Lecture 17

Pete Manolios
Northeastern

Unification for FOL

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}'\subseteq C$, $\underline{D}'\subseteq D$ s.t. σ is a unifier for $\underline{C}'\cup\underline{D}'^-$ and $K=(C\backslash\underline{C}' \cup D\backslash\underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1

$C = \{\neg R(x), R(f(x))\}\ D = \{\neg R(f(f(x))), P(x)\}$ corresponds to

$\langle \forall x\ (\neg R(x) \vee R(f(x)))\ \wedge\ (\neg R(f(f(x))) \vee P(x))\rangle$ equivalent to

$\langle \forall x\ \neg R(x) \vee R(f(x))\rangle\ \wedge\ \langle \forall x\ \neg R(f(f(x))) \vee P(x)\rangle$ equivalent to

$\langle \forall x\ \neg R(x) \vee R(f(x))\rangle\ \wedge\ \langle \forall y\ \neg R(f(f(y))) \vee P(y)\rangle$ corresponds to

$C = \{\neg R(x), R(f(x))\}\ D = \{\neg R(f(f(y)) \vee P(y)\}$

so I will rename variables in clauses as I see fit

Recall from the Prenex Normal Form algorithm (let z,y be x in the example)

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

U-resolvent example

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^-$ and $K=(C \backslash \underline{C}' \cup D \backslash \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1

$$C = \{\neg R(x), R(f(x))\} \quad D = \{\neg R(f(f(x))), P(x)\}$$

$$\{\neg R(x), \underline{R(f(x))}\} \quad \{\underline{\neg R(f(f(y)))}, P(y)\}$$

$$\sigma = f(y) \leftarrow x$$

$$\{\neg R(f(y)), P(y)\}$$

U-resolvent example

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^-$ and $K=(C \backslash \underline{C}' \cup D \backslash \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- Try this: $C = \{\neg S(c,x), \neg S(x,x)\},\ D = \{S(x,x), S(c,x)\}$

One possible U-resolution step

$$\{\underline{\neg S(c,x)}, \neg S(x,x)\} \qquad \{\underline{S(c,y)}, S(y,y)\}$$

$$\sigma = x \leftarrow y$$

$$\{\neg S(x,x), S(x,x)\}$$

Tautology, so useless

U-resolvent example

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^-$ and $K=(C \backslash \underline{C}' \cup D \backslash \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- Try this: $C = \{\neg S(c,x), \neg S(x,x)\},\ D = \{S(x,x), S(c,x)\}$

$$\{\underline{\neg S(c,x)}, \neg S(x,x)\} \quad \{\underline{S(c,y)}, S(y,y)\} \quad \sigma = x \leftarrow y \quad \Rightarrow \quad \{\neg S(x,x), S(x,x)\}$$

$$\{\neg S(c,x), \underline{\neg S(x,x)}\} \quad \{\underline{S(c,y)}, S(y,y)\} \quad \sigma = c \leftarrow y, x \quad \Rightarrow \quad \{\neg S(c,c), S(c,c)\}$$

$$\{\underline{\neg S(c,x)}, \neg S(x,x)\} \quad \{S(c,y), \underline{S(y,y)}\} \quad \sigma = c \leftarrow y, x \quad \Rightarrow \quad \{\neg S(c,c), S(c,c)\}$$

$$\{\neg S(c,x), \underline{\neg S(x,x)}\} \quad \{S(c,y), \underline{S(y,y)}\} \quad \sigma = x \leftarrow y \quad \Rightarrow \quad \{\neg S(c,x), S(c,x)\}$$

All are tautologies
(useless)

U-resolvent example

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^-$ and $K=(C \setminus \underline{C}' \cup D \setminus \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- Try this: $C = \{\neg S(c,x), \neg S(x,x)\},\ D = \{S(x,x), S(c,x)\}$

$\{\underline{\neg S(c,x), \neg S(x,x)}\}$ $\{\underline{S(c,y)}, S(y,y)\}$ $\{\neg S(c,x), \underline{\neg S(x,x)}\}$ $\{\underline{S(c,y), S(y,y)}\}$

$\sigma = c \leftarrow y, x$ $\sigma = c \leftarrow y, x$

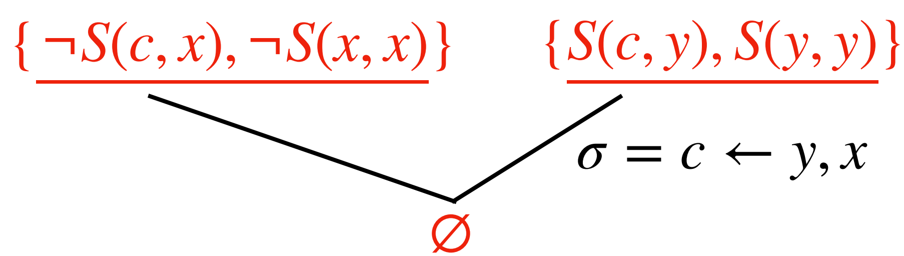
$\{S(c,c)\}$ $\{\neg S(c,c)\}$

$\sigma = \iota$ the identity substitution

$\emptyset$

U-resolvent example

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}'\subseteq C$, $\underline{D}'\subseteq D$ s.t. σ is a unifier for $\underline{C}'\cup\underline{D}'^-$ and $K=(C\backslash\underline{C}' \cup D\backslash\underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- Try this: $C = \{\neg S(c,x), \neg S(x,x)\},\ D = \{S(x,x), S(c,x)\}$



- This is the Barber of Seville problem: Prove that there is no barber who shaves all those, and those only, who do not shave themselves.

$$\neg\langle \exists b\ \langle \forall x\ S(b,x) \equiv \neg S(x,x)\rangle\rangle$$

Unification for FOL

- Let C be a clause; if we negate all literals in C, we get C^-
- A unifier for a clause $C=\{l_1,\ldots,l_n\}$ is a unifier for $\{(l_1,l_2), (l_2, l_3), \ldots, (l_{n-1},l_n)\}$
- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^-$ and $K=(C \backslash \underline{C}' \cup D \backslash \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- Lemma: Let C, D be clauses. Then
  - every resolvent of ground instances of C, D is a ground instance of a U-resolvent of C, D
  - every ground instance of a U-resolvent of C, D is a resolvent of ground instances of C, D
- Let $\mathcal{K}$ be a set of ground clauses, $Res(\mathcal{K})=\mathcal{K} \cup \{K \mid K \text{ is a resolvent of } C,D\in\mathcal{K}\}$
- Let $\mathcal{K}$ be a set of FO clauses, $URes(\mathcal{K})=\mathcal{K} \cup \{K \mid K \text{ is a U-resolvent of } C,D\in\mathcal{K}\}$
- Let $URes_0(\mathcal{K})=\mathcal{K}$, $URes_{n+1}(\mathcal{K})=URes(URes_n(\mathcal{K}))$, $URes_\omega(\mathcal{K})=\bigcup_{n\in\omega}URes_n(\mathcal{K})$

Unification for FOL

- Let C, D be clauses (assume there are no common variables since we can rename vars). K is a U-resolvent of C, D iff there are non-empty $\underline{C}' \subseteq C$, $\underline{D}' \subseteq D$ s.t. σ is a unifier for $\underline{C}' \cup \underline{D}'^{-}$ and $K = (C \setminus \underline{C}' \cup D \setminus \underline{D}')\sigma$. Note $|\underline{C}'|$, $|\underline{D}'|$ can be >1
- $G(K)$ is the set of ground instances of K, $G(\mathcal{K}) = \cup_{K \in \mathcal{K}} G(K)$
- Lemma: $Res_n(G(\mathcal{K})) = G(URes_n(\mathcal{K}))$ and $Res_\omega(G(\mathcal{K})) = G(URes_\omega(\mathcal{K}))$
- Lemma: $\emptyset \in Res_\omega(G(\mathcal{K}))$ iff $\emptyset \in URes_\omega(\mathcal{K})$
- For Φ a set of ∀ formulas in CNF: $G(\mathcal{K}(\Phi)) = \mathcal{K}(G(\Phi))$, where $\mathcal{K}(\Phi)$ is set-representation of CNF
- Theorem: For Φ a set of ∀ formulas in CNF, Φ is Sat iff $\emptyset \notin URes_\omega(\mathcal{K}(\Phi))$
  - Proof: Φ is Sat iff G(Φ) is (propositionally) Sat iff $\mathcal{K}(G(\Phi))$ is Sat iff $G(\mathcal{K}(\Phi))$ is Sat iff $\emptyset \notin Res_\omega G(\mathcal{K}(\Phi))$ iff $\emptyset \notin URes_\omega \mathcal{K}(\Phi)$

FOL Checking with Unification

- FO validity checker: Given FO ϕ, negate & Skolemize to get universal ψ s.t. Valid(ϕ) iff UNSAT(ψ). Let G be the set of ground instances of ψ (possibly infinite, but countable). Let G_1, G_2 …, be a sequence of finite subsets of G s.t. $\forall g \subseteq G, |g| < \omega$, $\exists n$ s.t. $g \subseteq G_n$. If $\exists n$ s.t. Unsat G_n, then Unsat ψ and Valid ϕ
- Unification: intelligently instantiate formulas
- FO validity checker w/ unification: Given FO ϕ, negate & Skolemize to get universal ψ s.t. Valid(ϕ) iff UNSAT(ψ). Convert ψ into equivalent CNF $\mathcal{K}$. Then, Unsat ψ iff $\emptyset \in URes_\omega(\mathcal{K})$ iff $\exists n$ s.t. $\emptyset \in URes_n(\mathcal{K})$.
- We say that U-resolution is *refutation-compete*: If Unsat($\mathcal{K}$) then there is a proof using U-resolution (*i.e.*, you can derive $\emptyset$), so we have a semi-decision procedure for validity.

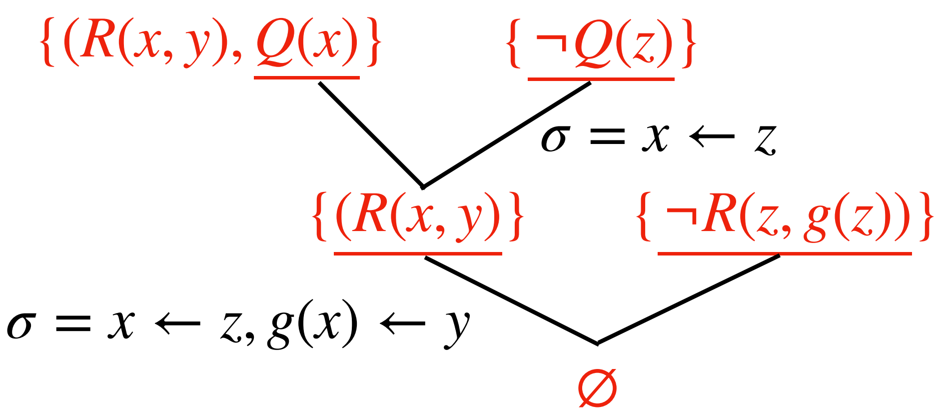
FOL Checking Examples

- FO validity checker w/ unification: Given FO ϕ, negate & Skolemize to get universal ψ s.t. Valid(ϕ) iff UNSAT(ψ). Convert ψ into equivalent CNF $\mathcal{K}$. Then, Unsat(ψ) iff $\emptyset \in \text{URes}_\omega(\mathcal{K})$ iff $\exists n$ s.t. $\emptyset \in \text{URes}_n(\mathcal{K})$.

$$\phi = \neg \langle \forall x, y \ (R(x,y) \vee Q(x)) \ \wedge \ \neg R(x, g(x)) \ \wedge \ \neg Q(y) \rangle$$

$$\psi = \langle \forall x, y \ (R(x,y) \vee Q(x)) \ \wedge \ \neg R(x, g(x)) \ \wedge \ \neg Q(y) \rangle$$

$$\mathcal{K} = \{\{R(x,y), Q(x)\}, \{\neg R(x, g(x))\}, \{\neg Q(y)\}\}$$



Let C, D be clauses (w/ no common variables). K is a U-resolvent of C, D iff there are non-empty $\underline{C'} \subseteq C$, $\underline{D'} \subseteq D$ s.t. σ is a unifier for $\underline{C'} \cup \underline{D'}^-$ and $K = (C \backslash \underline{C'} \cup D \backslash \underline{D'})\sigma$.

Recall

So, Unsat(ψ) and Valid(ϕ)

Subsumption & Replacement

- Let *C, D* be propositional clauses; $C \leq D$, *C subsumes D* if $C \subseteq D$, therefore $C \Rightarrow D$ and we can remove *D* and subsumed clauses
- Let *C, D* be FO clauses; $C \leq D$, *C subsumes D* if $\exists\sigma$ s.t. $C\sigma \subseteq D$ (matching!), hence $C \Rightarrow D$ and we so can remove D and subsumed clauses
- Theorem: For FO clauses, if $C \leq C'$ and $D \leq D'$ then any U-resolvent of *C*' and *D*' is subsumed by *C, D* or a U-resolvent of *C* and *D*.
- Corollary: If *C* is derivable by U-resolution, then $\exists C'$ derivable by U-resolution s.t. $C' \leq C$ and no clause is subsumed by any of its ancestors
- Corollary: If a U-resolution of a non-tautologous conclusion involves a tautology, ∃ a U-resolution proof that does not use any tautologies
- So, we can discard tautologies and subsumed clauses
  - Forward deletion: discard generated clauses that are subsumed by an existing clause
  - Backward replacement: if a generated clause subsumes an existing clause replace the existing clause with the newly generated one

Positive, Semantic Resolution

- *Positive resolution* (Robinson): Refutation completeness is preserved if we restrict resolution so that one of the clauses contains only positive literals
  - Hint: suppose that there are no positive clauses (all literals are positive), then the problem is SAT if you assign all atoms *false*; if there only positive clauses assign all atoms *true*; see proof in book
- Similarly for U-resolution
  - This cuts down the search space dramatically
  - This plays well with subsumption and replacement
- We could have required negative clauses (instead of positive clauses)
- More generally we have *semantic resolution*: if *S* is an Unsat set of FO clauses and *I* is an interpretation of the symbols used in *S*, there is a U-resolution proof of Unsat(*S*) where each U-resolution step involves a clause that is not true in *I*
  - Positive resolution is a special case where *I* assigns false to all atoms

Set of Support

- Partition T the input clauses into two disjoint sets, S, the *set of support* of T and the unsupported clauses U. Restrict U-resolution so that no two clauses in U are resolved together.
- Theorem: Let T be an Unsat set of clauses and let S be a subset of T where $T \backslash S$ is Sat; then there is a U-resolution proof of Usat(T) with set of support S
- Idea: focus U-resolution on finding resolvents that contribute to the solution
- For example say A is a set of standard mathematical axioms
  - You want to prove $B \Rightarrow C$
  - Using U-resolution you will want to derive the empty clause from A, B, $\neg C$
  - Since Sat(A) you can choose B, $\neg C$ as the set of support
  - Since A, B are Sat (presumably), you can choose $\neg C$ as the set of support
  - Suppose $\neg C$ is the only negative clause, then similar to negative resolution, but negative resolution is more restrictive; however, set of support often makes up for this by finding shorter proofs