3.16 Semantic Tableaux

Literature:

M. Fitting: First-Order Logic and Automated Theorem Proving, Springer-Verlag, New York, 1996, chapters 3, 6, 7.

R. M. Smullyan: First-Order Logic, Dover Publ., New York, 1968, revised 1995.

Like resolution, semantic tableaux were developed in the sixties, independently by Zbigniew Lis and Raymond Smullyan on the basis of work by Gentzen in the 30s and of Beth in the 50s.

Idea

Idea (for the propositional case):

A set $\{F \wedge G\} \cup N$ of formulas has a model if and only if $\{F \wedge G,\ F,\ G\} \cup N$ has a model.

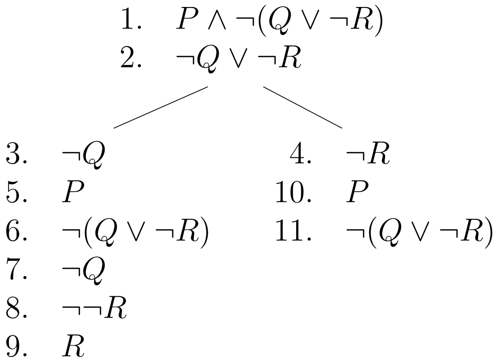
A set $\{F \vee G\} \cup N$ of formulas has a model if and only if $\{F \vee G,\ F\} \cup N$ or $\{F \vee G,\ G\} \cup N$ has a model.

(and similarly for other connectives).

To avoid duplication, represent sets as paths of a tree.

Continue splitting until two complementary formulas are found ⇒ inconsistency detected.

A Tableau for $\{P \wedge \neg(Q \vee \neg R),\ \neg Q \vee \neg R\}$



This tableau is not "maximal", however the first "path" is. This path is not "closed", hence the set $\{1, 2\}$ is satisfiable. (These notions will all be defined below.)

Properties

Properties of tableau calculi:

analytic: inferences according to the logical content of the symbols.

goal oriented: inferences operate directly on the goal to be proved (unlike, e. g., ordered resolution).

global: some inferences affect the entire proof state (set of formulas), as we will see later.

Propositional Expansion Rules

Expansion rules are applied to the formulas in a tableau and expand the tableau at a leaf. We append the conclusions of a rule (horizontally or vertically) at a *leaf*, whenever the premise of the expansion rule matches a formula appearing *anywhere* on the path from the root to that leaf.

Negation Elimination

$$\frac{\neg\neg F}{F} \qquad \frac{\neg\top}{\bot} \qquad \frac{\neg\bot}{\top}$$

α-Expansion

(for formulas that are essentially conjunctions: append subformulas α_1 and α_2 one on top of the other)

$$\begin{array}{l} \underline{\alpha} \\ \alpha_1 \\ \alpha_2 \end{array}$$

β-Expansion

(for formulas that are essentially disjunctions:
append β_1 and β_2 horizontally, i. e., branch into β_1 and β_2)

$$\frac{\beta}{\beta_1 \mid \beta_2}$$

Classification of Formulas

conjunctive			disjunctive		
α	α_1	α_2	β	β_1	β_2
$X \wedge Y$	X	Y	$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$\neg(X \vee Y)$	$\neg X$	$\neg Y$	$X \vee Y$	X	Y
$\neg(X \rightarrow Y)$	X	$\neg Y$	$X \rightarrow Y$	$\neg X$	Y

We assume that the binary connective $\leftrightarrow$ has been eliminated in advance.

Tableaux: Notions

A *semantic tableau* is a marked (by formulas), finite, unordered tree and inductively defined as follows: Let $\{F_1, \dots, F_n\}$ be a set of formulas.

(i) The tree consisting of a single path

$$\begin{array}{l} F_1 \\ \vdots \\ F_n \end{array}$$

is a tableau for $\{F_1, \dots, F_n\}$. (We do not draw edges if nodes have only one successor.)

(ii) If T is a tableau for $\{F_1, \dots, F_n\}$ and if T' results from T by applying an expansion rule then T' is also a tableau for $\{F_1, \dots, F_n\}$.

A *path* (from the root to a leaf) in a tableau is called *closed*, if it either contains $\bot$, or else it contains both some formula F and its negation $\neg F$. Otherwise the path is called *open*.

A tableau is called *closed*, if all paths are closed.

A *tableau proof* for F is a closed tableau for $\{\neg F\}$.

A path P in a tableau is called *maximal*, if for each non-atomic formula F on P there exists a node in P at which the expansion rule for F has been applied.

In that case, if F is a formula on P, P also contains:

(i) F_1 and F_2, if F is a α-formula,

(ii) F_1 or F_2, if F is a β-formula, and

(iii) F', if F is a negation formula, and F' the conclusion of the corresponding elimination rule.

A tableau is called *maximal*, if each path is closed or maximal.

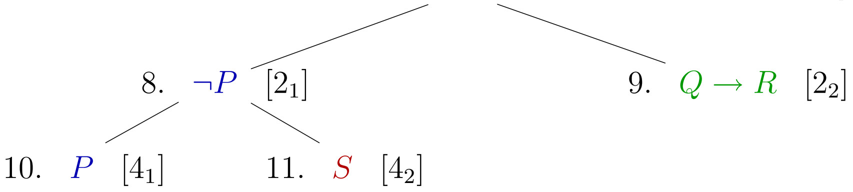
A tableau is called *strict*, if for each formula the corresponding expansion rule has been applied at most once on each path containing that formula.

A tableau is called *clausal*, if each of its formulas is a clause.

A Sample Proof

One starts out from the negation of the formula to be proved.

1.	$\neg[(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \vee S) \rightarrow ((Q \rightarrow R) \vee S))]$	
2.	$(P \rightarrow (Q \rightarrow R))$	$[1_1]$
3.	$\neg((P \vee S) \rightarrow ((Q \rightarrow R) \vee S))$	$[1_2]$
4.	$P \vee S$	$[3_1]$
5.	$\neg((Q \rightarrow R) \vee S))$	$[3_2]$
6.	$\neg(Q \rightarrow R)$	$[5_1]$
7.	$\neg S$	$[5_2]$



There are three paths, each of them closed.

Properties of Propositional Tableaux

We assume that T is a tableau for $\{F_1, \ldots, F_n\}$.

Theorem 3.45 $\{F_1, \ldots, F_n\}$ *satisfiable* $\Leftrightarrow$ *some path (i. e., the set of its formulas) in* T *is satisfiable.*

Proof. By induction over the structure of T. ☐

Corollary 3.46 T *closed* $\Rightarrow$ $\{F_1, \ldots, F_n\}$ *unsatisfiable*

Theorem 3.47 *Let* T *be a strict propositional tableau. Then* T *is finite.*

Proof. New formulas resulting from expansion are either $\bot$, $\top$ or subformulas of the expanded formula. By strictness, on each path a formula can be expanded at most once. Therefore, each path is finite, and a finitely branching tree with finite paths is finite by Lemma 1.8. ☐

Conclusion: Strict and maximal tableaux can be effectively constructed.

Refutational Completeness

Theorem 3.48 *Let P be a maximal, open path in a tableau. Then set of formulas on P is satisfiable.*

Proof. (The theorem holds for arbitrary tableaux, but in this proof we consider only the case of a clausal tableau. The full proof can be found, e.g., in Fitting 1996.)

Let N be the set of formulas on P. As P is open, $\bot$ is not in N. Let $C \vee A$ and $D \vee \neg A$ be two resolvable clauses in N. One of the two subclauses C or D, C say, is not empty, as otherwise P would be closed. Since P is maximal, in P the β-rule was applied on $C \vee A$. Therefore, P (and N) contains a proper subclause of $C \vee A$, and hence $C \vee A$ is redundant w. r. t. N. By the same reasoning, if N contains a clause that can be factored, that clause must be redundant w. r. t. N. In other words, N is saturated up to redundancy w. r. t. Res(olution). Now apply Theorem 3.17 to prove satisfiability of N. $\square$

Theorem 3.49 $\{F_1, \ldots, F_n\}$ *satisfiable* $\Leftrightarrow$ *there exists no closed strict tableau for* $\{F_1, \ldots, F_n\}$.

Proof. One direction is clear by Theorem 3.45. For the reverse direction, let T be a strict, maximal tableau for $\{F_1, \ldots, F_n\}$ and let P be an open path in T. By the previous theorem, the set of formulas on P, and hence by Theorem 3.45 the set $\{F_1, \ldots, F_n\}$, is satisfiable. $\square$

Consequences

The validity of a propositional formula F can be established by constructing a strict, maximal tableau for $\{\neg F\}$:

- T closed $\Leftrightarrow$ F valid.
- It suffices to test complementarity of paths w. r. t. atomic formulas (cf. reasoning in the proof of Theorem 3.48).
- Which of the potentially many strict, maximal tableaux one computes does not matter. In other words, tableau expansion rules can be applied don't-care nondeterministically ("*proof confluence*").
- The expansion strategy, however, can have a dramatic impact on tableau size.
- Since it is sufficient to saturate paths w. r. t. ordered resolution (up to redundancy), tableau expansion rules can be even more restricted, in particular by certain ordering constraints.

A Variant of the β-Rule

Since $F \vee G \models\!\!\!\!\!= F \vee (G \wedge \neg F)$, the β expansion rule

$$\frac{\beta}{\beta_1 \mid \beta_2}$$

can be replaced by the following variant:

$$\frac{\beta}{\begin{array}{c|l} \beta_1 & \beta_2 \\ & \neg\beta_1 \end{array}}$$

The variant β-rule can lead to much shorter proofs, but it is not always beneficial.

In general, it is most helpful if $\neg\beta_1$ can be at most (iteratively) α-expanded.