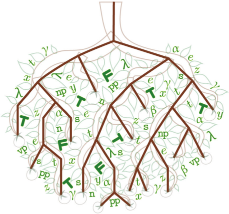
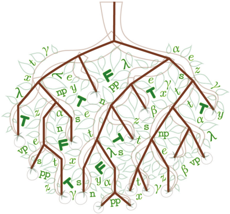
A Natural Proof System for Natural Language

NPS4NL-2: Semantic Tableau Method



Lasha Abzianidze Reinhard Muskens

university of groningen TILBURG UNIVERSITY



ESSLLI 2019 in Rīga, Latvija

Logic & proof systems

Logic consists of four components:

- Intuitive non-formal motivation
- Syntax of formulas: well-formed formulas vs ill-formed ones
- Semantics associated with the formulas
- Some type of proof calculus

A proof calculus/system:

- employed to systematically capture valid formulas and arguments
- is a syntactic game: there are legal and illegal moves
- comes in several flavours
- is usually a sound and complete

Semantic tableau method

A semantic tableau method [Beth, 1955, Hintikka, 1955] is a proof procedure for formal logics that checks formulas with truth constraints:

Input: A set of signed formulas
$P_1 : \mathbb{T}, \ldots, P_m : \mathbb{T},\ Q_1 : \mathbb{F}, \ldots, Q_n : \mathbb{F}$

Output: some or no model satisfying the truth constraints on the formulas
☞ A model search problem

Prove or refute

Whenever it rains, the roof leaks

How to verify truth of this statement?

Show that:

- In every situation it is true

  Approval route

  Check every situation when it rains and show the roof leaking
- In some situation it is not true

  Refutation route

  Find some situation when it rains and the roof isn't leaking

Proving by failing to refute

A tableau method tries to refute statement in order to prove it:

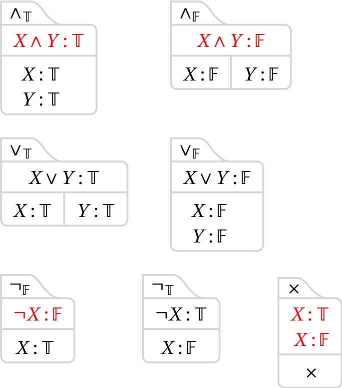
1. Given $P_1,...,P_m \vDash Q$ to prove
2. Try to refute $P_1,...,P_m \vDash Q$
    1. Build the counterexample: $P_1:\mathbb{T},\ldots,P_m:\mathbb{T},\ Q:\mathbb{F}$
    2. Try to satisfy the counterexample
3. If refutation succeeded, $P_1,...,P_m \vDash Q$ is disproved
4. Otherwise $P_1,...,P_m \vDash Q$ is proved

Propositional tableau method (signed version)

Prove: $P \wedge Q \vDash Q \wedge \neg P$

Counterexample: $P \wedge Q : \mathbb{T},\ \ Q \wedge \neg P : \mathbb{F}$

Propositional tableau rules:



$\wedge_{\mathbb{T}}$: $X \wedge Y : \mathbb{T}$ / $X : \mathbb{T}$, $Y : \mathbb{T}$

$\wedge_{\mathbb{F}}$: $X \wedge Y : \mathbb{F}$ / $X : \mathbb{F}$ | $Y : \mathbb{F}$

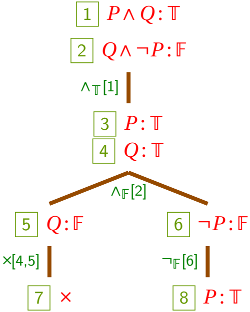
$\vee_{\mathbb{T}}$: $X \vee Y : \mathbb{T}$ / $X : \mathbb{T}$ | $Y : \mathbb{T}$

$\vee_{\mathbb{F}}$: $X \vee Y : \mathbb{F}$ / $X : \mathbb{F}$, $Y : \mathbb{F}$

$\neg_{\mathbb{F}}$: $\neg X : \mathbb{F}$ / $X : \mathbb{T}$

$\neg_{\mathbb{T}}$: $\neg X : \mathbb{T}$ / $X : \mathbb{F}$

×: $X : \mathbb{T}$, $X : \mathbb{F}$ / ×



1 $P \wedge Q : \mathbb{T}$

2 $Q \wedge \neg P : \mathbb{F}$

$\wedge_{\mathbb{T}}[1]$

3 $P : \mathbb{T}$

4 $Q : \mathbb{T}$

$\wedge_{\mathbb{F}}[2]$

5 $Q : \mathbb{F}$ | 6 $\neg P : \mathbb{F}$

×[4,5] | $\neg_{\mathbb{F}}[6]$

7 × | 8 $P : \mathbb{T}$

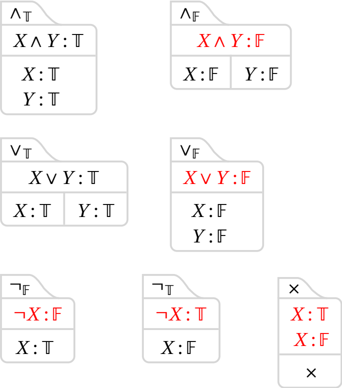
A situation supporting a counterexample: $P : \mathbb{T}, Q : \mathbb{T}$

Closed tableau

Prove: $\neg(P\wedge Q)\vDash\neg P\vee\neg Q$ Proved!

Counterexample: $\neg(P\wedge Q):\mathbb{T},\ \neg P\vee\neg Q:\mathbb{F}$

Propositional tableau rules:



$\wedge_{\mathbb{T}}$

$X\wedge Y:\mathbb{T}$

$X:\mathbb{T}$

$Y:\mathbb{T}$

$\wedge_{\mathbb{F}}$

$X\wedge Y:\mathbb{F}$

$X:\mathbb{F}$ | $Y:\mathbb{F}$

$\vee_{\mathbb{T}}$

$X\vee Y:\mathbb{T}$

$X:\mathbb{T}$ | $Y:\mathbb{T}$

$\vee_{\mathbb{F}}$

$X\vee Y:\mathbb{F}$

$X:\mathbb{F}$

$Y:\mathbb{F}$

$\neg_{\mathbb{F}}$

$\neg X:\mathbb{F}$

$X:\mathbb{T}$

$\neg_{\mathbb{T}}$

$\neg X:\mathbb{T}$

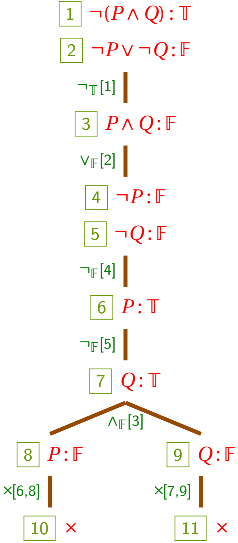
$X:\mathbb{F}$

×

$X:\mathbb{T}$

$X:\mathbb{F}$

×



$\boxed{1}\ \neg(P\wedge Q):\mathbb{T}$

$\boxed{2}\ \neg P\vee\neg Q:\mathbb{F}$

$\neg_{\mathbb{T}}[1]$

$\boxed{3}\ P\wedge Q:\mathbb{F}$

$\vee_{\mathbb{F}}[2]$

$\boxed{4}\ \neg P:\mathbb{F}$

$\boxed{5}\ \neg Q:\mathbb{F}$

$\neg_{\mathbb{F}}[4]$

$\boxed{6}\ P:\mathbb{T}$

$\neg_{\mathbb{F}}[5]$

$\boxed{7}\ Q:\mathbb{T}$

$\wedge_{\mathbb{F}}[3]$

$\boxed{8}\ P:\mathbb{F}$

$\boxed{9}\ Q:\mathbb{F}$

×[6,8]

×[7,9]

$\boxed{10}$ ×

$\boxed{11}$ ×

Different proof strategy

Prove: $\neg(P \wedge Q) \vDash \neg P \vee \neg Q$ Prover!

Counterexample: $\neg(P \wedge Q) : \mathbb{T},\ \neg P \vee \neg Q : \mathbb{F}$

Propositional tableau rules:

$\wedge_{\mathbb{T}}$

$$\begin{array}{c} X \wedge Y : \mathbb{T} \\ \hline X : \mathbb{T} \\ Y : \mathbb{T} \end{array}$$

$\wedge_{\mathbb{F}}$

$$\begin{array}{c|c} \multicolumn{2}{c}{X \wedge Y : \mathbb{F}} \\ \hline X : \mathbb{F} & Y : \mathbb{F} \end{array}$$

$\vee_{\mathbb{T}}$

$$\begin{array}{c|c} \multicolumn{2}{c}{X \vee Y : \mathbb{T}} \\ \hline X : \mathbb{T} & Y : \mathbb{T} \end{array}$$

$\vee_{\mathbb{F}}$

$$\begin{array}{c} X \vee Y : \mathbb{F} \\ \hline X : \mathbb{F} \\ Y : \mathbb{F} \end{array}$$

$\neg_{\mathbb{F}}$

$$\begin{array}{c} \neg X : \mathbb{F} \\ \hline X : \mathbb{T} \end{array}$$

$\neg_{\mathbb{T}}$

$$\begin{array}{c} \neg X : \mathbb{T} \\ \hline X : \mathbb{F} \end{array}$$

×

$$\begin{array}{c} X : \mathbb{T} \\ X : \mathbb{F} \\ \hline \times \end{array}$$

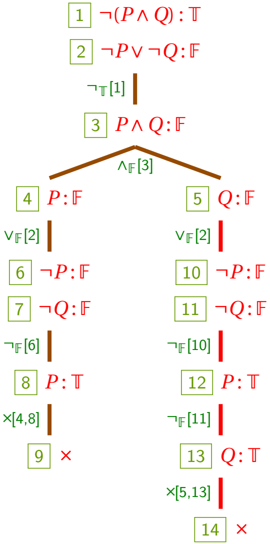
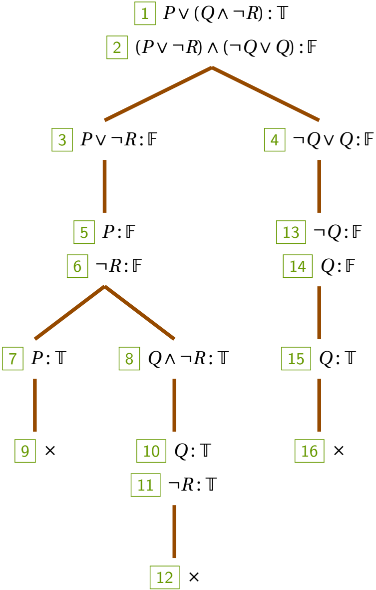


Tableau exercise



$\boxed{1}$ $P \vee (Q \wedge \neg R) : \mathbb{T}$

$\boxed{2}$ $(P \vee \neg R) \wedge (\neg Q \vee Q) : \mathbb{F}$

$\boxed{3}$ $P \vee \neg R : \mathbb{F}$

$\boxed{4}$ $\neg Q \vee Q : \mathbb{F}$

$\boxed{5}$ $P : \mathbb{F}$

$\boxed{6}$ $\neg R : \mathbb{F}$

$\boxed{13}$ $\neg Q : \mathbb{F}$

$\boxed{14}$ $Q : \mathbb{F}$

$\boxed{7}$ $P : \mathbb{T}$

$\boxed{8}$ $Q \wedge \neg R : \mathbb{T}$

$\boxed{15}$ $Q : \mathbb{T}$

$\boxed{9}$ ×

$\boxed{10}$ $Q : \mathbb{T}$

$\boxed{11}$ $\neg R : \mathbb{T}$

$\boxed{16}$ ×

$\boxed{12}$ ×

Quiz

1. If propositional formula ϕ is built up from n Boolean connectives, at most how many rule applications will be applicable to the tableau started with $\phi : \mathbb{T}$?
2. ... started with $\phi : \mathbb{F}$?
3. Can you think of tableau rules for $\rightarrow_{\mathbb{T}}$ and $\rightarrow_{\mathbb{F}}$?

Rules for quantifiers

Rules for ∃:

$\exists_{\mathbb{T}}$

$\exists x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is fresh

$\exists^{c}_{\mathbb{F}}$

$\exists x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is old

Rules for ∀:

$\forall_{\mathbb{F}}$

$\forall x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is fresh

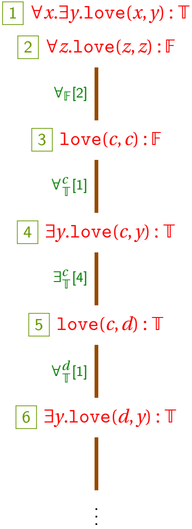
$\forall^{c}_{\mathbb{T}}$

$\forall x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is old

⚠ Dangerous zone!



$\boxed{1}$ $\forall x.\exists y.\texttt{love}(x,y) : \mathbb{T}$

$\boxed{2}$ $\forall z.\texttt{love}(z,z) : \mathbb{F}$

$\forall_{\mathbb{F}}[2]$

$\boxed{3}$ $\texttt{love}(c,c) : \mathbb{F}$

$\forall^{c}_{\mathbb{T}}[1]$

$\boxed{4}$ $\exists y.\texttt{love}(c,y) : \mathbb{T}$

$\exists^{c}_{\mathbb{T}}[4]$

$\boxed{5}$ $\texttt{love}(c,d) : \mathbb{T}$

$\forall^{d}_{\mathbb{T}}[1]$

$\boxed{6}$ $\exists y.\texttt{love}(d,y) : \mathbb{T}$

⋮

Non-empty domain

Rules for ∃:

$\exists_{\mathbb{T}}$

$\exists x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is fresh

$\exists^{c}_{\mathbb{F}}$

$\exists x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is old

Rules for ∀:

$\forall_{\mathbb{F}}$

$\forall x.\phi : \mathbb{F}$

$\phi[x/c] : \mathbb{F}$

c is fresh

$\forall^{c}_{\mathbb{T}}$

$\forall x.\phi : \mathbb{T}$

$\phi[x/c] : \mathbb{T}$

c is old

$\boxed{1}$ $\forall x.\big(\mathtt{run}(x) \wedge \neg\mathtt{run}(x)\big) : \mathbb{T}$

Non-empty domain constraint:
you can always have an entity

Simple type theory

We will use Simple Type Theory [Church, 1940] as a Higher-Order Logic.

A type system built up from e (*entity*) and t (*truth*) basic types:

- e and t are types;
- if α and β are types, so are $(\alpha\beta)$

Examples of types:

- t for sentences, e.g., *John sleeps*
- et for common nouns and intransitive verbs, e.g., *sleep*, *cat*
- $(et)(et)t$ for determiners
- $(et)(et)$ for adjectives
- eet for transitive verbs
- e for proper names (also $(et)t$ is possible)

Typed terms

We assume to have infinite number of constant and variable terms of each type.

Compound terms are combined and typed as:

- if B is of type $(\alpha\beta)$,
  and A is of type α,
  then BA is of type β.
- if variable x is of type α,
  and B is of type β,
  then $\lambda x.B$ is of type $(\alpha\beta)$.

Association conventions:

- $ABC = (AB)C$
- $(\alpha\beta\gamma) = \alpha(\beta\gamma)$, e.g., $(et)(et) = (et)et$

Modeling arithmetic functions

Types for numbers:

- Basic types $\mathbb{N}$ for natural numbers and $\mathbb{R}$ for real numbers.
- Compound types: $\mathbb{NR}$, $\mathbb{NNN}$, $\mathbb{RRR}$, $\mathbb{RR}$, $\mathbb{NN}$, $\mathbb{RN}$, ...

Typed terms:

- Constants: $1_{\mathbb{N}}$, $3.1415_{\mathbb{R}}$, $1_{\mathbb{R}}$, $+_{\mathbb{NNN}}$, $\times_{\mathbb{NNN}}$, $\surd_{\mathbb{NR}}$, ...
- Compound terms:

$+_{\mathbb{NNN}} 1_{\mathbb{N}}$ is of type $\mathbb{NN}$,

$\surd_{\mathbb{NR}} 1_{\mathbb{N}}$ is of type $\mathbb{R}$,

$\times_{\mathbb{NNN}} 1_{\mathbb{N}}$ is of type $\mathbb{NN}$,

$\lambda x_{\mathbb{R}}.1_{\mathbb{N}}$ is of type $\mathbb{RN}$

Conclusion

- A semantic tableau method
  "today [it is] one of the most popular, since it appears to bring together the proof-theoretical and the semantical approaches to the presentation of a logical system and is also very intuitive. In many universities it is the style first taught to students." [D'Agostino et al., 1999].
- Propositional tableau system: when applying a rule to a tableau entry, remember to do so for each branch it sits on.
- Dangerous zone: First-order logic tableau might not terminate
- Simple type theory: typed terms model higher-order functions

References I

Beth, E. W. (1955). Semantic Entailment and Formal Derivability. *Koninklijke Nederlandse Akademie van Wentenschappen, Proceedings of the Section of Sciences*, 18:309–342.

Church, A. (1940). A formulation of the simple theory of types. *Jurnal of Symbolic Logic*, 5(2):56–68.

D'Agostino, M., Gabbay, D. M., Hähnle, R., and Posegga, J., editors (1999). *Handbook of Tableau Methods*. Springer.

Hintikka, J. (1955). *Two Papers on Symbolic Logic: Form and Content in Quantification Theory and Reductions in the Theory of Types*. Number 8 in Acta philosophica Fennica. Societas Philosophica.