CS389L: Automated Logical Reasoning

Lecture 9: First-Order Resolution

Işıl Dillig

Review

- What is a unifier?

- What is Prenex Normal Form?

- What is Skolem Normal Form?

- How do you convert formula to Clausal Normal Form?

Clausal Normal Form Example

- Convert formula to clausal form:

$$\exists w.\forall x.((\exists z.q(w,z)) \to \exists y.(\neg p(x,y) \wedge r(y)))$$

- **Step 1,2a:** No free variables, convert to NNF:

$\exists w.\forall x.(\neg(\exists z.q(w,z)) \vee \exists y.(\neg p(x,y) \wedge r(y)))$　remove $\to$
$\exists w.\forall x.((\forall z.\neg q(w,z)) \vee \exists y.(\neg p(x,y) \wedge r(y)))$　push negations

- **Step 2b:** Move quantifiers out (necessary for PNF):

$$\exists w.\forall x.\exists y.\forall z.((\neg q(w,z)) \vee (\neg p(x,y) \wedge r(y)))$$

1

Example, cont

▶ In Prenex Normal Form:

$$\exists w.\forall x.\exists y.\forall z.((\neg q(w,z)) \lor (\neg p(x,y) \land r(y)))$$

▶ Step 3a: Now, skolemize w:

$$\forall x.\exists y.\forall z.((\neg q(c,z)) \lor (\neg p(x,y) \land r(y)))$$

▶ Step 3b: Skolemize y:

$$\forall x.\forall z.((\neg q(c,z)) \lor (\neg p(x,f(x)) \land r(f(x))))$$

Example, cont

▶ In Skolem Normal Form:

$$\forall x.\forall z.((\neg q(c,z)) \lor (\neg p(x,f(x)) \land r(f(x))))$$

▶ Step 4: Convert inner formula to CNF

$$\forall x.\forall z.(\neg q(c,z) \lor \neg p(x,f(x))) \land (\neg q(c,z) \lor r(f(x)))$$

▶ Step 5: Drop universal quantifiers:

$$(\neg q(c,z) \lor \neg p(x,f(x))) \land (\neg q(c,z) \lor r(f(x)))$$

▶ Step 6: Finally, write formula as a set of clauses

$$\{\neg q(c,z), \neg p(x,f(x))\}$$
$$\{\neg q(c,z), r(f(x))\}$$

A Word About Clausal Form

▶ Consider the clausal form $\{l_1, l_2, \ldots l_k\}, \ldots, \{l'_1, l'_2, \ldots, l'_n\}$

▶ Assuming clauses contain variables $x_1, \ldots x_n$, what is the meaning of this clausal form as a proper FOL formula?

▶ $\forall x_1, \ldots, x_n. (l_1 \lor l_2 \ldots \lor l_k) \land \ldots \land (l'_1 \lor l'_2 \ldots \lor l'_n)$

▶ Recall: Universal quantifiers distribute over conjuncts:

$$\forall \vec{x}. F_1 \land F_2 \iff \forall \vec{x} F_1 \land \forall \vec{x} F_2$$

▶ Thus above formula is equivalent to:

$$\forall x_1, \ldots, x_n. (l_1 \lor l_2 \ldots \lor l_k) \ldots \land$$
$$\forall x_1, \ldots, x_n. (l'_1 \lor l'_2 \ldots \lor l'_n)$$

2

A Word About Clausal Form, cont.

$$\forall x_1, \ldots, x_n. \, (l_1 \vee l_2 \ldots \vee l_k) \ldots \wedge$$
$$\forall x_1, \ldots, x_n. (l'_1 \vee l'_2 \ldots \vee l'_n)$$

▶ Recall: If we rename quantified variables, the resulting formula is equivalent to original one

$$\forall x. F \; \Leftrightarrow \; \forall y. F[y/x]$$

▶ Hence, the above formula is equivalent to:

$$\forall x_1, \ldots, x_n. \, (l_1 \vee l_2 \ldots \vee l_k) \ldots \wedge$$
$$\forall y_1, \ldots, y_n. (l'_1 \vee l'_2 \ldots \vee l'_n)[\vec{y}/\vec{x}]$$

▶ Thus, if two different clauses C_1 and C_2 contain same variable x, we can rename x to some other x' in one of C_1 or C_2

Clausal Form and Renaming Variables

▶ In rest of lecture, we assume that we rename variables in each clause so different clauses contain different variables.

▶ This is necessary to ensure that we don't get conflicting names as we do resolution.

▶ For instance, if we have two clauses $\{p(a, x)\}$ and $\{\neg p(x, b)\}$, we assume they are renamed as $\{p(a, x)\}$ and $\{\neg p(z, b)\}$

First Order Resolution

▶ To apply first-order resolution, convert formula to clausal form

▶ Rename variables to ensure each clause contains different variables

▶ Resolution:

$$\frac{\{A, B_1, \ldots, B_k\} \quad \{\neg C, D_1, \ldots, D_n\}}{\{B_1, \ldots, B_k, D_1, \ldots, D_n\}\sigma} \quad (\sigma = mgu(A, C))$$

3

Example

Resolution:

$$\frac{\{A, B_1, \ldots, B_k\} \quad \{\neg C, D_1, \ldots, D_n\}}{\{B_1, \ldots, B_k, D_1, \ldots, D_n\}\sigma} \quad (\sigma = mgu(A, C))$$

▶ What is the result of performing resolution on the following clauses?

$$\begin{aligned} \text{Clause 1}: \quad & \{p(a, y), r(g(y))\} \\ \text{Clause 2}: \quad & \{\neg p(x, f(x)), q(g(x))\} \end{aligned}$$

▶ Mgu for $p(a, y)$ and $p(x, f(x))$:

▶ Resolvent:

Intuition about First-Order Resolution

▶ Intuition: Consider two clauses: $\{happy(x), sad(x)\}$ and $\{\neg happy(joe), happy(sally)\}$

▶ The first clause says:

▶ This implies: $happy(joe) \lor sad(joe)$

▶ The second clause says:

▶ Two possibilities: Either Joe is happy or not.

▶ If $happy(joe)$, second clause implies $happy(sally)$

▶ If $\neg happy(joe)$, then we have $sad(joe)$

▶ In either case, we have $happy(sally) \lor sad(joe)$

Intuition about First-Order Resolution, cont.

$$\frac{\{A, B_1, \ldots, B_k\} \quad \{\neg C, D_1, \ldots, D_k\}}{\{B_1, \ldots, B_k, D_1, \ldots, D_k\}\sigma} \quad (\sigma = mgu(A, C))$$

▶ What happens if we apply resolution to $\{happy(x), sad(x)\}$ and $\{\neg happy(joe), happy(sally)\}$?

▶ Instantiate resolution rule with our clauses:

$$\frac{\{happy(x), sad(x)\} \quad \{\neg happy(joe), happy(sally)\}}{\{sad(x), happy(sally)\}[x \mapsto joe]\{sad(joe), happy(sally)\}}$$

▶ Same conclusion as before!

Intuition about First-Order Resolution, summary

- Just like propositional resolution, first-order resolution corresponds to a simple case analysis

- But more involved due to universal quantifiers

- To perform deduction, often need to instantiate universal quantifier with something specific like joe

- The use of unifiers in resolution corresponds to instantiation of universally quantifiers

- Quantifier instantiation is demand-driven; we only unify when it is possible to perform deduction

Why Most General Unifiers?

- Why do we need most general unifiers, not just any unifier?

-

- Example: Consider clauses: $\{happy(x), sad(x)\}$ $\{\neg sad(y)\}$

- Most general unifier:

- Resolvent:

- What does this mean in English?

Why Most General Unifiers?

Clauses: $\{happy(x), sad(x)\}$ $\{\neg sad(y)\}$

- Now, suppose we use a less general unifier, e.g. $[x \mapsto joe, y \mapsto joe]$

- Resolvent:

- Since "Everyone is happy" implies "Joe is happy", former deduction is much better!

- Using most general unifiers ensures our deductions are as general as possible

Incompleteness

- The inference rule for resolution so far is sound,but not complete: there are valid deductions it cannot derive.

- Consider the following clauses:

$$\text{Clause 1}:\{p(x), p(y)\}$$
$$\text{Clause 2}:\{\neg p(a), \neg p(b)\}$$

- What does the first clause say?

- Simpler way of saying the same thing:

- Clearly contradicts the second clause!

- So, we should derive the empty clause, i.e., contradiction

Incompleteness Example

- What can we deduce using resolution from these clauses?

$$\text{Clause 1}:\{p(x), p(y)\}$$
$$\text{Clause 2}:\{\neg p(a), \neg p(b)\}$$

- Using mgu for $p(x)$ and $p(a)$,

- Using mgu for $p(x)$ and $p(b)$,

- Using mgu for $p(y), p(a)$,

- Using mgu for $p(y), p(b)$,

- More deductions possible using new clauses, but redundant

- Conclusion: Using inference rule for resolution alone, we cannot derive the empty clause

Solution: Factoring

- To ensure we can deduce all valid facts, we need another inference rule for factoring.

- Factorization:

$$\frac{\{A, B, C_1 \ldots, C_k\}}{\{A, C_1, \ldots C_k\}\sigma} \quad (\sigma = mgu(A, B))$$

- Soundness of factorization: For any clause C and any substitution σ, $C\sigma$ is always a valid deduction

- Why?

Revisiting the Example

▶ Consider again the problematic example:

$$\text{Clause 1}: \{p(x), p(y)\}$$
$$\text{Clause 2}: \{\neg p(a), \neg p(b)\}$$

▶ Use factoring on first clause

▶ Mgu for $p(x)$ and $p(y)$:

▶ Result of factoring:

▶ Now, do resolution between clause 2 and 3.

Resolution with Implicit Factoring

▶ Can formulate resolution and factoring as single inference rule.

▶ Resolution with Implicit Factorization:

$$\frac{\{A_1, \ldots A_n, B_1, \ldots, B_k\} \quad \{\neg C, D_1, \ldots, D_k\}}{\{B_1, \ldots, B_k, D_1, \ldots, D_k\}\sigma} \quad (\sigma = mgu(A_1, \ldots A_n, C))$$

▶ From now on, by "resolution", we mean resolution with implicit factorization

Resolution with Implicit Factoring Example

▶ Consider the example we looked at before:

$$\frac{\{p(x), p(y)\} \quad \{\neg p(a), \neg p(b)\}}{\{\neg p(b)\}} \quad (? = mgu(p(x), p(y), p(a)))$$

▶ Now, apply resolution with implicit factoring one more time:

$$\frac{\{p(x), p(y)\} \quad \{\neg p(b)\}}{\{\}} \quad (? = mgu(p(x), p(y), p(b)))$$

7

Resolution Derivation

▶ A clause C is derivable from a set of clauses Δ if there is a sequence of clauses $\Psi_1, \ldots, \Psi_k$ terminating in C such that:

    1. $\Psi_i \in \Delta$, or

    2. Ψ_i is resolvent of some Ψ_j and Ψ_k such that $j < i \land k < i$

▶ Example: Consider clauses

$$\Delta = \{happy(x), sad(x)\}, \{\neg sad(y)\}$$

▶ Here, $\{happy(x)\}$ is derivable from Δ

▶ If a clause C is derivable from Δ, we write $\Delta \vdash C$

Resolution Refutation

▶ The derivation of the empty clause from a set of clauses Δ is called resolution refutation of Δ

▶ Consider set of clauses Δ:

$$\{happy(x), sad(x)\}$$
$$\{\neg sad(y)\}$$
$$\{\neg happy(mother(joe))\}$$

▶ Resolution refutation of Δ:

$$\frac{\dfrac{\{happy(x), sad(x)\} \quad \{\neg sad(y)\}}{\{happy(x)\}} \quad \{\neg happy(mother(joe))\}}{\{\,\}}$$

Refutational Soundness and Completeness

▶ Theorem: Resolution is sound, i.e., if $\Delta \vdash C$, then $\Delta \models C$

▶ Corollary: If there is a resolution refutation of Δ, Δ is indeed unsatisfiable

▶ In other words, we cannot conclude a satisfiable formula is unsatisfiable using resolution

▶ Resolution with implicit factorization is also complete, i.e., if $\Delta \models C$, then $\Delta \vdash C$

▶ Corollary: If F is unsatisfiable, then there exists a resolution refutation of F using only resolution with factorization.

▶ This is called the refutational completeness of resolution.

Validity Proofs using Resolution

- How to prove validity FOL formula using resolution?

- Use duality of validity and unsatisfiability:

$$\boxed{F \text{ is valid iff } \neg F \text{ is unsatisfiable}}$$

- We will use resolution to show $\neg F$ is unsatisfiable.

- First, convert $\neg F$ to clausal form C.

- If there is a resolution refutation of C, then, by soundness, F is valid.

Example

- Everybody loves somebody. Everybody loves a lover. Prove that everybody loves everybody.

- First sentence in FOL:

- Second sentence in FOL:

- Goal in FOL:

- Thus, want to prove validity of:

$$(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.((\exists v.loves(u, v)) \rightarrow loves(w, u)))$$
$$\rightarrow \forall z.\forall t.loves(z, t)$$

Example, cont.

- Want to prove negation unsatisfiable:

$$\neg((\forall x.\exists y.loves(x, y) \land \forall u.\forall w.((\exists v.loves(u, v)) \rightarrow loves(w, u)))$$
$$\rightarrow \forall z.\forall t.loves(z, t))$$

- Convert to PNF: in NNF, quantifiers in front

- Remove inner implication:

$$\neg((\forall x.\exists y.loves(x, y) \land \forall u.\forall w.((\neg(\exists v.loves(u, v))) \lor loves(w, u)))$$
$$\rightarrow \forall z.\forall t.loves(z, t))$$

- Remove outer implication:

$$\neg(\neg(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.((\neg(\exists v.loves(u, v))) \lor loves(w, u))$$
$$\lor \forall z.\forall t.loves(z, t))$$

9

Example, cont.

$$\neg(\neg(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.((\neg(\exists v.loves(u, v))) \lor loves(w, u)))$$
$$\lor \forall z.\forall t.loves(z, t))$$

- ► Push innermost negation in:

$$\neg(\neg(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.(\neg loves(u, v) \lor loves(w, u))$$
$$\lor \forall z.\forall t.loves(z, t))$$

- ► Push outermost negation in:

$$(\neg\neg(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.\neg loves(u, v)) \lor loves(w, u))$$
$$\land \neg(\forall z.\forall t.loves(z, t)))$$

Example, cont.

$$(\neg\neg(\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.\neg loves(u, v) \lor loves(w, u))$$
$$\land \neg(\forall z.\forall t.loves(z, t)))$$

- ► Eliminate double negation:

$$((\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.\neg loves(u, v) \lor loves(w, u))$$
$$\land \neg(\forall z.\forall t.loves(z, t)))$$

- ► Push negation on second line in:

$$((\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.\neg loves(u, v) \lor loves(w, u))$$
$$\land (\exists z.\exists t.\neg loves(z, t)))$$

Example, cont.

$$((\forall x.\exists y.loves(x, y) \land \forall u.\forall w.\forall v.(\neg loves(u, v) \lor loves(w, u)))$$
$$\land (\exists z.\exists t.\neg loves(z, t)))$$

- ► Now, move quantifiers to front. Restriction:

$$\exists z.\exists t.\forall x.\exists y.\forall u.\forall w.\forall v.$$
$$loves(x, y) \land (\neg loves(u, v) \lor loves(w, u)) \land \neg loves(z, t)$$

- ► Next, skolemize existentially quantified variables:

$$\forall u.\forall w.\forall v.\forall x.$$
$$loves(x, lover(x)) \land (\neg loves(u, v) \lor loves(w, u))$$
$$\land \neg loves(joe, jane)$$

10

Example, cont.

$$\forall u.\forall w.\forall v.\forall x.$$
$$loves(x, lover(x)) \land (\neg loves(u, v) \lor loves(w, u))$$
$$\land \neg loves(joe, jane)$$

▶ Now, drop quantifiers:

$$loves(x, lover(x)) \land (\neg loves(u, v) \lor loves(w, u))$$
$$\land \neg loves(joe, jane)$$

▶ Convert to CNF: already in CNF!

▶ In clausal form:

$$\{loves(x, lover(x))\}$$
$$\{\neg loves(u, v), loves(w, u)\}$$
$$\{\neg loves(joe, jane)\}$$

Example, cont.

▶ Finally, we can do resolution:

$$\{loves(x, lover(x))\}$$
$$\{\neg loves(u, v), loves(w, u)\}$$
$$\{\neg loves(joe, jane)\}$$

▶ Resolve first and second clauses. MGU:

▶ Resolvent:

▶ Resolve new clause with third clause.

▶ Mgu:

▶ Resolvent: {}

▶ Thus, we have proven the formula valid.

Example II

▶ Use resolution to prove validity of formula:

$$\neg(\exists y.\forall z.(p(z, y) \leftrightarrow \neg \exists x.(p(z, x) \land p(x, z))))$$

▶ Convert negation to clausal form:

$$\exists y.\forall z.(p(z, y) \leftrightarrow \neg \exists x.(p(z, x) \land p(x, z)))$$

▶ To convert to NNF, get rid of ↔:

$$\exists y.\forall z.(\neg p(z, y) \lor \neg \exists x.(p(z, x) \land p(x, z)) \land$$
$$(p(z, y) \lor \exists x.(p(z, x) \land p(x, z))))$$

Example II, cont

$$\exists y. \forall z. (\neg p(z, y) \vee \neg \exists x. (p(z, x) \wedge p(x, z)) \wedge$$
$$(p(z, y) \vee \exists x. (p(z, x) \wedge p(x, z))))$$

- ▶ Push negations in:

$$\exists y. \forall z. (\neg p(z, y) \vee \forall x. (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$(p(z, y) \vee \exists x. (p(z, x) \wedge p(x, z))))$$

- ▶ Rename quantified variables:

$$\exists y. \forall z. (\neg p(z, y) \vee \forall x. (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee \exists w. (p(z, w) \wedge p(w, z)))$$

Example II, cont.

$$\exists y. \forall z. (\neg p(z, y) \vee \forall x. (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee \exists w. (p(z, w) \wedge p(w, z)))$$

- ▶ In PNF:

$$\exists y. \forall z. \exists w. \forall x. (\neg p(z, y) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee (p(z, w) \wedge p(w, z)))$$

- ▶ Skolemize existentials:

$$\forall z. \forall x. (\neg p(z, a) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, a) \vee (p(z, f(z)) \wedge p(f(z), z)))$$

Example II, cont.

$$\forall z. \forall x. (\neg p(z, a) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, a) \vee (p(z, f(z)) \wedge p(f(z), z)))$$

- ▶ Drop quantifiers and convert to CNF:

$$(\neg p(z, a) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, a) \vee p(z, f(z)) \wedge$$
$$p(z, a) \vee p(f(z), z))$$

- ▶ In clausal form (with renamed variables):

$$C1 : \ \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2 : \ \{p(y, a), p(y, f(y))\}$$
$$C3 : \ \{p(w, a), p(f(w), w))\}$$

Example II, cont.

$$C1 : \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2 : \{p(y, a), p(y, f(y))\}$$
$$C3 : \{p(w, a), p(f(w), w))\}$$

▶ Resolve $C1$ and $C2$ using factoring.

▶ What is the MGU for $p(z, a), p(z, x), p(x, z), p(y, a)$?

▶ Resolvent:

Example II, cont.

$$C1 : \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2 : \{p(y, a), p(y, f(y))\}$$
$$C3 : \{p(w, a), p(f(w), w))\}$$
$$C4 : \{p(a, f(a))\}$$

▶ Now, resolve $C1$ and $C3$ (using factoring).

▶ What is the MGU for $p(z, a), p(z, x), p(x, z), p(w, a)$?

▶ Resolvent:

Example II, cont.

$$C1 : \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2 : \{p(y, a), p(y, f(y))\}$$
$$C3 : \{p(w, a), p(f(w), w))\}$$
$$C4 : \{p(a, f(a))\}$$
$$C5 : \{p(f(a), a)\}$$

▶ Resolve $C1$ and $C5$ (using factoring).

▶ What is the MGU of $p(z, a), p(z, x)$ and $p(f(a), a)$?

▶ Resolvent:

13

Example II, cont.

$$C1 : \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2 : \{p(y, a), p(y, f(y))\}$$
$$C3 : \{p(w, a), p(f(w), w))\}$$
$$C4 : \{p(a, f(a))\}$$
$$C5 : \{p(f(a), a)\}$$
$$C6 : \{\neg p(a, f(a))\}$$

▶ Finally, resolve $C4$ and $C6$.

▶ Resolvent: $\{\}$

▶ Thus, the original formula is valid.

Summary

▶ First-order theorem provers work by converting to clausal form and trying to find resolution refutation

▶ But there are no termination guarantees – may diverge if formula is satisfiable

▶ Next lecture: First-order theories

14