Lecture 15 - FO Resolution

Vaishnavi Sundararajan

COL703 - Logic for Computer Science

Recap: Unifiability

- A finite set of terms $T = \{t_i \mid 1 \leqslant i \leqslant n\}$ is said to be **unifiable** if there exists a θ (a **unifier** for T) such that $t_i\theta = t_j\theta$ for all $1 \leqslant i, j \leqslant n$.
- A substitution that is “less constrained” than another is said to be “more general”. Look for the most general unifier (mgu).
- Only two possible obstacles to unification:
  - Function clash (trying to unify $f(\ldots)$ with $g(\ldots)$ where $f \neq g$)
  - Occurs check (trying to unify x and t where $x \in \mathsf{vars}(t)$)
- If neither of these occurs, a set is unifiable!
- Apply transformations to get a system of equations in solved form
- Extract unifying substitution from this
- Algorithm always terminates, and is sound and complete.

Recap: Roadmap for resolution

- $\Gamma \vDash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ unsatisfiable
- Every sentence in FO has an equisatisfiable sentence in SCNF
- A sentence is unsatisfiable iff some finite set of ground instances of its qf subexpressions is unsatisfiable.
- Start with $\Gamma \cup \{\neg\varphi\}$ and get empty clause to show unsat.
- $\varphi = \forall x_1 x_2 \ldots x_n.\ [\psi]$ represented by clauses that denote qf CNF ψ
- Perform unification, eliminate literals **across one pair of clauses**
- Rename bound variables to keep variables across clauses distinct
- Unify as much as possible; multiple literals can cancel in one iteration (but only across one pair of clauses at a time)!

FO Resolution: Example

- φ: All men are mortal, and Socrates is a man
- Is "Socrates is mortal" logically entailed by the above?

FO Resolution: Example

- φ: All men are mortal, and Socrates is a man
- Is “Socrates is mortal” logically entailed by the above?
- What is the signature we need to formally write these statements?

FO Resolution: Example

- φ: All men are mortal, and Socrates is a man
- Is "Socrates is mortal" logically entailed by the above?
- What is the signature we need to formally write these statements?
- $\Sigma = (\{S\}, \emptyset, \{Man, Mortal\})$

FO Resolution: Example

- φ: All men are mortal, and Socrates is a man
- Is "Socrates is mortal" logically entailed by the above?
- What is the signature we need to formally write these statements?
- $\Sigma = (\{S\}, \emptyset, \{Man, Mortal\})$
- $\varphi = \forall x.\ [Man(x) \supset Mortal(x)] \wedge Man(S)$
- "S is mortal" $= Mortal(S)$
- Is it the case that $\forall x.\ [Man(x) \supset Mortal(x)] \wedge Man(S) \models Mortal(S)$?

FO Resolution: Example (contd.)

- Convert $\forall x.\ [\mathrm{Man}(x) \supset \mathrm{Mortal}(x)] \wedge \mathrm{Man}(S)$ to SCNF clauses
- φ denoted by clauses $\{\{\neg\mathrm{Man}(x), \mathrm{Mortal}(x)\}, \{\mathrm{Man}(S)\}\}$
- Resolve $\{\{\neg\mathrm{Man}(x), \mathrm{Mortal}(x)\}, \{\mathrm{Man}(S)\}, \{\neg\mathrm{Mortal}(S)\}\}$
- **Important**: Can always treat a sentence without quantifiers as being implicitly universally quantified
- Unify literals $\mathrm{Man}(x)$ and $\mathrm{Man}(S)$.
- This assigns the value S to x and yields $\{\{\mathrm{Mortal}(S)\}, \{\neg\mathrm{Mortal}(S)\}\}$
- Use propositional resolution to resolve this set of clauses, and get $\{\emptyset\}$

Example: Proof tree

$$\dfrac{\dfrac{\{\neg \mathrm{Man}(x), \mathrm{Mortal}(x)\} \qquad \{\mathrm{Man}(\mathrm{S})\}}{\{\mathrm{Mortal}(\mathrm{S})\}}\{S/x\} \qquad \{\neg \mathrm{Mortal}(\mathrm{S})\}}{\{\emptyset\}}\ \mathsf{res}$$

- Leaves are clauses which come directly from the original φ
- Each application of FO resolution marked by a unifier
- Might have to perform PL resolution
  - No variables/unification involved, and
  - One pair of contradictory literals eliminated
- Mark PL resolution by res, as earlier
- We will often omit the braces to improve readability

FO Resolution: Another example

- $X = \{\{P(x), R(x)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg P(w), Q(w)\}\}$
- Does $X \vDash \forall x. S(x)$?

FO Resolution: Another example

- $X = \{\{P(x), R(x)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg P(w), Q(w)\}\}$
- Does $X \vDash \forall x.\, S(x)$?
- Consider $X \cup \{\{\neg S(a)\}\}$, where a is a **constant** (**Exercise:** Why?)
- Unify $P(x)$ with $P(w)$, assign w to x
- Resolved clauses: $\{R(w), Q(w)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$

FO Resolution: Another example

- $X = \{\{P(x), R(x)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg P(w), Q(w)\}\}$
- Does $X \vDash \forall x. S(x)$?
- Consider $X \cup \{\{\neg S(a)\}\}$, where a is a **constant** (**Exercise:** Why?)
- Unify $P(x)$ with $P(w)$, assign w to x
- Resolved clauses: $\{R(w), Q(w)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$
- Unify $Q(w)$ with $Q(y)$, assign y to w
- Resolved clauses: $\{R(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$

FO Resolution: Another example

- $X = \{\{P(x), R(x)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg P(w), Q(w)\}\}$
- Does $X \vDash \forall x. S(x)$?
- Consider $X \cup \{\{\neg S(a)\}\}$, where a is a **constant** (**Exercise:** Why?)
- Unify $P(x)$ with $P(w)$, assign w to x
- Resolved clauses: $\{R(w), Q(w)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$
- Unify $Q(w)$ with $Q(y)$, assign y to w
- Resolved clauses: $\{R(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$
- Unify $R(y)$ with $R(z)$, assign z to y
- Resolved clauses: $\{S(u), S(z)\}, \{\neg S(a)\}$

FO Resolution: Another example

- $X = \{\{P(x), R(x)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg P(w), Q(w)\}\}$
- Does $X \vDash \forall x. S(x)$?
- Consider $X \cup \{\{\neg S(a)\}\}$, where a is a **constant** (**Exercise:** Why?)
- Unify $P(x)$ with $P(w)$, assign w to x
- Resolved clauses: $\{R(w), Q(w)\}, \{\neg Q(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$
- Unify $Q(w)$ with $Q(y)$, assign y to w
- Resolved clauses: $\{R(y), S(y)\}, \{\neg R(z), S(u), S(z)\}, \{\neg S(a)\}$
- Unify $R(y)$ with $R(z)$, assign z to y
- Resolved clauses: $\{S(u), S(z)\}, \{\neg S(a)\}$
- Unify $S(u)$ with $S(a)$ **and** $S(z)$ with $S(a)$, get $\emptyset$

FO Resolution: Proof tree

$$\dfrac{\dfrac{\dfrac{\dfrac{P(x), R(x) \quad \neg P(w), Q(w)}{R(w), Q(w)}\{w/x\} \quad \neg Q(y), S(y)}{R(y), S(y)}\{y/w\} \quad \neg R(z), S(u), S(z)}{S(u), S(z)}\{z/y\} \quad \neg S(a)}{\{\emptyset\}}\theta$$

where $\theta = \{a/u, a/z\}$

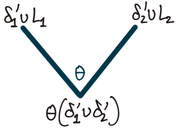
- Every application of resolution here involves unification
- Indicated by the unifier next to the rule
- Can we extract a general rule for FO resolution based on these examples?

FO Resolution: General rule

- Let δ_1, δ_2 be clauses s.t. $\mathsf{fv}(\delta_1) \cap \mathsf{fv}(\delta_2) = \emptyset$
- Let $P \in \mathscr{P}$ be a k-ary predicate symbol
- Let $L_1 = \left\{P(u_1, \ldots, u_k) \in \delta_1 \;\middle|\; u_1, \ldots, u_k \in \mathrm{T}(\Sigma)\right\}$ such that $\delta_1 = \delta_1' \cup L_1$
- Let $L_2 = \left\{\neg P(v_1, \ldots, v_k) \in \delta_2 \;\middle|\; v_1, \ldots, v_k \in \mathrm{T}(\Sigma)\right\}$ such that $\delta_2 = \delta_2' \cup L_2$
- Denote by $\overline{L}_2$ the set $\{P(v_1, \ldots, v_k) \in \delta_2 \mid v_1, \ldots, v_k \in \mathrm{T}(\Sigma)\}$
- Let $L_1 \cup \overline{L}_2$ be unifiable, with θ an mgu
- Apply the rule to premises δ_1 and δ_2
- The conclusion of the rule is the **resolvent** of δ_1 and δ_2

$$\frac{\delta_1' \cup L_1 \qquad \delta_2' \cup L_2}{\theta(\delta_1' \cup \delta_2')}\,\theta$$

Often drawn as



FO Resolution: Correctness

- Need to show **Soundness** and **Completeness** for the rule.
- Show for one application of the rule, and lift to larger proofs.
- What are we actually using resolution to show? Logical consequence.
- Enough to show that each application of the rule preserves logical consequence.

FO Resolution: Soundness

- **Soundness**: If one application of the resolution rule on δ_1 and δ_2 gives us δ, then $\delta_1 \cup \delta_2 \vDash \delta$.
- Consider some $\mathcal{I}$ such that $\mathcal{I} \vDash \delta_1 \cup \delta_2$.
- Then, $\mathcal{I} \vDash \forall \vec{x}_i. \left[\bigvee_{\ell \in \delta_i} \ell\right]$, for $i \in \{1, 2\}$
- Any substitution θ will map each x_{ij} to some term in $\mathrm{T}(\Sigma)$
- So $\mathcal{I} \vDash \theta\left(\bigvee_{\ell \in \delta_i} \ell\right)$ for $i \in \{1, 2\}$
- Suppose θ is a unifier of $L_1 \cup L_2$, and $(L_1 \cup L_2)\theta = \ell_\theta$. (Why ℓ and not L?)
- Then, we get $\mathcal{I} \vDash \bigvee\left(\{\ell_\theta\} \cup \delta_1'\theta\right)$ and $\mathcal{I} \vDash \bigvee\left(\{\neg\ell_\theta\} \cup \delta_2'\theta\right)$
- Let $\delta_1'\theta = \{\ell_i^1 \mid 1 \leqslant i \leqslant m_1\}$ and $\delta_2'\theta = \{\ell_i^2 \mid 1 \leqslant i \leqslant m_2\}$

FO Resolution: Soundness proof (contd.)

- $\delta_1'\theta = \{\ell_i^1 \mid 1 \leqslant i \leqslant m_1\}$ and $\delta_2'\theta = \{\ell_i^2 \mid 1 \leqslant i \leqslant m_2\}$
- Want to show that $\bigvee\{(\ell_\theta \cup \delta_1'\theta)\}, \bigvee\{(\neg\ell_\theta \cup \delta_2'\theta)\} \models \bigvee\{\delta_1'\theta \cup \delta_2'\theta\}$.
- Denote by α_i the expression $\bigvee(\delta_i'\theta)$ for $i \in \{1, 2\}$.
- Show that $(\ell_\theta \vee \alpha_1), (\neg\ell_\theta \vee \alpha_2) \models \alpha_1 \vee \alpha_2$.
- Suppose both δ_1' and δ_2' are empty. $m_1 = m_2 = 0$
  - Then, $\ell_\theta \vee \alpha_1 = \ell_\theta$, and $\neg\ell_\theta \vee \alpha_2 = \neg\ell_\theta$.
  - $\alpha_1 \vee \alpha_2$ is the empty disjunction, equivalent to $\ell_\theta \wedge \neg\ell_\theta$
  - $\ell_\theta, \neg\ell_\theta \models \ell_\theta \wedge \neg\ell_\theta$
- Suppose δ_1' is empty, but δ_2' is not. $m_1 = 0$ but $m_2 > 0$.
  - Then, $\ell_\theta \vee \alpha_1 = \ell_\theta$
  - Note that $\neg\ell_\theta \vee \alpha_2 \Leftrightarrow \ell_\theta \supset \alpha_2$
  - $\ell_\theta,\ \ell_\theta \supset \alpha_2 \models \alpha_2$

FO Resolution: Soundness proof (contd.)

- Similarly, when δ_1' is not empty, but δ_2' is, we get $\neg \ell_\theta, \neg \ell_\theta \supset \alpha_1 \models \alpha_1$
- Suppose δ_1' and δ_2' are both non-empty. $m_1, m_2 > 0$
  - Note that $\ell_\theta \vee \alpha_1 \Leftrightarrow \alpha_1 \vee \ell_\theta \Leftrightarrow \neg \alpha_1 \supset \ell_\theta$
  - Also note that $\neg \ell_\theta \vee \alpha_2 \Leftrightarrow \ell_\theta \supset \alpha_2$
  - $\neg \alpha_1 \supset \ell_\theta, \ell_\theta \supset \alpha_2 \models \neg \alpha_1 \supset \alpha_2$
  - Note that $\neg \alpha_1 \supset \alpha_2 \Leftrightarrow \alpha_1 \vee \alpha_2$, so we are done.

FO Resolution: Completeness

- **Completeness**: If a set S of clauses is unsatisfiable, then the empty clause is derivable from it.
- What happens if there are no variables in S? We just apply the propositional rule res.
- Completeness (ground clauses): Let S be a set of ground clauses. If S is not satisfiable, then res derives the empty clause from S.
- Proof is different now (we might eliminate multiple literals in one go) but enough to assume this and proceed.
- Need a “lifting lemma” which allows us to “lift” the derivation of empty clause by (ground) substitution instances to the derivation of empty clause by the original clauses themselves.

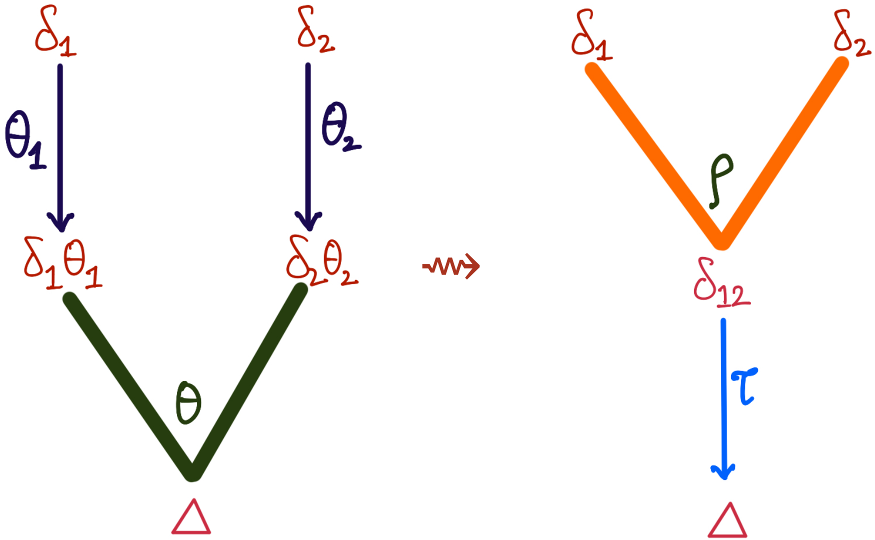
Lifting lemma

Lifting lemma: Let δ_1 and δ_2 be clauses with substitutions $\theta_1, \theta_2, \theta$ such that the following hold:

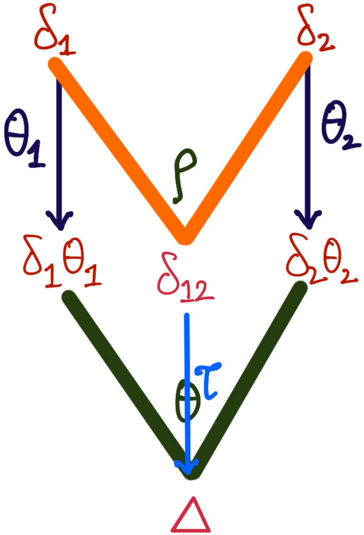
- $fv(\delta_1) \cap fv(\delta_2) = \emptyset$,
- $fv(\delta_1\theta_1) \cap fv(\delta_2\theta_2) = \emptyset$, and
- Δ is the resolvent of $\delta_1\theta_1$ and $\delta_2\theta_2$ obtained by a single application of the FO resolution rule, using unifier θ

Then, there exist a resolvent δ_{12} of δ_1 and δ_2 (obtained by a single application of the FO resolution rule, using unifier ρ) and a substitution τ such that Δ is equivalent to $\delta_{12}\tau$ upto variable renaming.

Lifting lemma: Pictorial representation



Lifting lemma: Pictorial representation



Lifting lemma: Example

Consider a signature $\Sigma = (\{a, b\}, \{f/1\}, \{P/1, Q/1, R/2\})$.

Let $\delta_1 = \{\neg P(x), Q(f(x))\}$ and $\delta_2 = \{\neg Q(y), R(f(y), z)\}$

Let $\ell_1 = Q(f(x))$ $\ell_2 = \neg Q(y)$ $\delta_1' = \{\neg P(x)\}$ $\delta_2' = \{R(f(y), z)\}$

Let $\theta_1 = \{x \mapsto f(f(a))\}$ and $\theta_2 = \{y \mapsto f(w), z \mapsto b\}$

$\delta_1\theta_1 = \{\neg P(f(f(a))), Q(f(f(f(a))))\}$ $\delta_2\theta_2 = \{\neg Q(f(w)), R(f(f(w)), b)\}$

The mgu for these is $\theta = \{w \mapsto f(f(a))\}$ and

$\Delta = \{\neg P(f(f(a))), R(f(f(f(f(a)))), b)\}$

Now, ℓ_1 and $\overline{\ell_2}$ also unify.

Lifting lemma: Example

Consider a signature $\Sigma = (\{a, b\}, \{f/1\}, \{P/1, Q/1, R/2\})$.

Let $\delta_1 = \{\neg P(x), Q(f(x))\}$ and $\delta_2 = \{\neg Q(y), R(f(y), z)\}$

Let $\ell_1 = Q(f(x))$ $\ell_2 = \neg Q(y)$ $\delta'_1 = \{\neg P(x)\}$ $\delta'_2 = \{R(f(y), z)\}$

Let $\theta_1 = \{x \mapsto f(f(a))\}$ and $\theta_2 = \{y \mapsto f(w), z \mapsto b\}$

$\delta_1\theta_1 = \{\neg P(f(f(a))), Q(f(f(f(a))))\}$ $\delta_2\theta_2 = \{\neg Q(f(w)), R(f(f(w)), b)\}$

The mgu for these is $\theta = \{w \mapsto f(f(a))\}$ and

$\Delta = \{\neg P(f(f(a))), R(f(f(f(f(a)))), b)\}$

Now, ℓ_1 and $\overline{\ell_2}$ also unify.

The mgu is $\rho = \{y \mapsto f(x)\}$, and $\delta_{12} = \{\neg P(x), R(f(f(x)), z)\}$.

$\Delta = \delta_{12}\tau$, where $\tau = \{x \mapsto f(f(a)), z \mapsto b\}$.

Lifting lemma: Proof

- Let $L_1 = \left\{P(u_1, \dots, u_k) \in \delta_1 \,\middle|\, u_1, \dots, u_k \in \mathrm{T}(\Sigma)\right\}$ such that $\delta_1 = \delta_1' \cup L_1$
- Let $L_2 = \left\{\neg P(v_1, \dots, v_k) \in \delta_2 \,\middle|\, v_1, \dots, v_k \in \mathrm{T}(\Sigma)\right\}$ such that $\delta_2 = \delta_2' \cup L_2$
- Let θ be an mgu of $L_1\theta_1 \cup \overline{L}_2\theta_2$ and $\Delta = (\delta_1'\theta_1 \cup \delta_2'\theta_2)\theta$.
- The domains and ranges of θ_1 and θ_2 are disjoint by assumption.
- So $\delta_1'\theta_1 = (\theta_1 \cup \theta_2)(\delta_1')$ and $\delta_2'\theta_2 = (\theta_1 \cup \theta_2)(\delta_2')$.
- Similarly, $L_1\theta_1 = (\theta_1 \cup \theta_2)(L_1)$ and $\overline{L}_2\theta_2 = (\theta_1 \cup \theta_2)(\overline{L}_2)$.
- θ is an mgu of $L_1\theta_1$ and $\overline{L}_2\theta_2$ (since we could apply resolution using θ)
- So $\theta \circ (\theta_1 \cup \theta_2)$ is a unifier for $L_1 \cup \overline{L}_2$.
- There is an mgu $\rho \succcurlyeq \theta \circ (\theta_1 \cup \theta_2)$ such that $\delta_{12} = \rho(\delta_1' \cup \delta_2')$ is the resolvent of δ_1 and δ_2.
- ρ is an mgu, so there is a τ such that $\tau \circ \rho = \theta \circ (\theta_1 \cup \theta_2)$.
- Thus, $\Delta = \tau(\rho(\delta_1' \cup \delta_2')) = (\theta \circ (\theta_1 \cup \theta_2))(\delta_1' \cup \delta_2')$.

FO Resolution: Completeness

- **Completeness**: If a set S of clauses is unsatisfiable, then the empty clause is derivable from it.
- By Herbrand's theorem, there exists an unsatisfiable $G = \{\gamma_i \mid 1 \leqslant i \leqslant m\} \subseteq_{\text{fin}} \Gamma^g(S)$.
- For every i, $\gamma_i = \delta_i\theta_i$ for $\delta_i \in S$ and some θ_i.
- By the lifting lemma, each application of res to clauses in G (which are of the form $\delta_i\theta_i$) can be lifted to finding an mgu for the δ_is.
- Need to do this for the entire proof tree.
- How do we lift the proof to the full tree?

FO Resolution: Completeness

- **Completeness**: If a set S of clauses is unsatisfiable, then the empty clause is derivable from it.
- By Herbrand's theorem, there exists an unsatisfiable $G = \{\gamma_i \mid 1 \leqslant i \leqslant m\} \subseteq_{\text{fin}} \Gamma^g(S)$.
- For every i, $\gamma_i = \delta_i\theta_i$ for $\delta_i \in S$ and some θ_i.
- By the lifting lemma, each application of res to clauses in G (which are of the form $\delta_i\theta_i$) can be lifted to finding an mgu for the δ_is.
- Need to do this for the entire proof tree.
- How do we lift the proof to the full tree? As always, induction.
- The proof is left as an **exercise**. (Convince yourself pictorially first!)