Arbeitskreis Modallogik

Was genau sind Tableaux?

Was genau sind Tableaux – Ein bedeutsamer Unterschied

[F]rom an abstract point of view the tree display of […] formulas is not a tableau [itself] but <u>an implementation of a tableau</u>.

(Fitting, 1999, S. 5, Hervorhebung hinzugefügt)

Was genau sind Tableaux – Ungeordnete Bäume

Ein ungeordneter Baum ist ein Tripel <S, l, R> mit folgenden Eigenschaften:

(1) S ist eine nicht-leere Menge, deren Elemente wir „Knoten“ (nodes) nennen.

(2) l ist eine Funktion, die jedem Knoten x eine positive ganze Zahl l(x) zuweist, die wir „Level von x“ nennen.

(3) R ist eine zweistellige Relation auf S, die wir als „x ist Vorgänger von y“ bzw. „y ist Nachfolger von x“ lesen. Diese Relation hat folgende Eigenschaften:

root

(a) Auf Level 1 gibt es nur einen einzigen Knoten – die Wurzel des Baums.

(b) Jeder Knoten außer der Wurzel hat genau einen Vorgänger.

(c) Für alle Knoten x, y gilt: Wenn Rxy, dann l(y) = l(x) + 1.

Was genau sind Tableaux – ein wenig Vokabular I

Ein Endknoten (tip) ist ein Knoten ohne Nachfolger.

Ein einfacher Knoten ist ein Knoten mit genau einem Nachfolger.

Ein Zweig ist ein Knoten mit mehr als einem Nachfolger.

Was genau sind Tableaux – ein wenig Vokabular

Ein Pfad P eines Tableaus T nach x ist eine Folge von endlich vielen oder abzählbar unendlich vielen Knoten von T, für die gilt:

(a) Der erste Knoten von P ist die Wurzel von T
(b) Der letzte Knoten von P ist x, oder es gibt keinen letzten Knoten
(c) Jeder Knoten bis auf x steht in Relation zum darauffolgenden Knoten.

Aus (3a), (3b) und (3c) folgt: Für jeden Knoten k von T gibt es einen Pfad nach k.

Was genau sind Tableaux – ein wenig Vokabular

Ein Ast (branch) eines Tableaus T ist ein Pfad, für den gilt: Entweder ist sein letzter Knoten ein Endknoten von T, oder es gibt keinen letzten Knoten von T.

y dominiert x genau dann, wenn y auf dem Pfad nach x liegt.

y liegt über x genau dann, wenn y x dominiert und $y \neq x$.

y liegt direkt über x genau dann, wenn y über x liegt und Ryx.

Was genau sind Tableaux – geordnete Bäume

Sei z ein Zweig, d. h. ein Knoten, der mehr als einen Nachfolger hat. Ein geordneter Baum $<S, l, R, \theta>$ ist ein ungeordneter Baum zusammen mit einer Funktion $\theta(z)$, für die gilt:

(a) $\theta(z)$ ist die Folge aller Nachfolger von z,

(b) $\theta(z)$ enthält keine Wiederholungen.

Was genau sind Tableaux – ein Definitionsvorschlag

Ein <u>Tableau</u> für eine Menge $M = \{m_1, \ldots, m_n\}$ ist ein geordneter Baum $<S, l, R, \theta>$, für den gilt:

1) Alle Elemente von S sind *Vorkommnisse von Formeln*
2) M ist Teilmenge von S
3) Jedes Element von M hat ein einzigartiges Level
4) Es gibt kein $s \in S$, sodass $s \notin M$ und Rsm
5) Jeder Zweig von T hat genau zwei Nachfolger (d. h. ist dyadisch)

Ein Ast, der sowohl α als auch ⌜~α⌝ enthält, ist <u>geschlossen</u>.

Ein Ast, der nicht geschlossen ist, ist <u>offen</u>.

Was genau sind Tableaux – Erweiterungen

Eine <u>Erweiterung T‘ von T</u> ist ein Tableau, für das gilt:

1) Jeder Knoten von T ist auch ein Knoten von T’.
2) Das Level für jeden Knoten von T ist dasselbe wie das Level des entsprechenden Knotens von T’.
3) Ein Knoten von T liegt über einem anderen Knoten von T genau dann, wenn der entsprechende eine Knoten von T’ über dem entsprechenden anderen Knoten von T’ liegt.
4) Wenn z ein Zweig von T ist, sind die Nachfolger von z in T auf dieselbe Weise geordnet wie in T’.
5) T‘ ist nicht T.

Eine <u>echte Erweiterung T_E‘ von T</u> ist eine Erweiterung T‘ von T, die T nur anhand der Tableau-Regeln (nächste Folie) erfolgt.

Was genau sind Tableaux – Regeln am Beispiel des Konditionals

Kommt ein Formelvorkommnis der Form $\ulcorner(\alpha \rightarrow \beta)\urcorner$ auf T vor, das auf einem offenen Ast liegt, gilt:

1) Die Knoten von T‘ sind die Elemente der Vereinigungsmenge der Knoten von T und $\{\ulcorner{\sim}\alpha\urcorner, \beta\}$.
2) Für alle Endknoten k_e von T gilt: l von T‘ ist die Vereinigungsmenge von l von T und $\{<\ulcorner{\sim}\alpha\urcorner, \ulcorner l(k_e)+1\urcorner>, <\beta, \ulcorner l(k_e)+1\urcorner>$.
3) Für alle Endknoten k_e von T gilt:
    a) k_e ist der Vorgänger von $\ulcorner{\sim}\alpha\urcorner$.
    b) k_e ist der Vorgänger von β.
4) Für alle Endknoten k_e von T gilt: θ von T‘ ist die Vereinigungsmenge von θ von T und $<k_e, <\ulcorner{\sim}\alpha\urcorner, \beta>>$.

Was genau sind Tableaux – ein Beispiel

$$T = \langle S, l, R, \theta \rangle$$

$T = \langle\{\sim p_1, \sim p_2, q, \sim q, p \rightarrow q\},$

$\{\langle p \rightarrow q, 1\rangle, \langle \sim p_1, 2\rangle, \langle \sim q, 3\rangle, \langle \sim p_2, 4\rangle, \langle q, 4\},$

$\{\langle p \rightarrow q, \sim p_1\rangle, \langle \sim p_1, \sim q\rangle, \langle \sim q, \sim p_2\rangle, \langle \sim q, q\rangle\},$

$\{\langle \sim q, \langle \sim p_2, q\rangle\rangle\}\rangle$

S	Knotenmenge
l	Level-Funktion
R	Vorgänger-Relation
θ	Nachfolger-Funktion (für Spaltknoten)

Was genau sind Tableaux – Darstellung

Wir stellen Tableaux dar, indem wir

(1) $\alpha_{V'}$ direkt mittig unter α_V schreiben gdw $\alpha_V R \alpha_{V'}$ und es kein $\alpha_{V''}$ gibt, sodass $l(\alpha_{V'}) = l(\alpha_{V''})$

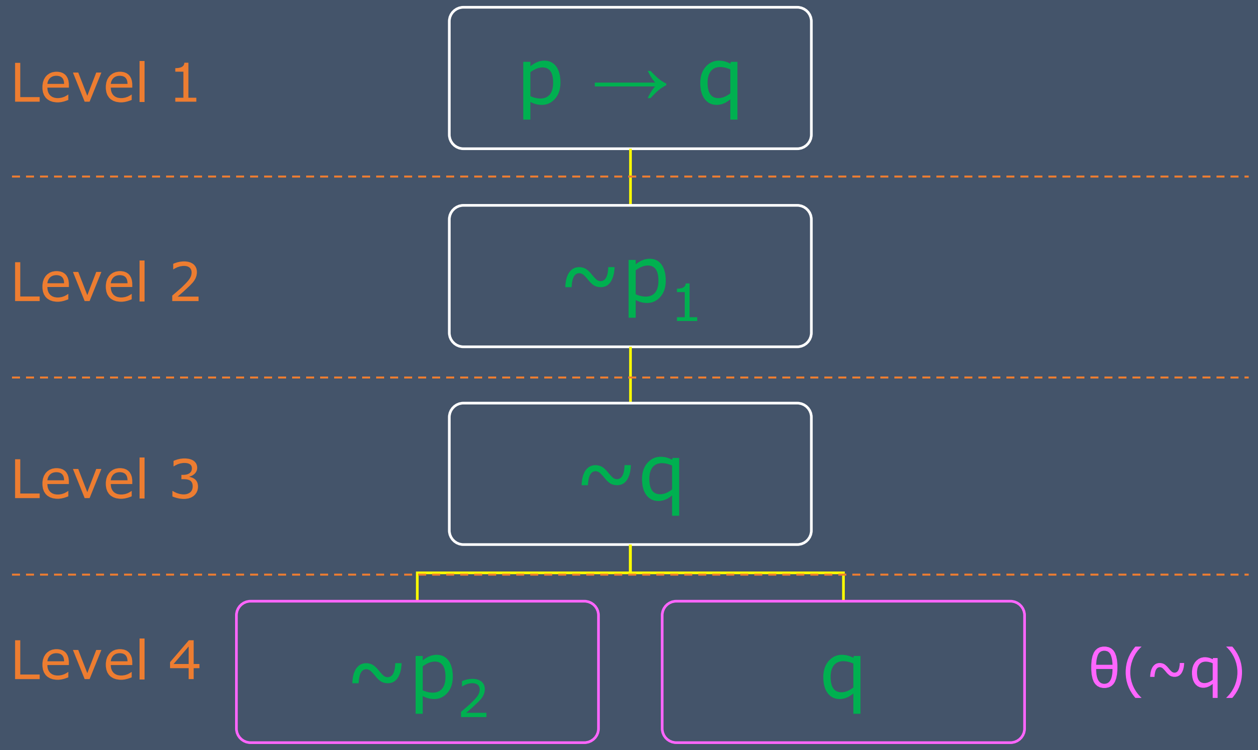
(2) $\alpha_{V'}$ direkt links unter α_V schreiben gdw $\alpha_V R \alpha_{V'}$ und es ein $\alpha_{V''}$ gibt, sodass $l(\alpha_{V'}) = l(\alpha_{V''})$ und $\theta(\alpha_V) = \langle \alpha_{V'}, \ldots \rangle$

(3) $\alpha_{V'}$ direkt rechts unter α_V schreiben gdw $\alpha_V R \alpha_{V'}$ und es ein $\alpha_{V''}$ gibt, sodass $l(\alpha_{V'}) = l(\alpha_{V''})$ und $\theta(\alpha_V) = \langle \ldots, \alpha_{V'} \rangle$

(4) α_V und $\alpha_{V'}$ mit einem Strich verbinden gdw $\alpha_V R \alpha_{V'}$.

Was genau sind Tableaux – Darstellung (Beispiel)

$T = \langle S, I, R, \theta \rangle$



Ast 1 (weil „~p“ die erste Komponente von $\theta(\sim q)$ ist.

Ast 2 (weil „q“ die zweite Komponente von $\theta(\sim q)$ ist.

Vollständigkeit – Erzeugte Modelle (Definition)

Sei b_S ein beliebiges offenes Astsegment eines Tableaus und A eine beliebige <u>atomare</u> Formel. Dann gilt:

$[\![...]\!]$ ist ein durch b_S erzeugtes Modell genau dann, wenn gilt:

Wenn A auf b_S ist, ist $[\![A]\!] = 1$.
Wenn ⌜~A⌝ auf b_S ist, ist $[\![A]\!] = 0$.

Vollständigkeitslemma (Definition)

Sei b ein beliebiger offener Ast eines Tableaus und α eine beliebige (möglicherweise komplexe) Formel. Sei ⟦…⟧ ein durch b erzeugtes Modell. Dann gilt:

Wenn α auf b ist, ist ⟦α⟧ = 1.

Wenn ⌜~α⌝ auf b ist, ist ⟦α⟧ = 0 (d. h. ⟦~α⟧ = 1).

→ Aus dem Vollständigkeitslemma ergibt sich, dass es für jeden offenen Ast eines Tableaus ein Modell gibt, das alle Formeln auf dem Ast wahr macht!

Was genau sind Tableaux – vollständige Tableaux

Eine <u>Anfangsliste</u> L für eine Menge M = $\{m_1, \ldots, m_n\}$ ist ein Tableau, sodass M=S.

Ein Widerspruchstableau ist ein Tableau, dessen Anfangsliste $\Gamma \cup \{\sim\beta\}$ ist.

Ein vollständiges Tableau T_V ist ein Tableau, für das gilt: Es gibt keine echte Erweiterung $T_{E'}$ für T_V.

Was sind Tableaux – der Grundgedanke

[I]f all the members of a finite set of truthfunctional sentences are satisfiable, i.e. true for some distribution of truth-values over their sentence letters, then they generate a finite open tree, while if they are not satisfiable they generate a closed one.

(Howson, Logic with Trees, S. 47)

WAS SIND TABLEAUX – GÜLTIGE SCHLÜSSE

If an inference is […] valid then the set [M] consisting of its premises and the negation of its conclusion is […] inconsistent […], and the Completeness Theorem implies that any [complete] tree generated by [M] will close. […] Thus the Completeness Theorem implies that if an inference is […] valid then there is a tree proof of its conclusion from its premises […].

(Howson, Logic with Trees, S. 52)

Literatur

Fitting, M. (1999). Introduction. In: Marcello D‘ Agostino et al.: *Handbook of Tableau Methods*, S. 1-44. Dordrecht: Springer.

Howson, C. (1997). *Logic with Trees*. *An Introduction to Symbolic Logic*. London: Routledge.

Lambert, K. (1997). *Free Logic. Their Foundations, Character and Some Applications Thereof*. Sankt Augustin: Academia.

Smullyan, R. (1955). Trees and Nest Structures. *Journal of Symbolic Logic 31 (3)*, S. 303-321.