Elements of Deductive Logic

11. Gaps & Gluts: tableau methods

J. Chandler

KUL 2012

Introduction

- Last session: a set of trivalent semantics accomodating the possibility of ‘gappy’ or ‘glutty’ sentences.
- This time: some tableau methods for testing for validity.
- SV validates same inferences as classical logic so no need for new methods there.
- Here, systems for:
  - K3
  - LP
- For systems for Ł3 and RM3, see Priest (2008, p. 150-151).
- These systems are demonstrably sound and complete wrt the relevant semantics (proof omitted; see Priest).

General comments

- If you can do tableaux for classical sentential logic, this will be easy: just a few small differences.
- Tableau method: find valuations such that the premises have a designated value and the conclusion does not.
- Classical semantics: find valuations such that the premises are true and the conclusion is *false*.
- So to evaluate $\varphi_1, \ldots, \varphi_n \vdash \psi$, we start the tree with:

$$\begin{array}{c} \varphi_1 \\ \ldots \\ \varphi_n \\ \sim\psi \end{array}$$

General comments (ctd.)

- In our gappy trivalent semantics: false isn’t the only non-designated value!

  ⇒ Find valuations such that the premises are true and the conclusion is *either (i) false or (ii) neither true nor false*.
- To evaluate $\varphi_1, \ldots, \varphi_n \vdash \psi$, we start the tree with:

$$\begin{array}{c} \varphi_1, + \\ \ldots \\ \varphi_n, + \\ \psi, - \end{array}$$

  Where:
  - ‘+’ =‘designated’ (1)
  - ‘-’ = ‘non-designated’ (0 or *i*)

General comments (ctd.)

- The closing rules also change…
- In classical semantics: the tableau closes iff we have, for any sentence φ, both φ and $\sim\varphi$ on the same branch.
- Here: the branch closes iff we have, for any sentence φ
  (i) both $\varphi, +$ and $\varphi, -$ on the same branch *or*
  (ii) both $\varphi, +$ and $\sim\varphi, +$ on the same branch
- Regarding (i): no sentence can be assigned both 1 and either 0 or *i* (because valuations only assign one value).
- Regarding (ii): no sentence can be such that both it and its negation are assigned 1 (because of the truth tables for negation).

General comments (ctd.)

- Countermodels: if a branch b fails to close, for every atomic sentence φ
  - if $\varphi, +$ is on b, then $v(\varphi) = 1$
  - if $\sim\varphi, +$ is on b, then $v(\varphi) = 0$
  - Otherwise $v(\varphi) = i$

  Note: we never have both $\varphi, +$ and $\sim\varphi, +$ on an open branch.
- Tip: if you have a *prima facie* countermodel, double-check that it *is* indeed one, using the truth tables.

Tableaux for K3: negation

- With the new t-tables come new tableau rules…
- Negation:

$f_{\sim}$	
1	0
0	1
i	i

Table for ~

$$\begin{array}{c} \sim\sim\varphi, + \\ | \\ \varphi, + \end{array}$$

Rule #1

$$\begin{array}{c} \sim\sim\varphi, - \\ | \\ \varphi, - \end{array}$$

Rule #2

$f_{\sim\sim}$	
1	1
0	0
i	i

Table for ~~

Tableaux for K3: conjunction

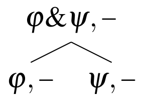
- Conjunction:

$f_{\&}$	1	0	i
1	1	0	i
0	0	0	0
i	i	0	i

Table for &

$$\begin{array}{c} \varphi \& \psi, + \\ | \\ \varphi, + \\ \psi, + \end{array}$$

Rule #3



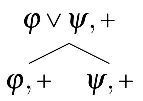
Rule #4

Tableaux for K3: disjunction

- Disjunction:

$f_\vee$	1	0	i
1	1	1	1
0	1	0	i
i	1	i	i

Table for $\vee$


$\varphi \vee \psi, +$

$\varphi, +$ $\psi, +$

Rule #5

$\varphi \vee \psi, -$
|
$\varphi, -$
$\psi, -$

Rule #6

Tableaux for K3: negated conjunction

- Negated conjunction:

$f_{\sim(.\&.)}$	1	0	i
1	0	1	i
0	1	1	1
i	i	1	i

Table for $\sim(.\&.)$

$\sim(\varphi \& \psi), +$
|
$\sim\varphi \vee \sim\psi, +$

Rule #7

$\sim(\varphi \& \psi), -$
|
$\sim\varphi \vee \sim\psi, -$

Rule #8

$f_\vee$	1	0	i
1	1	1	1
0	1	0	i
i	1	i	i

Table for $\vee$

$f_{\sim.\vee\sim.}$	1	0	i
1	0	1	i
0	1	1	1
i	i	1	i

Table for $\sim .\vee \sim .$

Tableaux for K3: negated disjunction

- Negated disjunction:

$f_{\sim(.\vee.)}$	1	0	i
1	0	0	0
0	0	1	i
i	0	i	i

Table for $\sim(.\vee.)$

$\sim(\varphi \vee \psi), +$
|
$\sim\varphi \& \sim\psi, +$

Rule #9

$\sim(\varphi \vee \psi), -$
|
$\sim\varphi \& \sim\psi, -$

Rule #10

$f_\&$	1	0	i
1	1	0	i
0	0	0	0
i	i	0	i

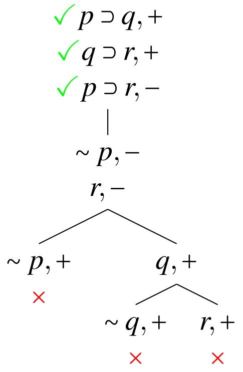
Table for &

$f_{\sim.\&\sim.}$	1	0	i
1	0	0	0
0	0	1	i
i	0	i	i

Table for $\sim .\& \sim .$

Example 1: closed tableau

- We show that $p \supset q, q \supset r \vdash_{K3} p \supset r$


✓ $p \supset q, +$
✓ $q \supset r, +$
✓ $p \supset r, -$
|
$\sim p, -$
$r, -$

$\sim p, +$ $q, +$
×
$\sim q, +$ $r, +$
× ×

(Branch b closes iff either (i) $\varphi, +$ and $\varphi, -$ on b or (ii) $\varphi, +$ and $\sim\varphi, +$ on b.)

Example 2: open tableau

- We show that $p \nvdash_{K3} q \vee \neg q$:

$$\begin{array}{c} p, + \\ \checkmark\, q \vee \sim q, - \\ | \\ q, - \\ \sim q, - \\ \uparrow \end{array}$$

Countermodel: $v(p) = 1$ and $v(q) = i$

(If b open, then (i) if $\varphi, +$ is on b, then $v(\varphi) = 1$, (ii) if $\sim \varphi, +$ is on b, then $v(\varphi) = 0$, (iii) otherwise $v(\varphi) = i$)

General comments

- Tableau method: find valuations such that the premises have a designated value and the conclusion does not.
- Classical semantics: find valuations such that the premises are *true* and the conclusion is false.
- In our glutty trivalent semantics: true isn't the only designated value!

  ⇒ Find valuations such that the premises are *either (i) true or (ii) both true and false* and the conclusion is false.
- Again, we use '+' and '-'.
- But this time, since i is designated:
  - '+' = 'designated' stands for 1 or i
  - '-' ='non-designated' stands only for 0

General comments (ctd.)

- Again, the closing rules differ from those for classical logic.
- Here: the branch closes iff we have, for some sentence φ
  (i) both $\varphi, +$ and $\varphi, -$ on the same branch *or*
  (ii) both $\varphi, -$ and $\sim \varphi, -$ on the same branch
- Regarding (i): no sentence can be assigned both 1 and either 0 or i (because valuations only assign one value).
- Regarding (ii): no sentence can be such that both it and its negation are assigned 0 (because of the truth tables for negation).
- LP therefore differs from K3 wrt (ii)...

LP vs K3

- Question 1:

  Why doesn't the tableau close here if we have both $\varphi, +$ and $\sim \varphi, +$ on the same branch, like it does for K3?
- Answer:

  Because here, '+' stands for 'either 1 or i' and we can have $v(\varphi) = i = v(\sim \varphi)$.
- Question 2:

  Why, in K3, doesn't the tableau close if we have both $\varphi, -$ and $\sim \varphi, -$ on the same branch, like it does here?
- Answer:

  Because, in K3, '-' stands for 'either 0 or i' and we can have $v(\varphi) = i = v(\sim \varphi)$.

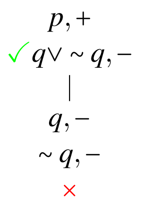
LP vs K3 (ctd.)

- Aside from this: same rules for branches.
- Countermodels: if a branch b fails to close, for every atomic sentence φ
  - if $\varphi, -$ is on b, then $v(\varphi) = 0$
  - if $\sim \varphi, -$ is on b, then $v(\varphi) = 1$
  - Otherwise $v(\varphi) = i$

  Note: we never have both $\varphi, -$ and $\sim \varphi, -$ on an open branch.

Example 1: closed tableau

- We show that $p \vdash_{LP} q \vee \neg q$:

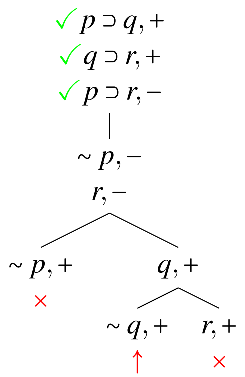


$p, +$
✓ $q \vee \sim q, -$
$q, -$
$\sim q, -$
×

(Branch b closes iff either (i) $\varphi, +$ and $\varphi, -$ on b or (ii) $\varphi, -$ and $\sim \varphi, -$ on b.)

Example 2: open tableau

- We show that $p \supset q, q \supset r \nvdash_{LP} p \supset r$



✓ $p \supset q, +$
✓ $q \supset r, +$
✓ $p \supset r, -$
$\sim p, -$
$r, -$
$\sim p, +$ $q, +$
×
$\sim q, +$ $r, +$
↑ ×

Countermodel: $v(p) = 1$, $v(q) = i$ and $v(r) = 0$
(If b open, then (i) if $\varphi, -$ on b, then $v(\varphi) = 0$, (ii) if $\sim \varphi, -$ on b, then $v(\varphi) = 1$, (iii) otherwise $v(\varphi) = i$)

Next session

- Topic: introducing predicate logic.
- Reading: Restall, Ch. 8, up to, but excluding, ‘Translation’.

References

- Priest, G. (2008). *An Introduction to Non-Classical Logic*. Cambridge: CUP.