Lecture 16

Pete Manolios
Northeastern

Coincidence Lemma

- Let $\mathscr{I}_1 = \langle A, a_1, \beta_1 \rangle$ be an S_1-interpretation and let $\mathscr{I}_2 = \langle A, a_2, \beta_2 \rangle$ be an S_2-interpretation (both have the same domain). Let $S = S_1 \cap S_2$.
  - 1. Let t be an S-term. If $\mathscr{I}_1$ and $\mathscr{I}_2$ agree on the S-symbols occurring in t and on the variables occurring in t, then $\mathscr{I}_1(t) = \mathscr{I}_2(t)$.
  - 2. Let ϕ be an S-formula. If $\mathscr{I}_1$ and $\mathscr{I}_2$ agree on the S-symbols and on the variables occurring free in ϕ, then $\mathscr{I}_1 \models \phi$ iff $\mathscr{I}_2 \models \phi$.
- Proof: By induction on S-terms and then on S-formulas
- This is a very useful lemma

Substitution

- Substituting *t* for *x* in ϕ yields ϕ', which says about *t* what ϕ says about *x*
- Consider ϕ = ∃*z* *z*+*z* ≡ *x*. Note that ⟨*N*,*β*⟩ ⊨ ϕ iff *β.x* is even
  - Replacing *x* by *y* gives, ϕ′ = ∃*zz*+*z*≡*y* ,where ⟨*N*,*β*⟩ ⊨ ϕ' iff *β.y* is even; good!
  - What about replacing *x* by *z*? This gives ϕ′ =∃*zz*+*z*≡*z*, but N ⊨ ϕ'; bad!
  - Have to deal with variable capture
  - The book provides a definition which replaces bound occurrences of *z* with a new variable in ϕ
- Theorem: For every term, *t*, $\mathscr{I}(t\frac{t_0\ldots t_r}{x_0\ldots x_r}) = \mathscr{I}\frac{\mathscr{I}(t_0)\ldots\mathscr{I}(t_r)}{x_0\ldots x_r}(t)$
- Theorem: For every formula, ϕ, $\mathscr{I} \models \phi\frac{t_0\ldots t_r}{x_0\ldots x_r}$ **iff** $\mathscr{I}\frac{\mathscr{I}(t_0)\ldots\mathscr{I}(t_r)}{x_0\ldots x_r} \models \phi$
- Theorem: If ϕ is Valid then so is $\phi\frac{t_0\ldots t_r}{x_0\ldots x_r}$

Formalization Examples

$\forall x Rxx$

$\forall x \forall y (Rxy) \Rightarrow (Ryx)$

$\forall x \forall y \forall z ((Rxy \wedge Ryz) \Rightarrow Rxz)$

Equivalence relations

$\langle \forall x :: xRx \rangle$

$\langle \forall x, y :: xRy \Rightarrow yRx \rangle$

$\langle \forall x, y, z :: xRy \wedge yRz \quad \Rightarrow \quad xRz \rangle$

The way I would write it

Define a new quantifier "there exists exactly one," written $\exists^{=1} x \phi$

Try it!

$\exists x (\phi \wedge \forall y (\phi \frac{y}{x} \Rightarrow x = y))$

Prenex Normal Form Example

For any FO ϕ, we can find an equivalent FO ψ where all quantifiers are to the left. Try it!

$$\langle \forall x :: P(x) \vee R(y)\rangle \Rightarrow \langle \exists y, \boxed{x} :: Q(y) \vee \neg\langle \exists x :: P(x) \wedge Q(x)\rangle\rangle$$

Constant propagation, remove vacuous quantifiers (x not free in body)

$$\langle \forall x :: P(x) \vee R(y)\rangle \boxed{\Rightarrow} \langle \exists y :: Q(y) \vee \boxed{\neg}\langle \exists x :: P(x) \wedge Q(x)\rangle\rangle$$

Convert to NNF (Negation Normal Form) by eliminating ⇒ , ≡ , **if**

$$\neg\langle \forall x :: P(x) \vee R(y)\rangle \vee \langle \exists y :: Q(y) \vee \langle \forall x :: \neg P(x) \vee \neg Q(x)\rangle\rangle$$

$$\langle \exists x :: \neg P(x) \wedge \neg R(y)\rangle \vee \langle \boxed{\exists y} :: Q(y) \vee \langle \boxed{\forall x} :: \neg P(x) \vee \neg Q(x)\rangle\rangle$$

Pull quantifiers to the left

$$\langle \exists x :: \neg P(x) \wedge \neg R(y)\rangle \vee \langle \exists y :: \langle \forall x :: Q(y) \vee \neg P(x) \vee \neg Q(x)\rangle\rangle$$

$$\langle \boxed{\exists z} :: (\neg P(z) \wedge \neg R(y)) \vee \langle \forall x :: Q(z) \vee \neg P(x) \vee \neg Q(x)\rangle\rangle\rangle$$

Merge exists, avoid variable capture

$$\langle \exists z :: \langle \forall x :: \underbrace{(\neg P(z) \wedge \neg R(y)) \vee Q(z) \vee \neg P(x) \vee \neg Q(x)}_{\text{matrix}}\rangle\rangle$$

Prenex Normal Form Algorithm

Constant propagation, remove vacuous quantifiers.

Start with the propositional logic algorithms and extend with:

$\langle \forall x :: \phi \rangle \;\equiv\; \phi$ **when** x **is not free in** ϕ

$\langle \exists x :: \phi \rangle \;\equiv\; \phi$ **when** x **is not free in** ϕ

Convert to NNF (Negation Normal Form) by eliminating $\Rightarrow$, $\equiv$, **if**

Start with the propositional logic algorithms and extend with:

$\neg\langle \forall x :: \phi \rangle \;\equiv\; \langle \exists x :: \neg\phi \rangle$

$\neg\langle \exists x :: \phi \rangle \;\equiv\; \langle \forall x :: \neg\phi \rangle$

Prenex Normal Form Algorithm

Constant propagation, remove vacuous quantifiers

Convert to NNF (Negation Normal Form) by eliminating $\Rightarrow$, $\equiv$, **if**

Pull quantifiers to the left (interesting part)

$\langle \forall x :: \phi \rangle \vee \psi \;\equiv\; \langle \forall x :: \phi \vee \psi \rangle$ **where x is not free in ψ**

$\psi \vee \langle \forall x :: \phi \rangle \;\equiv\; \langle \forall x :: \psi \vee \phi \rangle$ **where x is not free in ψ**

$\langle \exists x :: \phi \rangle \vee \psi \;\equiv\; \langle \exists x :: \phi \vee \psi \rangle$ **where x is not free in ψ**

$\psi \vee \langle \exists x :: \phi \rangle \;\equiv\; \langle \exists x :: \psi \vee \phi \rangle$ **where x is not free in ψ**

Similarly for conjunction, etc. Use substitution when x is free.

Minimizing the number of quantifiers is a good idea.

$\langle \forall x :: \phi \rangle \wedge \langle \forall y :: \psi \rangle \;\equiv\; \langle \forall z :: \phi\frac{z}{x} \wedge \psi\frac{z}{y} \rangle$ **where z is not free in LHS**

$\langle \exists x :: \phi \rangle \vee \langle \exists y :: \psi \rangle \;\equiv\; \langle \exists z :: \phi\frac{z}{x} \vee \psi\frac{z}{y} \rangle$ **where z is not free in LHS**

Skolem Normal Form Example

For any FO ϕ, we can find a universal ψ in an *expanded* language such that ϕ is satisfiable iff ψ is satisfiable. Try it!

$$\langle \exists x \ \langle \forall w \ \langle \exists y \ \langle \forall u, v \ \langle \exists z \ \phi(x, w, y, u, v, z) \rangle \rangle \rangle \rangle \rangle$$

First, PNF, and push existentials left (2nd order logic)

$$\langle \exists x, F_y \ \langle \forall w, u, v \ \langle \exists z \ \phi(x, w, F_y(w), u, v, z) \rangle \rangle \rangle$$

$$\langle \exists x, F_y, F_z \ \langle \forall w, u, v \ \phi(x, w, F_y(w), u, v, F_z(w, u, v)) \rangle \rangle$$

The key idea is the following equivalence

We need the axiom of choice for ping

$$\langle \exists \ldots \ \langle \forall x \ \langle \exists y \ \phi(\ldots, x, y) \rangle \rangle \rangle \ \equiv \langle \exists \ldots \ \langle \exists F_y \ \langle \forall x \ \phi(\ldots, x, F_y(x)) \rangle \rangle \rangle$$

This allows us to push existential quantifiers to the left

To get back to FO, note that

$$\textbf{Sat}\langle \exists \ldots \ \langle \forall x \ \langle \exists y \ \phi(\ldots, x, y) \rangle \rangle \rangle \ \ \textbf{iff} \ \ \textbf{Sat}\langle \forall x \ \phi(\ldots, x, F_y(x)) \rangle$$

So, to finish our example, we get, where c, F_y, F_z are new symbols

$$\langle \forall w, u, v \ \phi(c, w, F_y(w), u, v, F_z(w, u, v)) \rangle$$

Skolem Normal Form Algorithm

Convert formula to NNF

Notice that Skolemizing in arbitrary formulas doesn't work

$\langle \exists x\ P(x) \rangle \wedge \neg \langle \exists y\ P(y) \rangle$ $\langle \exists x\ P(x) \wedge \neg P(c) \rangle$ is not equisatisfiable

With NNF, we can apply Skolemization to any sub formula

$\langle \forall x, z\ \ x = z \vee \langle \exists y\ \ x \cdot y = 1 \rangle \rangle$ can be Skolemized as

$\langle \forall x, z\ \ x = z \vee x \cdot f(x) = 1 \rangle$ or we can convert to PNF

$\langle \forall x, z\ \langle \exists y\ \ x = z \vee x \cdot y = 1 \rangle \rangle$ and then Skolemize

$\langle \forall x, z\ \ x = z \vee x \cdot f(x, z) = 1 \rangle$ *order matters!*

So, it is better to Skolemize inside-out and then convert to PNF

Theorem: For any FO ϕ, we can find a universal ψ in an *expanded* language such that ϕ is satisfiable iff ψ is satisfiable. (From last slide)

Corollary: For any FO ϕ, we can find an existential ψ in an *expanded* language such that ϕ is valid iff ψ is valid (use ¬ϕ in above Theorem).