Logic LVA 703600 VU3

http://cl-informatik.uibk.ac.at/teaching/ws05/logic/

Georg Moser (VU)[1] Christian Vogt (VU)[2]

[1]georg.moser@uibk.ac.at
office hours: Thursday 1pm–3pm

[2]christian.vogt@uibk.ac.at
office hours: Tuesday 9am–11am

Autumn 2005

Motivation

let us look at a tableau proof of

$$\{(\forall x)(P(x) \vee Q(x)), (\forall x)\neg P(x), (\exists x)\neg Q(x)\}$$

closed	free-variable	
$(\forall x)(P(x) \vee Q(x))$	$(\forall x)(P(x) \vee Q(x))$	$\frac{\gamma}{\gamma(x)}$
$(\forall x)\neg P(x)$	$(\forall x)\neg P(x)$	(for an unbound variable x)
$(\exists x)\neg Q(x)$	$(\exists x)\neg Q(x)$	
$\neg Q(c)$	$\neg Q(c)$	
$\neg P(d)$	$\neg P(x)$	$\frac{\delta}{\delta(f(x_1,\dots,x_n))}$
$P(c) \vee Q(c)$	$P(y) \vee Q(y)$	(for f new, $\vec{x}$ all free variables in δ)
$P(c)$ / $Q(c)$	$P(y)$ / $Q(y)$	
$\neg P(c)$ / $\neg P(c)$		

Unification

to close the tableau, we have to find σ such that

$$Q(c)\sigma = Q(y)\sigma \qquad P(x)\sigma = P(y)\sigma$$

obviously $\sigma = \{x \mapsto c, y \mapsto c\}$ would be sufficient

- ➡ a unification problem is a finite set of equations

$$S = \{s_1 =^? t_1, \dots, s_n =^? t_n\}$$

- ➡ a unifier of S is a substitution such that

$$s_i\sigma = t_i\sigma \quad \text{for all } i = 1, \dots, n$$

- ➡ a substitution σ is more general than a substitution τ, if $\tau = \sigma\rho$ for some substitution ρ; we write $\sigma \lesssim \tau$
- ➡ a most general unifier (mgu) is a unifier σ s.t. for all unifiers τ: $\sigma \lesssim \tau$

Example

- ➡ the unification problem $\{f(y, h(a)) = f(h(x), h(z))\}$ is solvable with

$$\sigma_1 = \{y \mapsto h(x), z \mapsto a\}$$
$$\sigma_2 = \{x \mapsto k(w), y \mapsto h(k(w)), z \mapsto a\}$$

  but $\sigma_1 \lesssim \sigma_2$ and σ_1 is a mgu
- ➡ the unification problem $\{f(x, x) = f(a, b)\}$ is not solvable

Lemma idempotent substitutions

a substitution σ is idempotent if $\sigma = \sigma\sigma$; then

- ➡ a substitution σ is idempotent iff $\mathrm{dom}(\sigma) \cap \mathrm{vrg}(\sigma) = \emptyset$

Theorem If a unification problem S is solvable, then it has an idempotent mgu.

Solved Forms

- a unification problem $S = \{x_1 = t_1, \dots, x_n = t_n\}$ is in solved form if the x_i are pairwise distinct and none of the x_i occurs in any of the t_j
- for S in solved form, we define $\vec{S} = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$

Lemma let S be in solved form; then

- for any unifier σ of S: $\vec{S}\sigma = \sigma$
- $\vec{S}$ is an idempotent mgu of S

Proof (of the 2nd property)

- idempotency follows as $\mathrm{dom}(\vec{S}) \cap \mathrm{vrg}(\vec{S}) = \emptyset$
- $\vec{S}$ is a unifier: $x_i\vec{S} = t_i = t_i\vec{S}$
- $\vec{S}$ is even a mgu: for all unifiers σ: $\vec{S} \lesssim \sigma$ by the 1. property □

Unification Algorithm

Delete	$\dfrac{\{t = t\} \uplus S}{S}$	
Decompose	$\dfrac{\{f(t_1,\dots,t_n) = f(u_1,\dots,u_n)\} \uplus S}{\{t_1 = u_1,\dots,t_n = u_n\} \cup S}$	
Orient	$\dfrac{\{t = x\} \uplus S}{\{x = t\} \cup S}$	if $t \notin \mathbf{V}$
Eliminate	$\dfrac{\{x = t\} \uplus S}{\{x = t\} \cup S\{x \mapsto t\}}$	if $x \in \mathrm{var}(S) - \mathrm{var}(t)$

let $S \Rightarrow T$ denote that T is reachable from S

we define

$\mathrm{Unify}(S)$ = while there is some T such that $S \Rightarrow T$ do $S := T$; if S is in solved form then return $\vec{S}$ else fail

The Unification Theorem

Example we consider the problem

$$S = \{x = f(a), g(x,x) = g(x,y)\}$$

$$\begin{aligned}\{x = f(a), g(x,x) = g(x,y)\} &\Rightarrow \{x = f(a), g(f(a),f(a)) = g(f(a),y)\}\\ &\Rightarrow \{x = f(a), f(a) = f(a), f(a) = y\}\\ &\Rightarrow \{x = f(a), f(a) = y\}\\ &\Rightarrow \{x = f(a), y = f(a)\}\end{aligned}$$

□

Theorem Unification Theorem

- Unify terminates on all inputs
- if Unify returns σ, then σ is an idempotent mgu of S
- if S is solvable, Unify does not fail

Proof (of termination)

- a variable x is called solved if it occurs exactly once in S and $x = t \in S$ with $x \notin \mathrm{var}(t)$
- we write $|t|$ to denote the number of symbols in t
- define a measure (n_1, n_2, n_3) for S
  - n_1 is the number of variables in S that are unsolved
  - n_2 is the size of S (i.e. $\sum_{s=t \in S}(|s| + |t|)$)
  - n_3 the number of equations $t = x \in S$
- the measure decreases lexicographically □

Proof (of completeness)

it is easy to see that the transformation rules are unifier-preserving; moreover we use two fundamental properties of terms

- an equation $f(s_1,\dots,s_n) = g(t_1,\dots,t_m)$ for $f \neq g$ has no solution
- an equation $x = t$, $x \in \mathrm{var}(t)$ and $x \neq t$ has no solution □

Refinements of Unify

we introduce a special unification problem $\perp$ that has no solution and add the following rules:

Clash	$\dfrac{\{f(s_1,\dots,s_n) = g(t_1,\dots,t_m)\} \uplus S}{\perp}$	
Occur-Check	$\dfrac{\{x = t\} \uplus S}{\perp}$	if $x \in \mathrm{var}(t)$ and $x \neq t$

Example

consider the problem $\{f(x,x) = f(y,g(y))\}$

$$\begin{aligned}\{f(x,x) = f(y,g(y))\} &\Rightarrow \{x = y, x = g(y)\}\\ &\Rightarrow \{x = y, y = g(y)\}\\ &\Rightarrow \{\perp\} \qquad \text{Occur-Check} \qquad \square\end{aligned}$$

Definition L^{sko}

- let $L = L(\mathbf{R},\mathbf{F},\mathbf{C})$ be a language; let par denote a countable set of constants not in $\mathbf{C}$; let sko be a countable set of function symbols not in $\mathbf{F}$;
- the function symbols in sko are called Skolem functions
- we write L^{sko} to denote $L(\mathbf{R}, \mathbf{F} \cup \text{sko}, \mathbf{C} \cup \text{par})$

Remark free-variable tableau proofs will be of sentences of L and use formulas of L^{sko}

Definition tableau substitutions

- let σ be a substitution and $\mathbf{T}$ a tableau; we define $\mathbf{T}\sigma$ as the result of replacing every $X \in \mathbf{T}$ by $X\sigma$
- σ is free for a tableau $\mathbf{T}$ if σ is free for every formula in $\mathbf{T}$

Free-Variable Semantic Tableaux

- the language of free-variable tableau is L^{sko}
- the quantifier rules are

$$\frac{\gamma}{\gamma(x)} \qquad\qquad \frac{\delta}{\delta(f(x_1,\dots,x_n))}$$

(for an unbound variable x) (for f new Skolem, $\vec{x}$ all free variables in δ)

- σ is free for a tableau $\mathbf{T}$ if σ is free for every formula in $\mathbf{T}$
- tableau substitution rule: If $\mathbf{T}$ is a tableau for S and σ is free for $\mathbf{T}$ then $\mathbf{T}\sigma$ is also a tableau for S.

Example we consider a tableau-proof of

$$(\exists w)(\forall x)R(x,w,f(x,w)) \to (\exists w)(\forall x)(\exists y)R(x,w,y)$$

$\neg\{(\exists w)(\forall x)R(x,w,f(x,w)) \to (\exists w)(\forall x)(\exists y)R(x,w,y)\}$	
$(\exists w)(\forall x)R(x,w,f(x,w))$	
$\neg(\exists w)(\forall x)(\exists y)R(x,w,y)$	
$(\forall x)R(x,a,f(x,a))$	δ-rule with a Skolem
$\neg(\forall x)(\exists y)R(x,v_1,y)$	γ-rule with v_1 new
$\neg(\exists y)R(b(v_1),v_1,y)$	δ-rule with b Skolem
$R(v_2,a,f(v_2,a))$	γ-rule with v_2 new
$\neg R(b(v_1),v_1,v_3)$	γ-rule with v_3 new

as final step we apply the free substitution

$$\sigma = \{v_1 \mapsto a, v_2 \mapsto b(a), v_3 \mapsto f(b(a),a)\}$$

to make $R(v_2,a,f(v_2,a))$ and $\neg R(b(v_1),v_1,v_3)$ conflict $\square$

how-to find the substitution σ? ➡: use unification

but σ has to be free! ➡: consider atomic closure, only

why does this work:

- ➡ let A and $\neg B$ be quantifier-free and occur on a branch in $\mathbf{T}$
- ➡ suppose σ is a "unifier" for A and B
- ➡ clearly $\mathrm{vrg}(\sigma) \subset \mathsf{fvar}(A) \cup \mathsf{fvar}(B)$
- ➡ let $\{v_1, \ldots, v_k\}$ denote the variables introduced by a γ-rule; by definition the v_i are distinct from any bound variable
- ➡ note that $\mathsf{fvar}(A) \cup \mathsf{fvar}(B) \subseteq \{v_1, \ldots, v_k\}$
- ➡ hence σ is free for $\mathbf{T}$

Definition atomic closure rule

suppose $\mathbf{T}$ is a tableau for S; some branch of $\mathbf{T}$ contains A and $\neg B$, both atomic; then $\mathbf{T}\sigma$ is a tableau for S, where σ is a mgu of A and B

Soundness

Theorem Soundness Theorem

If the sentence X has a free-variable tableau proof, then X is valid.

Proof (sketch)

the new problem are the free-variables introduced by γ-rules, to handle these, we treat them as universally quantified □

we informally define tableau strategies: a tableau strategy $\mathcal{R}$ for a tableau $\mathbf{T}$ expresses that either

- ➡ no continuation of a tableau is possible (using side-information), or
- ➡ produces an expansion $\mathbf{T}'$ (and perhaps some side-information)

Example we can define a strategy $\mathbf{R}$ to express that

- ➡ only unused non-literals are expanded
- ➡ a priority order on the branches is enforced
- ➡ a priority order on formula occurrences is enforced □

Definition fairness

we call a strategy $\mathcal{R}$ fair if for any sentence X, the sequence $\mathbf{T}_1, \mathbf{T}_2, \ldots$ for X constructed according to $\mathcal{R}$ fulfils:

- ➡ every non-literal formula occurrence in $\mathbf{T}_n$ is eventually expanded on each branch where it occurs
- ➡ every γ-formula in $\mathbf{T}_n$ has the γ-rule applied to it arbitrarily often on each branch where it occurs

Example the above described strategy $\mathcal{R}$ is not fair

Definition most general atomic closure substitution

let $\mathbf{T}$ be a tableau with branches $\tau_1, \ldots, \tau_n$; for each i, A_i and $\neg B_i$ are pairs of literals on τ_i; suppose σ is a mgu of the "unification problem" $\{A_1 = B_1, \ldots, A_n = B_n\}$; we call σ a most general atomic closure substitution

Completeness

Theorem Completeness

Let $\mathcal{R}$ be any fair tableau strategy. If X is a valid sentence of L, X has a tableau proof which fulfils:

- ➡ all tableau expansion rules applications come first and are according to rule $\mathcal{R}$
- ➡ a single tableau substitution rule follows, using a substitution σ that is a most general atomic closure substitution

Motivation Unification Free-Variable Tableaux Soundness & Completeness

Summary

- ➡ unification
- ➡ unification algorithm by transformation
- ➡ free-variable semantic tableaux
- ➡ refinements of free-variable tableaux
- ➡ tableau strategy, fairness
- ➡ soundness & completeness