3.17 Semantic Tableaux for First-Order Logic

There are two ways to extend the tableau calculus to quantified formulas:

- using ground instantiation
- using free variables

Tableaux with Ground Instantiation

Classification of quantified formulas:

universal		existential	
γ	$\gamma(t)$	δ	$\delta(t)$
$\forall x F$	$F\{x \mapsto t\}$	$\exists x F$	$F\{x \mapsto t\}$
$\neg\exists x F$	$\neg F\{x \mapsto t\}$	$\neg\forall x F$	$\neg F\{x \mapsto t\}$

Idea:

Replace universally quantified formulas by appropriate ground instances.

γ-expansion

$$\frac{\gamma}{\gamma(t)}$$ where t is some ground term

δ-expansion

$$\frac{\delta}{\delta(c)}$$ where c is a new Skolem constant

Skolemization becomes part of the calculus and needs not necessarily be applied in a preprocessing step. Of course, one could do Skolemization beforehand, and then the δ-rule would not be needed.

Note:

Skolem *constants* are sufficient:
In a δ-formula $\exists x\, F$, $\exists$ is the outermost quantifier and x is the only free variable in F.

Problems:

Having to guess ground terms is unpractical.

Even worse, we may have to guess *several* ground instances, as strictness for γ is incomplete. For instance, constructing a closed tableau for

$$\{\forall x\,(P(x) \to P(f(x))),\ P(b),\ \neg P(f(f(b)))\}$$

is impossible without applying γ-expansion twice on one path.

Free-Variable Tableaux

An alternative approach:

Delay the instantiation of universally quantified variables.

Replace universally quantified variables by new free variables.

Intuitively, the free variables are universally quantified outside of the entire tableau.

γ-expansion

$$\frac{\gamma}{\gamma(x)} \quad \text{where } x \text{ is a new free variable}$$

δ-expansion

$$\frac{\delta}{\delta(f(x_1, \ldots, x_n))}$$

where f is a new Skolem function, and the x_i are the free variables in δ

Application of expansion rules has to be supplemented by a *substitution rule:*

(iii) If T is a tableau for $\{F_1, \ldots, F_n\}$ and if σ is a substitution, then $T\sigma$ is also a tableau for $\{F_1, \ldots, F_n\}$.

The substitution rule may, potentially, modify all the formulas of a tableau. This feature is what is makes the tableau method a *global proof method.* (Resolution, by comparison, is a local method.)

One can show that it is sufficient to consider substitutions σ for which there is a path in T containing two *literals* $\neg A$ and B such that $\sigma = \text{mgu}(A, B)$. Such tableaux are called *AMGU-Tableaux*.

Example

1.	$\neg[\exists w \forall x\ p(x, w, f(x, w)) \rightarrow \exists w \forall x \exists y\ p(x, w, y)]$	
2.	$\exists w \forall x\ p(x, w, f(x, w))$	1_1 $[\alpha]$
3.	$\neg \exists w \forall x \exists y\ p(x, w, y)$	1_2 $[\alpha]$
4.	$\forall x\ p(x, c, f(x, c))$	$2(c)$ $[\delta]$
5.	$\neg \forall x \exists y\ p(x, v_1, y)$	$3(v_1)$ $[\gamma]$
6.	$\neg \exists y\ p(b(v_1), v_1, y)$	$5(b(v_1))$ $[\delta]$
7.	$p(v_2, c, f(v_2, c))$	$4(v_2)$ $[\gamma]$
8.	$\neg p(b(v_1), v_1, v_3)$	$6(v_3)$ $[\gamma]$

7. and 8. are complementary (modulo unification):

$$v_2 \doteq b(v_1),\ c \doteq v_1,\ f(v_2, c) \doteq v_3$$

is solvable with an mgu $\sigma = \{v_1 \mapsto c,\ v_2 \mapsto b(c),\ v_3 \mapsto f(b(c), c)\}$, and hence, $T\sigma$ is a closed (linear) tableau for the formula in 1.

Problem:

Strictness for γ is still incomplete. For instance, constructing a closed tableau for

$$\{\forall x\,(P(x) \rightarrow P(f(x))),\ P(b),\ \neg P(f(f(b)))\}$$

is impossible without applying γ-expansion twice on one path.

Semantic Tableaux vs. Resolution

- Tableaux: global, goal-oriented, "backward".
- Resolution: local, "forward".
- 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.)
- Resolution can be combined with more powerful redundancy elimination methods; because of its global nature this is more difficult for the tableau method.
- Resolution can be refined to work well with equality; for tableaux this seems to be impossible.
- On the other hand tableau calculi can be easily extended to other logics; in particular tableau provers are very successful in modal and description logics.

3.18 Other Inference Systems

- Instantiation-based methods
  Resolution-based instance generation
  Disconnection calculus
  ...
- Natural deduction
- Sequent calculus/Gentzen calculus
- Hilbert calculus

Instantiation-Based Methods for FOL

Idea:

Overlaps of complementary literals produce instantiations (as in resolution);

However, contrary to resolution, clauses are not recombined.

Instead: treat remaining variables as constant and use efficient propositional proof methods, such as DPLL.

There are both saturation-based variants, such as partial instantiation [Hooker et al.] or resolution-based instance generation (Inst-Gen) [Ganzinger and Korovin], and tableau-style variants, such as the disconnection calculus [Billon; Letz and Stenz].

Natural Deduction

Natural deduction (Prawitz):

Models the concept of proofs from assumptions as humans do it (cf. Fitting or Huth/Ryan).

Sequent Calculus

Sequent calculus (Gentzen):

Assumptions internalized into the data structure of sequents

$$F_1, \ldots, F_m \vdash G_1, \ldots, G_k$$

meaning

$$F_1 \wedge \cdots \wedge F_m \rightarrow G_1 \vee \cdots \vee G_k$$

A kind of mixture between natural deduction and semantic tableaux.

Inferences rules, e.g.:

$$\frac{\Gamma \vdash \Delta}{\Gamma, F \vdash \Delta}\ (WL) \qquad \frac{\Gamma, F \vdash \Delta \qquad \Sigma, G \vdash \Pi}{\Gamma, \Sigma, F \vee G \vdash \Delta, \Pi}\ (\vee L)$$

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash F, \Delta}\ (WR) \qquad \frac{\Gamma \vdash F, \Delta \qquad \Sigma \vdash G, \Pi}{\Gamma, \Sigma \vdash F \wedge G, \Delta, \Pi}\ (\wedge R)$$

Perfect symmetry between the handling of assumptions and their consequences.

Can be used both backwards and forwards.

Hilbert Calculus

Hilbert calculus:

Direct proof method (proves a theorem from axioms, rather than refuting its negation)

Axiom schemes, e. g.,

$$F \to (G \to F)$$
$$(F \to (G \to H)) \to ((F \to G) \to (F \to H))$$

plus Modus ponens:

$$\frac{F \qquad F \to G}{G}$$

Unsuitable for finding or reading proofs, but sometimes used for *specifying* (e.g. modal) logics.

4 First-Order Logic with Equality

Equality is the most important relation in mathematics and functional programming.

In principle, problems in first-order logic with equality can be handled by any prover for first-order logic without equality:

4.1 Handling Equality Naively

Proposition 4.1 *Let F be a closed first-order formula with equality. Let $\sim \notin \Pi$ be a new predicate symbol. The set $Eq(\Sigma)$ contains the formulas*

$$\begin{gathered}
\forall x\,(x \sim x) \\
\forall x, y\,(x \sim y \to y \sim x) \\
\forall x, y, z\,(x \sim y \wedge y \sim z \to x \sim z) \\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_n \sim y_n \to f(x_1, \ldots, x_n) \sim f(y_1, \ldots, y_n)) \\
\forall \vec{x}, \vec{y}\,(x_1 \sim y_1 \wedge \cdots \wedge x_m \sim y_m \wedge p(x_1, \ldots, x_m) \to p(y_1, \ldots, y_m))
\end{gathered}$$

for every $f \in \Omega$ and $p \in \Pi$. Let $\tilde{F}$ be the formula that one obtains from F if every occurrence of $\approx$ is replaced by $\sim$. Then F is satisfiable if and only if $Eq(\Sigma) \cup \{\tilde{F}\}$ is satisfiable.

Proof. Let $\Sigma = (\Omega, \Pi)$, let $\Sigma_1 = (\Omega, \Pi \cup \{\sim\})$.

For the "only if" part assume that F is satisfiable and let $\mathcal{A}$ be a Σ-model of F. Then we define a Σ_1-algebra $\mathcal{B}$ in such a way that $\mathcal{B}$ and $\mathcal{A}$ have the same universe, $f_\mathcal{B} = f_\mathcal{A}$ for every $f \in \Omega$, $p_\mathcal{B} = p_\mathcal{A}$ for every $p \in \Pi$, and $\sim_\mathcal{B}$ is the identity relation on the universe. It is easy to check that $\mathcal{B}$ is a model of both $\tilde{F}$ and of $Eq(\Sigma)$.

The proof of the "if" part consists of two steps.

Assume that the Σ_1-algebra $\mathcal{B} = (U_\mathcal{B},\ (f_\mathcal{B} : U^n \to U)_{f\in\Omega},\ (p_\mathcal{B} \subseteq U_\mathcal{B}^m)_{p\in\Pi\cup\{\sim\}})$ is a model of $Eq(\Sigma) \cup \{\tilde{F}\}$. In the first step, we can show that the interpretation $\sim_\mathcal{B}$ of $\sim$ in $\mathcal{B}$ is a congruence relation. We will prove this for the symmetry property, the other properties of congruence relations, that is, reflexivity, transitivity, and congruence with respect to functions and predicates are shown analogously. Let $a, a' \in U_\mathcal{B}$ such that $a \sim_\mathcal{B} a'$. We have to show that $a' \sim_\mathcal{B} a$. Since $\mathcal{B}$ is a model of $Eq(\Sigma)$, $\mathcal{B}(\beta)(\forall x, y\,(x \sim y \to y \sim x)) = 1$ for every β, hence $\mathcal{B}(\beta[x \mapsto b_1, y \mapsto b_2])(x \sim y \to y \sim x) = 1$ for every β and every $b_1, b_2 \in U_\mathcal{B}$. Set $b_1 = a$ and $b_2 = a'$, then $1 = \mathcal{B}(\beta[x \mapsto a, y \mapsto a'])(x \sim y \to y \sim x) = \max(1 - \mathcal{B}(\beta[x \mapsto a, y \mapsto a'])(x \sim y), \mathcal{B}(\beta[x \mapsto a, y \mapsto a'])(y \sim x))$, and since $a \sim_\mathcal{B} a'$ holds by assumption, $a' \sim_\mathcal{B} a$ must also hold.

In the second step, we will now construct a Σ-algebra $\mathcal{A}$ from $\mathcal{B}$ and the congruence relation $\sim_\mathcal{B}$. Let $[a]$ be the congruence class of an element $a \in U_\mathcal{B}$ with respect to $\sim_\mathcal{B}$. The universe $U_\mathcal{A}$ of $\mathcal{A}$ is the set $\{\, [a] \mid a \in U_\mathcal{B} \,\}$ of congruence classes of the universe of $\mathcal{B}$. For a

function symbol $f \in \Omega$, we define $f_{\mathcal{A}}([a_1], \ldots, [a_n]) = [f_{\mathcal{B}}(a_1, \ldots, a_n)]$, and for a predicate symbol $p \in \Pi$, we define $([a_1], \ldots, [a_n]) \in p_{\mathcal{A}}$ if and only if $(a_1, \ldots, a_n) \in p_{\mathcal{B}}$. Observe that this is well-defined: If we take different representatives of the same congruence class, we get the same result by congruence of $\sim_{\mathcal{B}}$. Now for every Σ-term t and every $\mathcal{B}$-assignment β, $[\mathcal{B}(\beta)(t)] = \mathcal{A}(\gamma)(t)$, where γ is the $\mathcal{A}$-assignment that maps every variable x to $[\beta(x)]$, and analogously for every Σ-formula G, $\mathcal{B}(\beta)(\tilde{G}) = \mathcal{A}(\gamma)(G)$. Both properties can easily shown by structural induction. Consequently, $\mathcal{A}$ is a model of F. □

By giving the equality axioms explicitly, first-order problems with equality can in principle be solved by a standard resolution or tableaux prover.

But this is unfortunately not efficient (mainly due to the transitivity and congruence axioms).

Equality is theoretically difficult: First-order functional programming is Turing-complete.

But: resolution theorem provers cannot even solve equational problems that are intuitively easy.

Consequence: to handle equality efficiently, knowledge must be integrated into the theorem prover.

Roadmap

How to proceed:

- This semester: Equations (unit clauses with equality)

  Term rewrite systems
  Expressing semantic consequence syntactically
  Knuth-Bendix-Completion
  Entailment for equations

- Next semester: Equational clauses

  Combining resolution and KB-completion → Superposition
  Entailment for clauses with equality

4.2 Rewrite Systems

Let E be a set of (implicitly universally quantified) equations.

The *rewrite relation* $\to_E \subseteq \mathrm{T}_\Sigma(X) \times \mathrm{T}_\Sigma(X)$ is defined by

$$\begin{aligned} s \to_E t \quad \text{iff} \quad & \text{there exist } (l \approx r) \in E,\ p \in \mathrm{pos}(s), \\ & \text{and } \sigma : X \to \mathrm{T}_\Sigma(X), \\ & \text{such that } s/p = l\sigma \text{ and } t = s[r\sigma]_p. \end{aligned}$$

An instance of the lhs (left-hand side) of an equation is called a *redex* (reducible expression). *Contracting* a redex means replacing it with the corresponding instance of the rhs (right-hand side) of the rule.

An equation $l \approx r$ is also called a *rewrite rule*, if l is not a variable and $\mathrm{var}(l) \supseteq \mathrm{var}(r)$.

Notation: $l \to r$.

A set of rewrite rules is called a *term rewrite system (TRS)*.

We say that a set of equations E or a TRS R is terminating, if the rewrite relation $\to_E$ or $\to_R$ has this property.

(Analogously for other properties of abstract reduction systems).

Note: If E is terminating, then it is a TRS.

E-Algebras

Let E be a set of universally quantified equations. A model of E is also called an *E-algebra*.

If $E \models \forall\vec{x}(s \approx t)$, i. e., $\forall\vec{x}(s \approx t)$ is valid in all E-algebras, we write this also as $s \approx_E t$.

Goal:
Use the rewrite relation $\to_E$ to express the semantic consequence relation syntactically:

$s \approx_E t$ if and only if $s \leftrightarrow_E^* t$.

Let E be a set of equations over $\mathrm{T}_\Sigma(X)$. The following inference system allows to derive consequences of E:

$$E \vdash t \approx t \qquad \textit{(Reflexivity)}$$

$$\frac{E \vdash t \approx t'}{E \vdash t' \approx t} \qquad \textit{(Symmetry)}$$

$$\frac{E \vdash t \approx t' \qquad E \vdash t' \approx t''}{E \vdash t \approx t''} \qquad \textit{(Transitivity)}$$

$$\frac{E \vdash t_1 \approx t_1' \quad \ldots \quad E \vdash t_n \approx t_n'}{E \vdash f(t_1, \ldots, t_n) \approx f(t_1', \ldots, t_n')} \qquad \textit{(Congruence)}$$

$$E \vdash t\sigma \approx t'\sigma \quad \text{if } (t \approx t') \in E \text{ and } \sigma : X \to \mathrm{T}_\Sigma(X) \qquad \textit{(Instance)}$$

Lemma 4.2 *The following properties are equivalent:*

(i) $s \leftrightarrow_E^* t$

(ii) $E \vdash s \approx t$ *is derivable.*

Proof. (i)⇒(ii): $s \leftrightarrow_E t$ implies $E \vdash s \approx t$ by induction on the depth of the position where the rewrite rule is applied; then $s \leftrightarrow_E^* t$ implies $E \vdash s \approx t$ by induction on the number of rewrite steps in $s \leftrightarrow_E^* t$.

(ii)⇒(i): By induction on the size (number of symbols) of the derivation for $E \vdash s \approx t$. □

Constructing a *quotient algebra:*

Let X be a set of variables.

For $t \in \mathrm{T}_\Sigma(X)$ let $[t] = \{\, t' \in \mathrm{T}_\Sigma(X) \mid E \vdash t \approx t' \,\}$ be the *congruence class* of t.

Define a Σ-algebra $\mathrm{T}_\Sigma(X)/E$ (abbreviated by $\mathcal{T}$) as follows:

$U_\mathcal{T} = \{\, [t] \mid t \in \mathrm{T}_\Sigma(X) \,\}$.

$f_\mathcal{T}([t_1], \ldots, [t_n]) = [f(t_1, \ldots, t_n)]$ for $f \in \Omega$.

Lemma 4.3 *$f_\mathcal{T}$ is well-defined: If $[t_i] = [t_i']$, then $[f(t_1, \ldots, t_n)] = [f(t_1', \ldots, t_n')]$.*

Proof. Follows directly from the *Congruence* rule for $\vdash$. □

Lemma 4.4 *$\mathcal{T} = \mathrm{T}_\Sigma(X)/E$ is an E-algebra.*

Proof. Let $\forall x_1 \ldots x_n (s \approx t)$ be an equation in E; let β be an arbitrary assignment.

We have to show that $\mathcal{T}(\beta)(\forall \vec{x}(s \approx t)) = 1$, or equivalently, that $\mathcal{T}(\gamma)(s) = \mathcal{T}(\gamma)(t)$ for all $\gamma = \beta[\, x_i \mapsto [t_i] \mid 1 \le i \le n \,]$ with $[t_i] \in U_\mathcal{T}$.

Let $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, then $s\sigma \in \mathcal{T}(\gamma)(s)$ and $t\sigma \in \mathcal{T}(\gamma)(t)$.

By the *Instance* rule, $E \vdash s\sigma \approx t\sigma$ is derivable, hence $\mathcal{T}(\gamma)(s) = [s\sigma] = [t\sigma] = \mathcal{T}(\gamma)(t)$. □

Lemma 4.5 *Let X be a countably infinite set of variables; let $s, t \in \mathrm{T}_\Sigma(X)$. If $\mathrm{T}_\Sigma(X)/E \models \forall \vec{x}(s \approx t)$, then $E \vdash s \approx t$ is derivable.*

Proof. Assume that $\mathcal{T} \models \forall \vec{x}(s \approx t)$, i. e., $\mathcal{T}(\beta)(\forall \vec{x}(s \approx t)) = 1$. Consequently, $\mathcal{T}(\gamma)(s) = \mathcal{T}(\gamma)(t)$ for all $\gamma = \beta[\, x_i \mapsto [t_i] \mid 1 \le i \le n \,]$ with $[t_i] \in U_\mathcal{T}$.

Choose $t_i = x_i$, then $[s] = \mathcal{T}(\gamma)(s) = \mathcal{T}(\gamma)(t) = [t]$, so $E \vdash s \approx t$ is derivable by definition of $\mathcal{T}$. □

Theorem 4.6 ("Birkhoff's Theorem") *Let X be a countably infinite set of variables, let E be a set of (universally quantified) equations. Then the following properties are equivalent for all $s, t \in \mathrm{T}_\Sigma(X)$:*

(i) $s \leftrightarrow_E^* t$.

(ii) $E \vdash s \approx t$ *is derivable.*

(iii) $s \approx_E t$, *i. e.,* $E \models \forall \vec{x}(s \approx t)$.

(iv) $\mathrm{T}_\Sigma(X)/E \models \forall \vec{x}(s \approx t)$.

Proof. (i)⇔(ii): Lemma 4.2.

(ii)⇒(iii): By induction on the size of the derivation for $E \vdash s \approx t$.

(iii)⇒(iv): Obvious, since $\mathcal{T} = \mathrm{T}_\Sigma(X)/E$ is an E-algebra.

(iv)⇒(ii): Lemma 4.5. □

Universal Algebra

$\mathrm{T}_\Sigma(X)/E = \mathrm{T}_\Sigma(X)/{\approx_E} = \mathrm{T}_\Sigma(X)/{\leftrightarrow_E^*}$ is called the *free E-algebra* with generating set $X/{\approx_E} = \{\, [x] \mid x \in X \,\}$:

Every mapping $\varphi : X/{\approx_E} \to \mathcal{B}$ for some E-algebra $\mathcal{B}$ can be extended to a homomorphism $\hat{\varphi} : \mathrm{T}_\Sigma(X)/E \to \mathcal{B}$.

$\mathrm{T}_\Sigma(\emptyset)/E = \mathrm{T}_\Sigma(\emptyset)/{\approx_E} = \mathrm{T}_\Sigma(\emptyset)/{\leftrightarrow_E^*}$ is called the *initial E-algebra*.

$\approx_E = \{\, (s,t) \mid E \models s \approx t \,\}$ is called the *equational theory* of E.

$\approx_E^I = \{\, (s,t) \mid \mathrm{T}_\Sigma(\emptyset)/E \models s \approx t \,\}$ is called the *inductive theory* of E.

Example:

Let $E = \{\forall x(x + 0 \approx x),\ \forall x \forall y(x + s(y) \approx s(x + y))\}$. Then $x + y \approx_E^I y + x$, but $x + y \not\approx_E y + x$.