commutative, they are equivalent. One or two columns in the truth table for the two subformulas? Again, saving a column is beneficial but in general, detecting equivalence of two subformulas may become as difficult as checking whether the overall formula is valid. A compromise, often performed in practice, are normal forms that guarantee that certain occurrences of equivalent subformulas can be found in polynomial time. For the running example, we can simply assume some ordering on the propositional variables and assume that for a disjunction of two propositional variables, the smaller variable always comes first. So if $P < Q$ then the normal form of $P \lor Q$ and $Q \lor P$ is in fact $P \lor Q$.

C

In practice, nobody uses truth tables as a reasoning procedure. Worst case, computing a truth table for checking the status of a formula ϕ requires $O(2^n)$ steps, where n is the number of different propositional variables in ϕ. But this is actually not the reason why the procedure is impractical, because the worst case behavior of all other procedures for propositional logic known today is also of exponential complexity. So why are truth tables not a good procedure? The answer is: because they do not adapt to the inherent structure of a formula. The reasoning mechanism of a truth table for two formulas ϕ and ψ sharing the same propositional variables is exactly the same: we enumerate all valuations. However, if ϕ is, e.g., of the form $\phi = P \land \phi'$ and we are interested in the satisfiability of ϕ, then ϕ can only become true for a valuation $\mathcal{A}$ with $\mathcal{A}(P) = 1$. Hence, 2^{n-1} rows of ϕ's truth table are superfluous. All procedures I will introduce in the sequel, automatically detect this (and further) specific structures of a formula and use it to speed up the reasoning process.

2.4 Propositional Tableaux

Like resolution, semantic tableaux were developed in the sixties, independently by Lis [33] and Smullyan [49] on the basis of work by Gentzen in the 30s [23] and of Beth [8] in the 50s. For an at that time state of the art overview consider Fitting's book [21].

In contrast to the calculi introduced in subsequent sections, semantic tableau does not rely on a normal form of input formulas but actually applies to any propositional formula. The formulas are divided into α- and β-formulas, where intuitively an α formula represents an (implicit) conjunction and a β formula an (implicit) disjunction.

Definition 2.4.1 (α-, β-Formulas)**.** A formula ϕ is called an α-formula if ϕ is a formula $\neg\neg\phi_1$, $\phi_1 \land \phi_2$, $\phi_1 \leftrightarrow \phi_2$, $\neg(\phi_1 \lor \phi_2)$, or $\neg(\phi_1 \rightarrow \phi_2)$. A formula ϕ is called a β-formula if ϕ is a formula $\phi_1 \lor \phi_2$, $\phi_1 \rightarrow \phi_2$, $\neg(\phi_1 \land \phi_2)$, or $\neg(\phi_1 \leftrightarrow \phi_2)$.

A common property of α-, β-formulas is that they can be decomposed into direct descendants representing (modulo negation) subformulas of the respective formulas. Then an α-formula is valid iff all its descendants are valid and a β-formula is valid iff one of its descendants is valid. Therefore, the literature uses both the notions semantic tableaux and analytic tableaux.

Definition 2.4.2 (Direct Descendant)**.** Given an α- or β-formula ϕ, Figure 2.4 shows its direct descendants.

Duplicating ϕ for the α-descendants of $\neg\neg\phi$ is a trick for conformity. Any propositional formula is either an α-formula or a β-formula or a literal.

Proposition 2.4.3. For any valuation $\mathcal{A}$: (i) if ϕ is an α-formula then $\mathcal{A}(\phi) = 1$ iff $\mathcal{A}(\phi_1) = 1$ and $\mathcal{A}(\phi_2) = 1$ for its descendants ϕ_1, ϕ_2. (ii) if ϕ is a β-formula then $\mathcal{A}(\phi) = 1$ iff $\mathcal{A}(\phi_1) = 1$ or $\mathcal{A}(\phi_2) = 1$ for its descendants ϕ_1, ϕ_2.

The tableau calculus operates on states that are sets of sequences of formulas. Semantically, the set represents a disjunction of sequences that are interpreted as conjunctions of the respective formulas. A sequence of formulas $(\phi_1, \ldots, \phi_n)$ is called *closed* if there are two formulas ϕ_i and ϕ_j in the sequence where $\phi_i = \text{comp}(\phi_j)$. A state is *closed* if all its formula sequences are closed. A state actually represents a tree and this tree is called a tableau in the literature. So if a state is closed, the respective tree, the tableau is closed too. The tableau calculus is a calculus showing unsatisfiability of a formula. Such calculi are called *refutational* calculi. Later on soundness and completeness of the calculus imply that a formula ϕ is valid iff the rules of tableau produce a closed state starting with $N = \{(\neg\phi)\}$.

A formula ϕ occurring in some sequence is called *open* if in case ϕ is an α-formula not both direct descendants are already part of the sequence and if it is a β-formula none of its descendants is part of the sequence.

α-Expansion $N \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n)\} \Rightarrow_{\text{T}} N \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_1, \psi_2)\}$
provided ψ is an open α-formula, ψ_1, ψ_2 its direct descendants and the sequence is not closed.

β-Expansion $N \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n)\} \Rightarrow_{\text{T}} N \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_1)\} \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_2)\}$
provided ψ is an open β-formula, ψ_1, ψ_2 its direct descendants and the sequence is not closed.

For example, consider proving validity of the formula $(P \wedge \neg(Q \vee \neg R)) \rightarrow (Q \wedge R)$. Applying the tableau rules generates the following derivation:

$$\begin{array}{l}
\{(\neg[(P \wedge \neg(Q \vee \neg R)) \rightarrow (Q \wedge R)])\} \\
\quad \alpha\text{-Expansion} \Rightarrow_{\text{T}}^{*} \{(\neg[(P \wedge \neg(Q \vee \neg R)) \rightarrow (Q \wedge R)], \\
\quad\quad P \wedge \neg(Q \vee \neg R), \neg(Q \wedge R), P, \neg(Q \vee \neg R), \neg Q, \neg\neg R, R)\} \\
\quad \beta\text{-Expansion} \Rightarrow_{\text{T}} \{(\neg[(P \wedge \neg(Q \vee \neg R)) \rightarrow (Q \wedge R)], \\
\quad\quad P \wedge \neg(Q \vee \neg R), \neg(Q \wedge R), P, \neg(Q \vee \neg R), \neg Q, \neg\neg R, R, \neg Q), \\
\quad\quad (\neg[(P \wedge \neg(Q \vee \neg R)) \rightarrow (Q \wedge R)], \\
\quad\quad P \wedge \neg(Q \vee \neg R), \neg(Q \wedge R), P, \neg(Q \vee \neg R), \neg Q, \neg\neg R, R, \neg R)\}
\end{array}$$

The state after β-expansion is final, i.e., no more rule can be applied. The first sequence is not closed, whereas the second sequence is closed because it contains R and $\neg R$. Thus, the formula is not valid but satisfiable. A tree representation, where common formulas of sequences are shared, can be found in Figure 2.5. This is the traditional way of tableau presentation.

Theorem 2.4.4 (Propositional Tableau is Sound)**.** If for a formula ϕ the tableau calculus computes $\{(\neg\phi)\} \Rightarrow_{\mathrm{T}}^{*} N$ and N is closed, then ϕ is valid.

Proof. It is sufficient to show the following: (i) if N is closed then the disjunction of the conjunction of all sequence formulas is unsatisfiable (ii) the two tableau rules preserve satisfiability.

Part (i) is obvious: if N is closed all its sequences are closed. A sequence is closed if it contains a formula and its negation. The conjunction of two such formulas is unsatisfiable.

Part (ii) is shown by induction on the length of the derivation and then by a case analysis for the two rules. α-Expansion: for any valuation $\mathcal{A}$ if $\mathcal{A}(\psi) = 1$ then $\mathcal{A}(\psi_1) = \mathcal{A}(\psi_2) = 1$. β-Expansion: for any valuation $\mathcal{A}$ if $\mathcal{A}(\psi) = 1$ then $\mathcal{A}(\psi_1) = 1$ or $\mathcal{A}(\psi_2) = 1$ (see Proposition 2.4.3). □

Theorem 2.4.5 (Propositional Tableau Terminates)**.** Starting from a start state $\{(\phi)\}$ for some formula ϕ, the relation $\Rightarrow_{\mathrm{T}}^{+}$ is well-founded.

Proof. Take the two-folded multiset extension of the lexicographic extension of $>$ on the naturals to triples (n, k, l) generated by the a measure μ. It is first defined on formulas by $\mu(\phi) := (n, k, l)$ where n is the number of equivalence symbols in ϕ, k is the sum of all disjunction, conjunction, implication symbols in ϕ and l is $|\phi|$. On sequences $(\phi_1, \ldots, \phi_n)$ the measure is defined to deliver a multiset by $\mu((\phi_1, \ldots, \phi_n)) := \{t_1, \ldots, t_n\}$ where $t_i = \mu(\phi_i)$ if ϕ_i is open in the sequence and $t_i = (0, 0, 0)$ otherwise. Finally, μ is extended to states N by computing the multiset $\mu(N) := \{\mu(s) \mid s \in N\}$.

Note, that α-, as well as β-expansion strictly extend sequences. Once a formula is closed in a sequence by applying an expansion rule, it remains closed forever in the sequence.

An α-expansion on a formula $\psi_1 \wedge \psi_2$ on the sequence $(\phi_1, \ldots, \psi_1 \wedge \psi_2, \ldots, \phi_n)$ results in $(\phi_1, \ldots, \psi_1 \wedge \psi_2, \ldots, \phi_n, \psi_1, \psi_2)$. It needs to be shown $\mu((\phi_1, \ldots, \psi_1 \wedge \psi_2, \ldots, \phi_n)) >_{\mathrm{mul}} \mu((\phi_1, \ldots, \psi_1 \wedge \psi_2, \ldots, \phi_n, \psi_1, \psi_2))$. In the second sequence $\mu(\psi_1 \wedge \psi_2) = (0, 0, 0)$ because the formula is closed. For the triple (n, k, l) assigned by μ to $\psi_1 \wedge \psi_2$ in the first sequence, it holds $(n, k, l) >_{\mathrm{lex}} \mu(\psi_1)$, $(n, k, l) >_{\mathrm{lex}} \mu(\psi_2)$ and $(n, k, l) >_{\mathrm{lex}} (0, 0, 0)$, the former because the ψ_i are subformulas and the latter because $l \neq 0$. This proves the case.

A β-expansion on a formula $\psi_1 \vee \psi_2$ on the sequence $(\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n)$ results in $(\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n, \psi_1)$, $(\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n, \psi_2)$. It needs to be shown $\mu((\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n)) >_{\mathrm{mul}} \mu((\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n, \psi_1))$ and $\mu((\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n)) >_{\mathrm{mul}} \mu((\phi_1, \ldots, \psi_1 \vee \psi_2, \ldots, \phi_n, \psi_2))$. In the derived sequences $\mu(\psi_1 \vee \psi_2) = (0, 0, 0)$ because the formula is closed. For the triple (n, k, l) assigned by μ to $\psi_1 \vee \psi_2$ in the starting sequence, it holds $(n, k, l) >_{\mathrm{lex}}$

$\mu(\psi_1)$, $(n,k,l) >_{\text{lex}} \mu(\psi_2)$ and $(n,k,l) >_{\text{lex}} (0,0,0)$, the former because the ψ_i are subformulas and the latter because $l \neq 0$. This proves the case. □

Theorem 2.4.6 (Propositional Tableau is Complete)**.** If ϕ is valid, tableau computes a closed state out of $\{(\neg\phi)\}$.

Proof. If ϕ is valid then $\neg\phi$ is unsatisfiable. Now assume after termination the resulting state and hence at least one sequence is not closed. For this sequence consider a valuation $\mathcal{A}$ consisting of the literals in the sequence. By assumption there are no opposite literals, so $\mathcal{A}$ is well-defined. I prove by contradiction that $\mathcal{A}$ is a model for the sequence. Assume it is not. Then there is a minimal formula in the sequence, with respect to the ordering on triples considered in the proof of Theorem 2.4.5, that is not satisfied by $\mathcal{A}$. By definition of $\mathcal{A}$ the formula cannot be a literal. So it is an α-formula or a β-formula. In all cases at least one descendant formula is contained in the sequence, is smaller than the original formula, false in $\mathcal{A}$ (Proposition 2.4.3) and hence contradicts the assumption. Therefore, $\mathcal{A}$ satisfies the sequence contradicting that $\neg\phi$ is unsatisfiable. □

Corollary 2.4.7 (Propositional Tableau generates Models)**.** Let ϕ be a formula, $\{(\phi)\} \Rightarrow^*_{\text{T}} N$ and $s \in N$ be a sequence that is not closed and neither α-expansion nor β-expansion are applicable to s. Then the literals in s form a (partial) valuation that is a model for ϕ.

Proof. See Exercise **??**. □

Consider the example tableau shown in Figure 2.5. The open first branch corresponds to the valuation $\mathcal{A} = \{P, R, \neg Q\}$ which is a model of the formula $\neg[(P \land \neg(Q \lor \neg R)) \rightarrow (Q \land R)]$.

The tableau calculus naturally evolves out of the semantics of the operators. However, from a proof search and proof length point of view it has severe deficits. Consider, for example, the abstract tableau in Figure 2.6. Let's assume it is closed. Let's further assume that the closedness does not depend on the K_j, K'_j literals. Then there is an exponentially smaller closed tableau for the formula that consists of picking exactly one of the identical L_i, L'_i subtrees. The calculus does not "learn" from the fact that closedness does not depend on the K_j, K'_j literals. Actually, this can be overcome and one way of looking at CDCL, Section 2.9, is to consider it as a solution to the problem of unnecessary repetitions of already closed branches. Concerning proof length, there are clause sets where an exponential blow up compared to resolution, Section 2.6, or CDCL, Section 2.9, cannot be prevented. For example, on a clause set where every clause rules out exactly one valuation of n variables, the shortest resolution proof is exponentially shorter than the shortest tableau proof. In addition, the resolution proof can be found in a deterministic way by simplification, see Example 2.6.4. For two variables the respective clause set is $(P \lor Q) \land (P \lor \neg Q) \land (\neg P \lor Q) \land (\neg P \lor \neg Q)$.