGIC

7. A closer look at tableaux

J. Chandler

KUL 2012

Recap

- Our current concern: which sentential argument forms are valid and which are not?
- A few sessions ago:
  - Validity of sentential form is down to the truth-functional behaviour of the connectives.
  - Validity can be proven/disproven using a bottom-up method known as truth tables.
- But truth tables are *very* inefficient!
- Last time:
  - The general idea behind a fast, top-down approach known as tableaux.
  - Some examples of the method in action.
- This time: a more rigorous look at the tableau method.

General terminology and concepts

- A tableau for a set of sentences Γ is an inverted tree, such that:
  - (i) The members of Γ lie at at its root
  - (i) Any further sentence that is on one of its branches is derived from some other sentence further up that branch, in accordance with an appropriate resolving rule.
- A branch is closed iff it contains both a sentence and its negation.
- A branch is open iff it is not closed.
- A tableau is closed iff all of its branches are closed.
- A tableau is open iff at least one of its branches is open.
- A tableau is completed iff no resolving rule can be applied to any sentences on *open* branches.

Resolving rules: overview

- We have 4 ‘binary’ connectives (connectives with 2 inputs)
- We actually need 2 rules for each of these.
  - one for sentences of the form $\varphi \bullet \psi$,
  - one for sentences of the form $\sim(\varphi \bullet \psi)$.
- That’s 8 rules.
- We also have 1 ‘unary’ connective (connective with 1 input)
- Grand total: 9 rules
- Warning: there are two *serious* typos in the textbook here!!

Resolving rules: conjunctions

- To resolve a sentence of the form $\varphi \& \psi$, extend any open branch in which the sentence occurs with a single branch segment containing both φ and ψ.

$$\begin{array}{c} (\varphi \& \psi) \\ | \\ \varphi \\ \psi \end{array}$$

Resolving rules: disjunctions

- To resolve a sentence of the form $\varphi \vee \psi$, extend any open branch in which the sentence occurs with two branch segments, one containing φ and one containing ψ.

$$\begin{array}{c} (\varphi \vee \psi) \\ \swarrow \quad \searrow \\ \varphi \qquad \psi \end{array}$$

Resolving rules: negated disjunctions

- Wrongly stated in the textbook!
- To resolve a sentence of the form $\sim(\varphi \vee \psi)$, extend any open branch in which the sentence occurs with a single branch segment containing both $\sim\varphi$ and $\sim\psi$.

$$\begin{array}{c} \sim(\varphi \vee \psi) \\ | \\ \sim\varphi \\ \sim\psi \end{array}$$

- Rationale: we exploit the equivalence
  $\sim(\varphi \vee \psi) \Leftrightarrow \sim\varphi \& \sim\psi$

Resolving rules: negated conjunctions

- To resolve a sentence of the form $\sim(\varphi \& \psi)$, extend any open branch in which the sentence occurs with two new branch segments, one containing $\sim\varphi$ and one containing $\sim\psi$.

$$\begin{array}{c} \sim(\varphi \& \psi) \\ \swarrow \quad \searrow \\ \sim\varphi \qquad \sim\psi \end{array}$$

- Rationale: we exploit the equivalence
  $\sim(\varphi \& \psi) \Leftrightarrow \sim\varphi \vee \sim\psi$

Resolving rules: conditionals

- To resolve a sentence of the form $\varphi \supset \psi$, extend any open branch in which the sentence occurs with two new branch segments, one containing $\sim\varphi$ and one containing ψ.

$$(\varphi \supset \psi)$$
$$\sim\varphi \qquad \psi$$

- Rationale: we exploit the equivalence
$$\varphi \supset \psi \Leftrightarrow \sim\varphi \vee \psi$$

Resolving rules: negated conditionals

- Wrongly stated in the textbook!
- To resolve a sentence of the form $\sim(\varphi \supset \psi)$, extend any open branch in which the sentence occurs with a single branch segment containing both φ and $\sim\psi$.

$$\sim(\varphi \supset \psi)$$
$$|$$
$$\varphi$$
$$\sim\psi$$

- Rationale: we exploit the equivalence
$$\sim(\varphi \supset \psi) \Leftrightarrow \varphi \& \sim\psi$$

Resolving rules: biconditionals

- To resolve a sentence of the form $\varphi \equiv \psi$, extend any open branch in which the sentence occurs with two new branch segments, one containing both φ and ψ and one containing both $\sim\varphi$ and $\sim\psi$.

$$(\varphi \equiv \psi)$$
$$\varphi \qquad \sim\varphi$$
$$\psi \qquad \sim\psi$$

Resolving rules: negated biconditionals

- To resolve a sentence of the form $\sim(\varphi \equiv \psi)$, extend any open branch in which the sentence occurs with two new branch segments, one containing both $\sim\varphi$ and ψ and one containing both φ and $\sim\psi$.

$$\sim(\varphi \equiv \psi)$$
$$\sim\varphi \qquad \varphi$$
$$\psi \qquad \sim\psi$$

- Rationale: we exploit the equivalence
$$\sim(\varphi \equiv \psi) \Leftrightarrow (\sim\varphi \& \psi) \vee (\varphi \& \sim\psi)$$

Resolving rules: double negations

- To resolve a sentence of the form $\sim\sim\varphi$, extend any open branch in which the sentence occurs with a single branch segment containing φ.

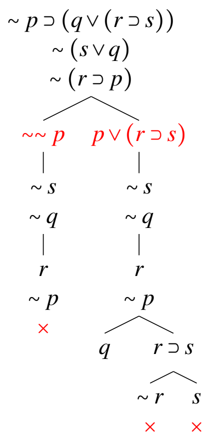
$$\begin{array}{c}\sim\sim\psi \\ | \\ \psi\end{array}$$

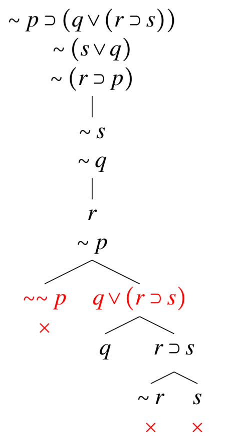
Rule application order

- Some good news:
  - (i) If you keep applying the rules, in *whatever order*, you are guaranteed to obtain a completed tableau in a *finite number of steps*.
  - (ii) If one completed tableau for Γ is closed (open), then *every* completed tableau for Γ is closed (open).
- Some even better news:
  - (iii) You can reduce this finite number of steps by following some guidelines regarding the order of application of the rules.

Tip: early vs late branching

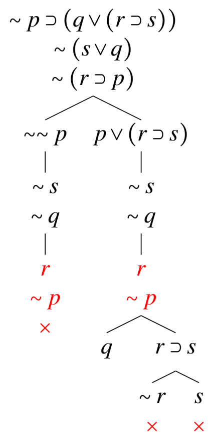
- Tip#1: prioritize non-branching rules.

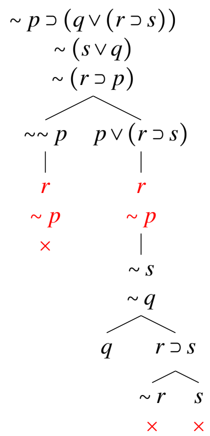




Tip: early vs late closure

- Tip #2: prioritize rules that immediately close branches.





Some notation

- There is a special symbol $\vdash$ to talk about tableaux for a set of sentences Γ.
- We write $\Gamma \vdash$ to indicate that there exists a closed completed tableau for Γ.
- We write $\Gamma \nvdash$ to indicate that there exists an open completed tableau for Γ.
- We write $\Gamma \vdash \varphi$ to abbreviate $\Gamma \cup \{\sim \varphi\} \vdash$.

  (Where $\Gamma \cup \{\sim \varphi\}$ denotes the set whose members include those of Γ, plus $\sim \varphi$.)
- We write $\Gamma \nvdash \varphi$ to abbreviate $\Gamma \cup \{\sim \varphi\} \nvdash$.

Introducing soundness and completeness

- The claim that the tableau proof method can be used to test for validity breaks down into two subclaims.
- The first subclaim is that the method is sound:

  If $\Gamma \vdash \varphi$, then $\Gamma \vDash \varphi$

  If there exists a closed completed tableau for $\Gamma \cup \{\sim \varphi\}$, then the argument from Γ to φ is valid.
- The second subclaim is that the method is complete:

  If $\Gamma \vDash \varphi$, then $\Gamma \vdash \varphi$

  If the argument from Γ to φ is valid, then there exists a closed completed tableau for $\Gamma \cup \{\sim \varphi\}$.

Introducing soundness and completeness (ctd.)

- Note that I haven't given you a rigorous *proof* that either of these subclaims are true.
- I will give you a brief sketch of the proof for soundness.
- For the full version, as well as for the proof for completeness, see Restall, pp. 69–74.
- Regarding soundness, we want to show that if $\Delta \vdash$ then $\Delta \vDash$ (where $\Delta = \Gamma \cup \{\sim \varphi\}$).
- Proof strategy: We assume, for sake of argument, that $\Delta \vdash$ but $\Delta \nvDash$ and show that this would lead to a contradiction.
- Upshot: It cannot be the case that $\Delta \vdash$ but $\Delta \nvDash$; in other words, if $\Delta \vdash$, then $\Delta \vDash$.

Soundness: proof sketch

(1) $\Delta \vdash$ but $\Delta \nvDash$. *(Assumption)*

(2) There is a valuation, call it v, that satisfies all sentences in Δ. *(From the 2^{nd} conjunct of (1))*

(3) If v satisfies a sentence φ, then it satisfies the sentences in at least one of the branch segments resulting from the resolution of φ *(From rules and tables for connectives)*

(4) Every completed tree for Δ has a branch, call it b, such that all sentences on b are satisfied by v. *(From (2) and (3))*

(5) There is no φ such that $v(\varphi) = v(\sim \varphi) = 1$ *(From table for $\sim$)*

(6) Branch b is open. *(From (5))*

(7) Every completed tree for Δ has an open branch (namely b). *(From (4) and (6))*

(8) $\Delta \nvdash$, in contradiction with (1). *(From (7))*

Next session

- Exercise class: please do exercise set # 3.