$Q(a, a) \lor \neg P(a)$
$Q(b, b) \lor \neg P(b)$
$Q(g(a), g(a)) \lor \neg P(g(a))$
$\ldots$
$\neg P(a) \lor P(g(a))$
$\neg P(b) \lor P(g(b))$
$\neg P(g(a)) \lor P(g(g(a)))$
$\ldots\}$
is satisfiable. For example by the Herbrand models
$I_1 := \emptyset$
$I_2 := \{P(b), Q(b, b), P(g(b)), Q(g(b), g(b)), \ldots\}$

3.6 First-Order Tableau

The different versions of tableau for first-order logic differ in particular in the treatment of variables by the tableau rules. The first variant I consider is standard first-order tableau where variables are instantiated by ground terms. For this section, if not stated otherwise, all considered formulas are closed and do not contain equations.

Definition 3.6.1 (γ-,δ-Formulas)**.** A formula ϕ is called a γ-formula if ϕ is a formula $\forall x_S.\psi$ or $\neg\exists x_S.\psi$. A formula ϕ is called a δ-formula if ϕ is a formula $\exists x_S.\psi$ or $\neg\forall x_S.\psi$.

Definition 3.6.2 (Direct Standard Tableau Descendant)**.** Given a γ- or δ-formula ϕ, Figure 3.1 shows its direct descendants.

As a proper extension to propositional tableau (Section 2.4) the first-order tableau calculus operates on states that are sets of pairs of a sequence of formulas and a set of constants. Semantically, the set of pairs represents a disjunction of the respective sequences that are interpreted as conjunctions of the respective formulas. The sets of constants is needed to ensure ground terms for all potential γ-instantiations in the sequence.

A sequence of formulas $(\phi_1, \phi_2, \ldots)$ 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. Given a formula ϕ, the initial tableau for ϕ is $\{((\neg\phi), \{c_1, \ldots, c_k\})\}$ where the c_i are constants for each sort S such that ϕ contains a variable of sort S but no ground term t in ϕ has sort S.

A first-order formula ϕ occurring in some sequence in N of a pair (M, J) is called *open* if in case ϕ is an α-formula not both direct descendants are already part of M, if it is a β-formula none of its descendants is part of M, if it is a δ-formula no direct descendant is part of M, and if it is a γ-formula not all direct descendants are part of M where only ground terms are considered

γ	Descendant $\gamma(t)$
$\forall x_S.\psi$	$\psi\{x_S \mapsto t\}$
$\neg\exists x_S.\psi$	$\neg\psi\{x_S \mapsto t\}$
	for a ground term $t \in T_S(\Sigma)$

δ	Descendant $\delta(c)$
$\exists x_S.\psi$	$\psi\{x_S \mapsto c\}$
$\neg\forall x_S.\psi$	$\neg\psi\{x_S \mapsto c\}$
	for a fresh constant $c \in T_S(\Sigma)$

Figure 3.1: γ- and δ-Formulas

that can be built from constants J and function symbols in M such that their depth is bounded by the maximal term depth occurring in M. In general, the number of overall γ descendants of a tableau cannot be limited for a successful tableau proof. A tableau can become infinite. This motivates the below notion of saturation.

For the standard first-order tableau rules to make sense "enough" Skolem constants (ground terms) are needed in the signature, e.g., countably infinitely many for each sort. Then the tableau rules are

α-Expansion $N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n), J)\} \Rightarrow_{\mathrm{FT}} N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_1, \psi_2), J)\}$

provided ψ is an open α-formula, ψ_1, ψ_2 its direct descendants and the sequence is not closed.

β-Expansion $N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n), J)\} \Rightarrow_{\mathrm{FT}} N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_1), J)\} \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n, \psi_2), J)\}$

provided ψ is an open β-formula, ψ_1, ψ_2 its direct descendants and the sequence is not closed.

γ-Expansion $N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n), J)\} \Rightarrow_{\mathrm{FT}} N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n, \psi'), J)\}$

provided ψ is a γ-formula, ψ' a $\gamma(t)$ descendant where t is a ground term in the signature of the sequence and J, the depth of t is bounded by the maximal term depth in $(\phi_1, \ldots, \psi, \ldots, \phi_n)$ and the sequence is not closed.

δ-Expansion $N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n), J)\} \Rightarrow_{\mathrm{FT}} N \uplus \{(\phi_1, \ldots, \psi, \ldots, \phi_n, \psi'), J \cup \{c\})\}$

provided ψ is an open δ-formula, ψ' a $\delta(c)$ descendant where c is a fresh constant and the sequence is not closed.

where the α-, and β-expansion rules are copies of the propositional rules, see Section 2.4, adjusted to the extended state format. The α-, and β-expansion rules don't manipulate the set of ground terms of a state. Note that for a particular γ-expansion the number of potential ground terms that can be used for instantiation is always finite.

A possibly infinite tableau derivation $s_0 \Rightarrow_{\mathrm{FT}} s_1 \Rightarrow_{\mathrm{FT}} \ldots$ is called *saturated* if for all its open sequences M_i of some pair $(M_i, J_i) \in s_i$ where not all successor sequences of M_i are closed and all formulas ϕ occurring in M_i, there is an index $j > i$ and some pair $(M_j, J_j) \in s_j$, M_i is a prefix of M_j, if in case ϕ is an α-formula then both direct descendants are part of M_j, if it is a β-formula then one of its descendants is part of M_j, if it is a δ-formula then one direct descendant is part of M_j , and if it is γ-formula then all direct descendants from ground terms that can be built from function symbols in J_i and M_i such that their depth is bounded by the maximal term depth in M_i are part of the sequence M_j.

In general, the γ rule has to be applied several times to the same formula in order to close a tableau. For instance, constructing a closed tableau from initial state

$$\{((\forall x_S.(P(x_S) \rightarrow P(f(x_S))),\ P(b),\ \neg P(f(f(b)))), \emptyset)\}$$

is impossible without applying γ-Expansion twice to $\forall x_S.(P(x_S) \rightarrow P(f(x_S)))$ on some branch, where $b :\rightarrow S$, $f : S \rightarrow S$ and $P \subseteq S$. Below is the derivation of a closed tableau where I only show the added formulas and often abbreviate the parent sequence with an indexed M.

$$\{((\forall x_S.(P(x_S) \rightarrow P(f(x_S))),\ P(b),\ \neg P(f(f(b)))), \emptyset)$$
$$\Rightarrow^{\gamma}_{\mathrm{FT}} \quad \{((M_1, P(b) \rightarrow P(f(b))), \emptyset)$$
$$\Rightarrow^{\beta}_{\mathrm{FT}} \quad \{((M_2, \neg P(b)), \emptyset), ((M_2, P(f(b))), \emptyset)\}$$

the sequence $(M_2, \neg P(b))$ is closed, so from now on omitted

$$\Rightarrow^{\gamma}_{\mathrm{FT}} \quad \{((M_2, P(f(b)), P(f(b)) \rightarrow P(f(f(b)))), \emptyset)\}$$
$$\Rightarrow^{\beta}_{\mathrm{FT}} \quad \{((M_3, \neg P(f(b))), \emptyset), ((M_3, P(f(f(b)))), \emptyset)\}$$

now the tableau is closed

where $M_1 = (\forall x_S.(P(x_S) \rightarrow P(f(x_S))),\ P(b),\ \neg P(f(f(b))))$, $M_2 = M_1, (P(b) \rightarrow P(f(b)))$ and $M_3 = M_2, (P(f(b)), P(f(b)) \rightarrow P(f(f(b))))$. Note that at any point of the derivation, the only ground terms to be used by γ-Expansion are $b, f(b), f(f(b))$ because of the depth restriction, where the first and second are needed in the derivation and instantiation with $f(f(b))$ produces a tautology in the presence of $\neg P(f(f(b))))$. No extra constants are needed because the formula already contains a ground term of sort S.

Example 3.6.3 (Extra Constants)**.** Consider the valid formula

$$\phi = \neg(\forall x_S.\exists y_S.(R(x_S, y_S) \vee P(b))\ \wedge\ \forall x_S, y_S.\neg R(x_S, y_S)\ \wedge\ \neg P(b))$$

where $b :\rightarrow T$, $P \subseteq T$, and $R \subseteq S \times S$. The initial tableau, with double negation removed is

$$\{((\forall x_S.\exists y_S.(R(x_S, y_S) \vee P(b))\ \wedge\ \forall x_S, y_S.\neg R(x_S, y_S)\ \wedge\ \neg P(b)), \{c\})\}$$

where $c :\rightarrow S$.

$\{((\text{comp}(\phi)), \{c\})\}$
$\Rightarrow^{\alpha,*}_{\text{FT}}$ $\{((\text{comp}(\phi), \forall x_S.\exists y_S.(R(x_S, y_S) \lor P(b)), \forall x_S, y_S.\neg R(x_S, y_S), \neg P(b)), \{c\})$
$\Rightarrow^{\gamma}_{\text{FT}}$ $\{((M_1, \exists y_S.(R(c, y_S) \lor P(b))), \{c\})$
$\Rightarrow^{\delta}_{\text{FT}}$ $\{((M_1, \exists y_S.(R(c, y_S) \lor P(b)), R(c, d) \lor P(b)), \{c, d\})$
where $d :\to S$ is fresh
$\Rightarrow^{\beta}_{\text{FT}}$ $\{((M_2, R(c, d)), \{c, d\}), ((M_2, P(b)), \{c, d\})\}$
the sequence $(M_2, P(b))$ is closed, so from now on omitted
$\Rightarrow^{\gamma,*}_{\text{FT}}$ $\{((M_2, R(c, d), \forall y_S.\neg R(c, y_S), \neg R(c, d)), \{c, d\})$
now the tableau is closed

Example 3.6.4 (Infinite Derivations)**.** Consider the satisfiable formula

$$\phi = \forall x_S.(P(x_S) \to P(f(x_S))) \land P(a)$$

where $a :\to S$, $P \subseteq S$, and $f : S \to S$. The tableau calculus does not terminate on this formula:

$\{((\forall x_S.(P(x_S) \to P(f(x_S))) \land P(a)), \emptyset)\}$
$\Rightarrow^{\alpha}_{\text{FT}}$ $\{((\phi, \forall x_S.(P(x_S) \to P(f(x_S))), P(a)), \emptyset)\}$
$\Rightarrow^{\gamma}_{\text{FT}}$ $\{((\phi, \forall x_S.(P(x_S) \to P(f(x_S))), P(a), P(a) \to P(f(a))), \emptyset)\}$
$\Rightarrow^{\beta}_{\text{FT}}$ $\{((M_1, \neg P(a)), \emptyset), ((M_1, P(f(a))), \emptyset)\}$
the sequence $(M_1, \neg P(a))$ is closed, so from now on omitted
$\Rightarrow^{\gamma}_{\text{FT}}$ $\{((M_1, P(f(a)), P(f(a)) \to P(f(f(a)))), \emptyset)\}$
$\ldots$

The infinite branch computes the infinite Herbrand model $P(f^i(a))$, see below.

Theorem 3.6.5 (Standard First-Order Tableau is Sound)**.** If for a closed formula ϕ the tableau calculus derives $\{((\neg\phi), J)\} \Rightarrow^{*}_{\text{FT}} N$ and N is closed, then ϕ is valid.

Proof. A branch of the tableau, i.e., the sequence out of a pair $((\phi_1, \ldots, \phi_n), J')$ such that $\phi_j = \text{comp}(\phi_i)$ for some i, j is unsatisfiable. If all tableau rules are sound, i.e., preserve satisfiability, then if all branches are closed the formula of the start state is unsatisfiable and its negation valid. So it remains to show soundness for the tableau rules. For α- and β-Expansion this is already shown in Theorem 2.4.4. It remains to show soundness of γ- and δ-Expansion.

γ-Expansion: let $\mathcal{A}$ be a model for $(\phi_1, \ldots, \psi, \ldots, \phi_n)$, i.e., $\mathcal{A} \models \phi_1 \land \ldots \land \psi \land \ldots \land \phi_n$. In particular, $\mathcal{A} \models \psi$ where $\psi = \forall x_S.\chi$. So for any valuation $\beta[x_s \mapsto a]$, $a \in S^{\mathcal{A}}$ it holds $\mathcal{A}, \beta[x_s \mapsto a] \models \chi$. Hence, $\mathcal{A}, \beta[x_s \mapsto t^{\mathcal{A}}] \models \chi$ for any ground term t of sort S and therefore $\mathcal{A} \models \psi'$, for any $\gamma(t)$ descendant ψ'. The case $\psi = \neg\exists x_S.\chi$ can be shown analogously.

δ-Expansion: let $\mathcal{A}$ be a model for $(\phi_1, \ldots, \psi, \ldots, \phi_n)$, i.e., $\mathcal{A} \models \phi_1 \wedge \ldots \wedge \psi \wedge \ldots \wedge \phi_n$. In particular, $\mathcal{A} \models \psi$ where $\psi = \exists x_S.\chi$. So there is a valuation $\beta[x_s \mapsto a]$, for some $a \in S^{\mathcal{A}}$ such that $\mathcal{A}, \beta[x_s \mapsto a] \models \chi$. Now construct an interpretation $\mathcal{A}'$ that is identical to $\mathcal{A}$, except $c^{\mathcal{A}'} = a$ for the fresh constant c of the $\delta(c)$ descendant. The constant c does not occur in $(\phi_1, \ldots, \psi, \ldots, \phi_n)$, so $\mathcal{A}' \models \phi_1 \wedge \ldots \wedge \psi \wedge \ldots \wedge \phi_n$ and by construction $\mathcal{A}' \models \psi'$, for the $\delta(c)$ descendant ψ'. The case $\psi = \neg\forall x_S.\chi$ can be shown analogously. □

Completeness of a first-order calculus is more complicated than in propositional logic, because it requires the consideration of infinite derivations. For the tableau calculus I prove that if a possibly infinite saturated derivation contains an open branch, then this branch constitutes a model for the formulas in the sequence. This idea is pretty old and goes back to Hintikka and Smullyan's completeness proof [38].

Definition 3.6.6 (Hintikka Set)**.** Let M be a possibly infinite set of closed first-order formulas such that for each variable x_S occurring in M, there is at least one ground term t of sort S occurring in M. M is a *Hintikka set* if for all formulas $\phi \in M$:

1. if ϕ is a literal, then $\operatorname{comp}(\phi) \notin M$
2. if ϕ is an α formula then α_1 and α_2 are in M
3. if ϕ is a β formula then β_1 or β_2 is in M
4. if ϕ is a γ formula then for all ground terms t of correct sort occurring in M the formula $\gamma(t)$ is in M
5. if ϕ is a δ formula then $\delta(c)$ is in M for some constant c

Lemma 3.6.7 (Hintikka's Lemma)**.** A Hintikka set is satisfiable.

Proof. Let M be a Hintikka set. I construct an algebra $\mathcal{A}$ out of M. For any sort S I define $S^{\mathcal{A}} = \{t \mid t \text{ is a ground term of sort } S \text{ in } M\}$. By Definition 3.6.6 the set $S^{\mathcal{A}}$ is not empty. For any function symbol $f : S_1 \times \ldots \times S_n \to S$ I define for all $s_i \in S_i^{\mathcal{A}}$ the function $f^{\mathcal{A}}(s_1, \ldots, s_n) = f(s_1, \ldots, s_n)$ if $f(s_1, \ldots, s_n) \in S^{\mathcal{A}}$ and $f^{\mathcal{A}}(s_1, \ldots, s_n) = s$ for some arbitrary ground term $s \in S^{\mathcal{A}}$ otherwise. So, functions are totally defined on their respective sorts. Predicates are interpreted by $P^{\mathcal{A}} = \{(t_1, \ldots, t_n) \mid P(t_1, \ldots, t_n) \in M\}$. Note that if $(t_1, \ldots, t_n) \in P^{\mathcal{A}}$ for some predicate $P \subseteq S_1 \times \ldots \times S_n$ then $t_i \in S_i^{\mathcal{A}}$ by construction. Then, by structural induction, for any formula $\phi \in M$ it holds $\mathcal{A} \models \phi$.

If ϕ is an atom $\mathcal{A} \models \phi$ holds by construction. If ϕ is a negative literal $\neg P(t_1, \ldots, t_n)$ then because M is a Hintikka set $P(t_1, \ldots, t_n) \notin M$ and hence $\mathcal{A} \models \phi$.

If ϕ is an α or β formula a simple application of the definition of a Hintikka set together with the induction hypothesis proves $\mathcal{A} \models \phi$.

If ϕ is a γ formula $\forall x_S.\phi'$ (or $\neg\exists x_S.\phi'$), then by definition of a Hintikka set $\gamma(t)$ is in M for all ground terms t of the correct sort occurring in M. Hence,

$\gamma(t) \in M$ for all $t \in S^{\mathcal{A}}$. By induction hypothesis $\mathcal{A} \models \gamma(t)$ for all $\gamma(t)$ and hence $\mathcal{A} \models \phi$.

If ϕ is a δ formula then by definition of a Hintikka set $\delta(c)$ is in M for some constant c. By induction hypothesis $\mathcal{A} \models \delta(c)$ and hence $\mathcal{A} \models \phi$. □

Lemma 3.6.8 (Open Branches of Saturated Tableaux constitute Hintikka Sets)**.** Let $s_0 \Rightarrow_{\mathrm{FT}} s_1 \Rightarrow_{\mathrm{FT}} \ldots$ be a saturated tableau derivation containing the derivation of an open branch $(M_1, J_1) \Rightarrow (M_2, J_2) \Rightarrow \ldots$ where M_i is always a strict subsequence of M_{i+1}. Then $M = \bigcup_i M_i$ is a Hintikka set.

Proof. Firstly, there is at least one ground term in M for each sort occurring in M due to the construction of the initial state and the definition of the γ-Expansion rule. By construction, all formulas in M are closed. It remains to show the different conditions for a Hintikka set. Condition 1 of a Hintikka set holds because the branch is open. Conditions 2, 3, 5 match exactly the condition for the branch to be saturated. For condition 5 consider the state s_j containing the pair (M_j, J_j) of the branch where some arbitrary ground term t appears for the first time in M_j. The γ-Expansion rules enables instantiation with t starting with M_j, because t occurs in M_j. Then by the fact that the branch is saturated, $\gamma(t) \in M_k$ for some $k > j$ and hence $\gamma(t) \in M$. □

Now its time to prove completeness for standard first-order tableau. The basic idea is, similar to superposition in the propositional and also first-order case, Section 3.12. If a tableau cannot be closed there exists a model for some branch.

Theorem 3.6.9 (Standard First-Order Tableau is Complete)**.** If ϕ is valid then the tableau calculus computes $\{((\neg\phi), J)\} \Rightarrow^*_{\mathrm{FT}} N$ and N is closed.

Proof. Proof by contradiction. Assume N is not closed. Therefore, it must contain an open branch. By Lemma 3.6.8 this branch constitutes a Hintikka set. By Lemma 3.6.7 the branch constitutes a model for $\neg\phi$, hence ϕ cannot be valid. □

One of the disadvantages of the standard tableau calculus is the guessing of ground terms in γ-Extensions. To get rid of this, the idea is to simply keep the universally quantified variable. Then branches are no longer closed by syntactically complementary formulas, but by complementary formulas modulo "appropriate instantiation" of the universally quantified variables. This requires a procedure that computes, in the simplest case, for two literals a substitution that makes them complementary, i.e., the respective atoms equal. Searching for a substitution making two terms, atoms (formulas) equal is called *unification*, see Section 3.7.

Lemma 3.6.10 (Compactness of First-Order Logic)**.** Let N be a, possibly countably infinite, set of first-order logic ground clauses. Then N is unsatisfiable iff there is a finite subset $N' \subseteq N$ such that N' is unsatisfiable.

Proof. If N is unsatisfiable, saturation via the tableau calculus generates a closed tableau. So there is an i such that $N \Rightarrow^i_{\mathrm{TAB}} N'$ and N' is closed. Every closed branch is the result of finitely many tableau rule applications on finitely many clauses $\{C_1, \ldots, C_n\} \subseteq N$. Let M be the union of all these finite clause sets, so $M \subseteq N$. Tableau is sound, so M is a finite, unsatisfiable subset of N. $\square$

3.7 Unification

Definition 3.7.1 (Unifier)**.** Two terms s and t of the same sort are said to be *unifiable* if there exists a well-sorted substitution σ so that $s\sigma = t\sigma$, the substitution σ is then called a well-sorted *unifier* of s and t. The unifier σ is called *most general unifier*, written $\sigma = mgu(s,t)$, if any other well-sorted unifier τ of s and t it can be represented as $\tau = \sigma\tau'$, for some well-sorted substitution τ'.

Obviously, two terms of different sort cannot be made equal by well-sorted instantiation. Since well-sortedness is preserved by all rules of the unification calculus, we assume from now an that all equations, terms, and substitutions are well-sorted.

The first calculus is the naive standard unification calculus that is typically found in the (old) literature on automated reasoning [14]. A state of the naive standard unification calculus is a set of equations E or $\bot$, where $\bot$ denotes that no unifier exists. The set E is also called a *unification problem.* The start state for checking whether two terms s, t, $\mathrm{sort}(s) = \mathrm{sort}(t)$, (or two non-equational atoms A, B) are unifiable is the set $E = \{s = t\}$ ($E = \{A = B\}$). A variable x is *solved* in E if $E = \{x = t\} \uplus E'$, $x \notin \mathrm{vars}(t)$ and $x \notin \mathrm{vars}(E)$.

A variable $x \in \mathrm{vars}(E)$ is called *solved* in E if $E = E' \uplus \{x = t\}$ and $x \notin \mathrm{vars}(t)$ and $x \notin \mathrm{vars}(E')$.

Tautology $\quad E \uplus \{t = t\} \;\Rightarrow_{\mathrm{SU}}\; E$

Decomposition $\quad E \uplus \{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\} \;\Rightarrow_{\mathrm{SU}}\; E \cup \{s_1 = t_1, \ldots, s_n = t_n\}$

Clash $\quad E \uplus \{f(s_1, \ldots, s_n) = g(s_1, \ldots, s_m)\} \;\Rightarrow_{\mathrm{SU}}\; \bot$
if $f \neq g$

Substitution $\quad E \uplus \{x = t\} \;\Rightarrow_{\mathrm{SU}}\; E\{x \mapsto t\} \cup \{x = t\}$
if $x \in vars(E)$ and $x \notin vars(t)$

Occurs Check $\quad E \uplus \{x = t\} \;\Rightarrow_{\mathrm{SU}}\; \bot$
if $x \neq t$ and $x \in vars(t)$

Orient $E \uplus \{t = x\} \ \Rightarrow_{\mathrm{SU}} \ E \cup \{x = t\}$
if $t \notin \mathcal{X}$

Theorem 3.7.2 (Soundness, Completeness and Termination of $\Rightarrow_{\mathrm{SU}}$)**.** If s, t are two terms with $\mathrm{sort}(s) = \mathrm{sort}(t)$ then

1. if $\{s = t\} \Rightarrow^*_{\mathrm{SU}} E$ then any equation $(s' = t') \in E$ is well-sorted, i.e., $\mathrm{sort}(s') = \mathrm{sort}(t')$.

2. $\Rightarrow_{\mathrm{SU}}$ terminates on $\{s = t\}$.

3. if $\{s = t\} \Rightarrow^*_{\mathrm{SU}} E$ then σ is a unifier (mgu) of E iff σ is a unifier (mgu) of $\{s = t\}$.

4. if $\{s = t\} \Rightarrow^*_{\mathrm{SU}} \bot$ then s and t are not unifiable.

5. if $\{s = t\} \Rightarrow^*_{\mathrm{SU}} \{x_1 = t_1, \ldots, x_n = t_n\}$ and this is a normal form, then $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is an mgu of s, t.

Proof. 1. by induction on the length of the derivation and a case analysis for the different rules.
2. for a state $E = \{s_1 = t_1, \ldots, s_n = t_n\}$ take the measure $\mu(E) := (n, M, k)$ where n is the number of unsolved variables, M the multiset of all term depths of the s_i, t_i and k the number of equations $t = x$ in E where t is not a variable. The state $\bot$ is mapped to $(0, \emptyset, 0)$. Then the lexicographic combination of $>$ on the naturals and its multiset extension shows that any rule application decrements the measure.
3. by induction on the length of the derivation and a case analysis for the different rules. Clearly, for any state where Clash, or Occurs Check generate $\bot$ the respective equation is not unifiable.
4. a direct consequence of 3.
5. if $E = \{x_1 = t_1, \ldots, x_n = t_n\}$ is a normal form, then for all $x_i = t_i$ we have $x_i \notin \mathrm{vars}(t_i)$ and $x_i \notin \mathrm{vars}(E \setminus \{x_i = t_i\})$, so $\{x_1 = t_1, \ldots, x_n = t_n\}\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\} = \{t_1 = t_1, \ldots, t_n = t_n\}$ and hence $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is an mgu of $\{x_1 = t_1, \ldots, x_n = t_n\}$. By 3. it is also an mgu of s, t. □

Example 3.7.3 (Size of Standard Unification Problems)**.** Any normal form of the unification problem E given by
$$\{f(x_1, g(x_1, x_1), x_3, \ldots, g(x_n, x_n)) = f(g(x_0, x_0), x_2, g(x_2, x_2), \ldots, x_{n+1})\}$$
with respect to $\Rightarrow_{\mathrm{SU}}$ is exponentially larger than E.

The second calculus, polynomial unification, prevents the problem of exponential growth by introducing an implicit representation for the mgu. For this calculus the size of a normal form is always polynomial in the size of the input unification problem.

Tautology $E \uplus \{t = t\} \ \Rightarrow_{\mathrm{PU}} \ E$

Decomposition $E \uplus \{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\} \;\Rightarrow_{\mathrm{PU}}\; E \uplus \{s_1 = t_1, \ldots, s_n = t_n\}$

Clash $E \uplus \{f(t_1, \ldots, t_n) = g(s_1, \ldots, s_m)\} \;\Rightarrow_{\mathrm{PU}}\; \bot$
if $f \neq g$

Occurs Check $E \uplus \{x = t\} \;\Rightarrow_{\mathrm{PU}}\; \bot$
if $x \neq t$ and $x \in vars(t)$

Orient $E \uplus \{t = x\} \;\Rightarrow_{\mathrm{PU}}\; E \uplus \{x = t\}$
if $t \notin \mathcal{X}$

Substitution $E \uplus \{x = y\} \;\Rightarrow_{\mathrm{PU}}\; E\{x \mapsto y\} \uplus \{x = y\}$
if $x \in vars(E)$ and $x \neq y$

Cycle $E \uplus \{x_1 = t_1, \ldots, x_n = t_n\} \;\Rightarrow_{\mathrm{PU}}\; \bot$
if there are positions p_i with $t_i|_{p_i} = x_{i+1}, t_n|_{p_n} = x_1$ and some $p_i \neq \epsilon$

Merge $E \uplus \{x = t, x = s\} \;\Rightarrow_{\mathrm{PU}}\; E \uplus \{x = t, t = s\}$
if $t, s \notin \mathcal{X}$ and $|t| \leq |s|$

Theorem 3.7.4 (Soundness, Completeness and Termination of $\Rightarrow_{\mathrm{PU}}$)**.** If s, t are two terms with $\mathrm{sort}(s) = \mathrm{sort}(t)$ then

1. if $\{s = t\} \Rightarrow^*_{\mathrm{PU}} E$ then any equation $(s' = t') \in E$ is well-sorted, i.e., $\mathrm{sort}(s') = \mathrm{sort}(t')$.

2. $\Rightarrow_{\mathrm{PU}}$ terminates on $\{s = t\}$.

3. if $\{s = t\} \Rightarrow^*_{\mathrm{PU}} E$ then σ is a unifier (mgu) of E iff σ is a unifier (mgu) of $\{s = t\}$.

4. if $\{s = t\} \Rightarrow^*_{\mathrm{PU}} \bot$ then s and t are not unifiable.

Theorem 3.7.5 (Normal Forms generated by $\Rightarrow_{\mathrm{PU}}$)**.** Let $\{s = t\} \Rightarrow^*_{\mathrm{PU}} \{x_1 = t_1, \ldots, x_n = t_n\}$ be a normal form. Then

1. $x_i \neq x_j$ for all $i \neq j$ and without loss of generality $x_i \notin \mathrm{vars}(t_{i+k})$ for all i, k, $1 \leq i < n$, $i + k \leq n$.

2. the substitution $\{x_1 \mapsto t_1\}\{x_2 \mapsto t_2\} \ldots \{x_n \mapsto t_n\}$ is an mgu of $s = t$.

Proof. 1. If $x_i = x_j$ for some $i \neq j$ then Merge is applicable. If $x_i \in \mathrm{vars}(t_i)$ for some i then Occurs Check is applicable. If the x_i cannot be ordered in the described way, then either Substitution or Cycle is applicable.
2. Since $x_i \notin \mathrm{vars}(t_{i+k})$ the composition yields the mgu. $\square$

γ	Descendant $\gamma(y)$
$\forall x_S.\psi$	$\psi\{x_S \mapsto y_S\}$
$\neg\exists x_S.\psi$	$\neg\psi\{x_S \mapsto y_S\}$
	for a fresh variable y_S

δ	Descendant $\delta(f(y_1,\ldots,y_n))$
$\exists x_S.\psi$	$\psi\{x_S \mapsto f(y_1,\ldots,y_n)\}$
$\neg\forall x_S.\psi$	$\neg\psi\{x_S \mapsto f(y_1,\ldots,y_n)\}$
	for some fresh Skolem function f,
	$f : \text{sort}(y_1) \times \ldots \times \text{sort}(y_n) \to S$

Figure 3.2: γ- and δ-Formulas

3.8 First-Order Free-Variable Tableau

An important disadvantage of standard first-order tableau is that the γ ground term instances need to be guessed. The main complexity in proving a formula to be valid lies in this guessing as for otherwise tableau terminates with a proof. Guessing useless ground terms may result in infinite branches. A natural idea is to guess ground terms that can eventually be used to close a branch. Of course, it is not known which ground term will close a branch. Therefore, it would be great to postpone the γ instantiations. This is the idea of free-variable first-order tableau. Instead of guessing a ground term for a γ formula, free-variable tableau introduces a fresh variable. Then a branch can be closed if two complementary literals have a common ground instance, i.e., their atoms are unifiable. The instantiation is delayed until a branch is closed for two literals via unification. As a consequence, for δ formulas no longer constants are introduced but shallow, so called *Skolem* terms in the formerly universally quantified variables that had the δ formula in their scope.

The new calculus needs to keep track of scopes of variables, so I move from a state as a set of pairs of a sequence and a set of constants, see standard first-order tableau Section 3.6, to a set of sequences of pairs (M_i, X_i) where X_i is a set of variables.

Definition 3.8.1 (Direct Free-Variable Tableau Descendant)**.** Given a γ- or δ-formula ϕ, Figure 3.2 shows its direct descendants.

The notion of closedness, Section 3.6, transfers exactly from standard to free-variable tableau. For α- and β-formulas the definition of an *open* formula remains unchanged as well. A γ- or δ-formula is called *open* in (M, X) if no direct descendant is contained in M. Note that instantiation of a tableau may remove direct descendants of γ- or δ-formulas by substituting terms for variables. Then a branch, pair (M, X), sequence M, is *open* if it is not closed and there is an open formula in M or there is pair of unifiable, complementary literals in M.

γ-Expansion $\quad N \uplus \{((\phi_1,\ldots,\psi,\ldots,\phi_n), X)\} \Rightarrow_{\text{FT}} N \cup \{((\phi_1,\ldots,\psi,\ldots,\phi_n,\psi'), X \cup$

$\{y\})\}$
provided ψ is a γ-formula, ψ' a $\gamma(y)$ descendant where y is fresh to the overall tableau and the sequence is not closed.

δ-Expansion $N \uplus \{((\phi_1, \ldots, \psi, \ldots, \phi_n), X)\} \Rightarrow_{\mathrm{FT}} N \cup \{(\phi_1, \ldots, \psi, \ldots, \phi_n, \psi'), X)\}$
provided ψ is an open δ-formula, $X = \{y_1, \ldots, y_n\}$, ψ' a $\delta(f(y_1, \ldots, y_n))$ descendant where f is fresh to the sequence, and the sequence is not closed.

Branch-Closing $N \uplus \{((\phi_1, \ldots, \phi_n), X)\} \Rightarrow_{\mathrm{FT}} (N \cup \{((\phi_1, \ldots, \phi_n), X)\})\sigma$
provided there are complementary literals ϕ_i and ϕ_j, $\mathrm{atom}(\phi_i)\sigma = \mathrm{atom}(\phi_j)\sigma$ for an mgu σ, and the sequence is not closed.

The first-order free-variable tableau calculus consists of the rules α-, and β-expansion, see Section 3.6, which are adapted to pairs of sequences and variable sets, and the above three rules γ-Expansion, δ-Expansion and Branch-Closing. It remains to define the instantiation of a tableau by a substitution. As usual the application of a substitution to a set means application to the elements. For a pair $((\phi_1, \ldots, \phi_n), X)$ it is defined by $((\phi_1, \ldots, \phi_n), X)\sigma := ((\phi_1\sigma, \ldots, \phi_n\sigma), X \setminus \mathrm{dom}(\sigma))$.

For free-varianle tableau, the γ rule has to be applied several times to the same formula as well in order to close a tableau, see the below example in Section 3.6. Constructing a closed tableau from initial state

$$\{((\forall x_S.(P(x_S) \to P(f(x_S))),\ P(b),\ \neg P(f(f(b)))), \emptyset)\}$$

is impossible without applying γ-Expansion twice to $\forall x_S.(P(x_S) \to P(f(x_S)))$ on some branch, where $b :\to S$, $f : S \to S$ and $P \subseteq S$. Below is the derivation of a closed tableau where I only show the added formulas and often abbreviate the parent sequence with an indexed M.

$$\begin{array}{ll}
\multicolumn{2}{l}{\{((\forall x_S.(P(x_S) \to P(f(x_S))),\ P(b),\ \neg P(f(f(b)))), \emptyset)} \\
\Rightarrow^{\gamma}_{\mathrm{FT}} & \{((M_1, P(y_S) \to P(f(y_S))), \{y_s\}) \\
\Rightarrow^{\beta}_{\mathrm{FT}} & \{((M_2, \neg P(y_S)), \{y_s\}), ((M_2, P(f(y_S))), \{y_s\})\} \\
\Rightarrow^{\mathrm{Closing}}_{\mathrm{FT}} & \{((M_2\sigma, \neg P(b)), \emptyset), ((M_2\sigma, P(f(b))), \emptyset)\} \\
\multicolumn{2}{l}{\text{the unifier is } \sigma = \{y_S \mapsto b\}, \text{of the literals } \neg P(y_S) \text{ and } P(b)} \\
\Rightarrow^{\gamma}_{\mathrm{FT}} & \{((M_2\sigma, P(f(b)), P(f(z_S)) \to P(f(f(z_S))))), \{z_S\})\} \\
\Rightarrow^{\beta}_{\mathrm{FT}} & \{((M_3, \neg P(f(z_S))), \{z_S\}), ((M_3, P(f(f(z_S)))), \{z_S\})\} \\
\Rightarrow^{\mathrm{Closing}}_{\mathrm{FT}} & \{((M_3\delta, \neg P(f(b))), \emptyset), ((M_3\delta, P(f(f(b)))), \emptyset)\} \\
\multicolumn{2}{l}{\text{the unifier is } \delta = \{z_S \mapsto f(b)\}, \text{of the literals } \neg P(z_S) \text{ and } P(f(b))} \\
\multicolumn{2}{l}{\text{now the tableau is closed}}
\end{array}$$

where $M_1 = (\forall x_S.(P(x_S) \to P(f(x_S))),\ P(b),\ \neg P(f(f(b))))$, $M_2 = M_1, (P(y_S) \to P(f(y_S)))$ and $M_3 = M_2\sigma, (P(f(b)), P(f(z_S)) \to P(f(f(z_S)))$.

A possibly infinite tableau derivation $s_0 \Rightarrow_{\mathrm{FT}} s_1 \Rightarrow_{\mathrm{FT}} \dots$ is called *saturated* if for all its open sequences M_i of some pair $(M_i, X_i) \in s_i$ where not all successor sequences of M_i are closed and all formulas ϕ occurring in M_i, there is an index $j > i$ and some pair $(M_j, X_j) \in s_j$, M_i is a prefix of M_j, if in case ϕ is an α-formula then both direct descendants are part of M_j, if it is a β-formula then one of its descendants is part of M_j, if it is a δ- or γ-formula then one direct descendant is part of M_j, and if Branch-Closing is applicable to M_i then M_j is closed.

Theorem 3.8.2 (Free-variable First-Order Tableau is Sound and Complete)**.** A formula ϕ is valid iff free-variable tableau computes a closed state out of $\{(\neg\phi, \emptyset)\}$.

Proof Idea: By lifting from standard first-order tableau. □

Here is another example including δ-Expansion applications. I assume the exsitence of exactly one sort with the respective defintions for the constants, functions, variables, and predicates.

$$\begin{array}{ll}
\multicolumn{2}{l}{\{((\neg[\exists w \forall x R(x,w,f(x,w)) \rightarrow \exists w \forall x \exists y R(x,w,y)]), \emptyset)} \\
\Rightarrow^{\alpha,*}_{\mathrm{FT}} & \{((M_1, \exists w \forall x\ R(x,w,f(x,w)), \neg\exists w \forall x \exists y\ R(x,w,y)), \emptyset)\} \\
\Rightarrow^{\delta}_{\mathrm{FT}} & \{((M_2, \forall x\ R(x,c,f(x,c))), \emptyset)\} \\
\Rightarrow^{\gamma}_{\mathrm{FT}} & \{((M_2, \forall x\ R(x,c,f(x,c)), \neg\forall x \exists y\ R(x,v_1,y)), \{v_1\})\} \\
\Rightarrow^{\delta}_{\mathrm{FT}} & \{((M_2, \forall x\ R(x,c,f(x,c)), \neg\forall x \exists y\ R(x,v_1,y), \forall x\ R(x,c,f(x,c)), \neg\exists y\ R(g(v_1),v_1,y)), \{v_1\})\} \\
\Rightarrow^{\gamma}_{\mathrm{FT}} & \{((M_3, R(v_2,c,f(v_2,c))), \{v_1,v_2\})\} \\
\Rightarrow^{\gamma}_{\mathrm{FT}} & \{((M_3, R(v_2,c,f(v_2,c)), \neg R(g(v_1),v_1,v_3)), \{v_1,v_2,v_3\})\} \\
\Rightarrow^{\mathrm{Closing}}_{\mathrm{FT}} & \{((M_3\sigma, R(g(c),c,f(g(c),c)), \neg R(g(c),c,f(g(c),c))), \emptyset)\}
\end{array}$$

the unifier is $\sigma = \{v_1 \mapsto c,\ v_2 \mapsto g(c),\ v_3 \mapsto f(g(c),c)\}$

now the tableau is closed

where $M_1 = \neg[\exists w \forall x R(x,w,f(x,w)) \rightarrow \exists w \forall x \exists y R(x,w,y)]$,
$M_2 = M_1, \exists w \forall x\ R(x,w,f(x,w)), \neg\exists w \forall x \exists y\ R(x,w,y)$, and
$M_3 = M_2, \forall x\ R(x,c,f(x,c)), \neg\forall x \exists y\ R(x,v_1,y), \forall x\ R(x,c,f(x,c)), \neg\exists y\ R(g(v_1),v_1,y)$.

Semantic Tableau vs. Resolution

1. Tableau: global, goal-oriented, "backward".
2. Resolution: local, "forward".
3. Goal-orientation is a clear advantage if only a small subset of a large set of formulas is necessary for a proof. (Note that resolution provers saturate also those parts of the clause set that are irrelevant for proving the goal.)