CS228 Logic for Computer Science 2021

Lecture 19: FOL Resolution

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2021-03-05

Topic 19.1

Refutation proof systems

Recall: derivations starting from CNF

We have a set of formulas in the lhs, which is viewed as the conjunction of the formulas.

$$\Sigma \vdash F$$

The conjunction of CNF formulas is also a CNF formula.

If all formulas are in CNF, we may assume Σ as a set of clauses.

Recall: refutation proof system

Let us suppose we are asked to derive $\Sigma \vdash F$.

We assume Σ is finite. We can relax this due to compactness of FOL.

We will convert $\bigwedge \Sigma \wedge \neg F$ into a set of FOL clauses Σ'.

We apply the a refutation proof method on Σ'.

If we derive $\bot$ clause, $\Sigma \vdash F$ is derivable.

Topic 19.2

Unification and resolution

Applying resolution in FOL

We apply resolution when an atom and its negation are in two clauses.

$$\textsc{Resolution}\frac{F \vee C \quad \neg F \vee D}{C \vee D}$$

A complication: we may have terms in the FOL atoms with variables.

We can make two terms equal by substitutions.

Example 19.1

Consider two clauses $P(x, f(y)) \vee C$ *and* $\neg P(z, z) \vee D$

We may be able to make $P(x, f(y))$ *and* $P(z, z)$ *equal by unification.*

Three issues with unification

Before looking at the proof rules, we need a clear understanding of the following three issues.

1. Did we learn about unifying atoms?
2. Is substitution a valid operation for derivations?
3. How do we handle variables across clauses?

Issue 1: unification of atoms

We can lift the idea of unifying terms to atoms.

Simply, treat a predicate as a function.

Example 19.2

Consider atoms $P(x, f(y))$ *and* $P(z, z)$.

We can unify them using mgu $\sigma = \{x \mapsto f(y), z \mapsto f(y)\}$.

We obtain

- $P(x, f(y))\sigma = P(f(y), f(y))$
- $P(z, z)\sigma = P(f(y), f(y))$

Commentary: The following simultaneous application of ∀-Elim needs proving as derived rule. We presented it as an exercise in the formal proofs lecture.

Issue 2: deriving from substitution?

We know that the following derivation is valid if Σ is a set of sentences.

1. $\Sigma \vdash \forall x, y.F(x, y)$ — Premise
2. $\Sigma \vdash F(t_1(x, y), t_2(x, y))$ — ∀-Elim
3. $\Sigma \vdash \forall x, y.\ F(t_1(x, y), t_2(x, y))$ — ∀-Intro

Therefore the following derivations in our clauses are sound

$$\frac{C}{C\sigma}\sigma \text{ is a substitution.}$$

Example 19.3

The following derivation is a valid derivation

$$\frac{P(x) \vee Q(y)}{P(x) \vee Q(x)}\sigma = \{y \mapsto x\}$$

Issue 3: variables across clauses are not the same

Recall: universal quantifiers distribute over conjunction.

So we can easily distribute the quantifiers and scope only each clauses.

Example 19.4

Consider $\forall w.\forall y.(R(f(w), y) \wedge \neg R(w, c))$.
After the distribution the formula appears as follows,

$$\forall w.\forall y.R(f(w), y) \wedge \forall w.\forall y.\neg R(w, c)$$

Therefore, we may view the variables occurring in different clauses as different variables. Even if we use the same name.

Source of confusion. Pay attention!

Topic 19.3

Resolution theorem proving

Resolution theorem proving

Commentary: Please note that the consequences are also statements in the proof system. We simply do not write "$F \vdash$" repeatedly, because the left hand side does not change over the course of a resolution proof.

Input: a set of FOL clauses F

Inference rules:

$$\textsc{Assumption}\frac{}{C}\, C \in F$$

$$\textsc{Resolution}\frac{\neg A \vee C \quad B \vee D}{(C \vee D)\sigma}\, \sigma = mgu(A, B)$$

Example: resolution proof

Example 19.5

Consider statement $\emptyset \vdash (\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$.

We translate $\neg(\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ *into the following FOL CNF*

$$R(f(w), y) \wedge \neg R(w, c)$$

Note that w *and* w *in both clauses are different variables.*

We apply resolution.

$$\textsc{Resolution}\frac{R(f(w), y) \quad \neg R(w, c)}{\bot\sigma}\sigma = \{w \mapsto f(w), y \mapsto c\}$$

Therefore, $\neg(\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ *is unsat, i.e,* $\emptyset \vdash (\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ *holds.*

Example: resolution with unification

Example 19.6

Consider two clauses $P(x, y) \vee Q(y)$ *and* $\neg P(x, x) \vee R(f(x))$.

x *within a clause should be treated as same variable.*

If we unify $P(x, y)$ *and* $\neg P(x, x)$*, we obtain most general unifier* $\{x \mapsto x, y \mapsto x\}$.

Therefore,

$$\textsc{Resolution}\frac{P(x, y) \vee Q(y) \qquad \neg P(x, x) \vee R(f(x))}{Q(x) \vee R(f(x))}\sigma = \{x \mapsto x, y \mapsto x\}$$

Commentary: In exams, use $x_1, x_2, \ldots$ to indicated the variables in different clauses. The evaluation will be easier for us and you will be less likely to make mistake.

Why MGU? – not just any unifier

Commentary: Most general consequences allow the maximum opportunity of unifications later, thereby allow us to apply resolution in the maximum possible ways. Therefore, we have have maximum opportunity of finding the empty clause.

MGU keeps maximum generality in the consequence

Example 19.7

We may derive the following, using a σ that is not mgu of $P(x, y)$ and $P(x, x)$.

$$\textsc{Resolution}\frac{P(x,y) \vee Q(y) \qquad \neg P(x,x) \vee R(f(x))}{Q(d) \vee R(f(d))}\sigma = \{x \mapsto d, x \mapsto d, y \mapsto d\}$$

The above conclusion can always be derived from the mgu consequence

$$\frac{\dfrac{P(x,y) \vee Q(y) \qquad \neg P(x,x) \vee R(f(x))}{Q(x) \vee R(f(x))}\sigma = \{x \mapsto x, y \mapsto x\}}{Q(d) \vee R(f(d))}\sigma = \{x \mapsto d\}$$

Once a clause becomes specific, we can not go back. Why not keep it general?

Resolution theorem proving : factoring

A clause may have copies of facts that can be unified.

Commentary: Factoring may appear as a superfluous rule. A bad note in an otherwise beautiful symphony of logic. However, factoring is essential for FOL completeness. It captures the idea that we can express same concept in many possible ways. Sometime, a prover needs to recognize the situation and identify the similarities.

We need a rule that allows us to simplify clauses.

$$\textsc{Factor}\frac{L_1 \vee .. \vee L_k \vee C}{(L_1 \vee C)\sigma}\sigma = mgu(L_1, .., L_k)$$

Example 19.8

Let suppose we have a clause $P(x) \vee P(y)$. This clause is not economical.

We can derive $P(x)$ using factoring as follows

$$\textsc{Factor}\frac{P(x) \vee P(y)}{P(x)}\sigma = mgu(P(x), P(y)) = \{y \mapsto x\}$$

Example: why FACTOR rule?

Commentary: Here is the application of resolution for step 3.
$\frac{P(x) \vee P(y) \quad \neg P(x) \vee \neg P(y)}{P(x) \vee \neg P(y)} \sigma = \{x \mapsto x, y \mapsto y, y \mapsto x\}$
Please note that the above σ is not an output of Robinson algorithm, since y and y do not occur in the unifying atoms. How do we understand this?

Example 19.9

1.	$P(x) \vee P(y)$	*Assumption*
2.	$\neg P(x) \vee \neg P(y)$	*Assumption*
3.	$P(x) \vee \neg P(y)$	RESOLUTION *applied to 1 and 2*
4.	$P(x)$	*Applied* FACTOR *applied to 1*
5.	$\neg P(x)$	FACTOR *applied to 2*
6.	$\bot$	RESOLUTION *applied to 4 and 5*

No progress without FACTOR

In the above, we have written the consequences as a sequence, which is equivalent to the DAGs.

Exercise 19.1

Why do we not need similar rule for propositional logic?

No variables

Resolution theorem proving : apply equality over clauses

The rule does not differentiate between $s = t$ and $t = s$

$$\textsc{Paramodulation}\ \frac{s = t \vee C \quad D(u)}{(C \vee D(t))\sigma}\sigma = mgu(s, u)$$

Example 19.10

Consider clauses $f(x) = d \vee \underbrace{P(x)}_{C}$ *and* $\underbrace{Q(f(y))}_{D}$

$$\frac{f(x) = d \vee P(x) \qquad Q(f(y))}{P(y) \vee Q(d)}\sigma = mgu(f(x), f(y)) = \{x \mapsto y\}$$

Commentary: From some σ we have the following implications $(s = t \vee C) \Rightarrow (s\sigma = t\sigma \vee C\sigma)$, $D \Rightarrow D\sigma$, $(s\sigma = t\sigma \vee C\sigma) \wedge (D\sigma) \Rightarrow (s\sigma = t\sigma \wedge (D\sigma)) \vee C\sigma$, $(s\sigma = t\sigma \wedge (D\sigma)) \Rightarrow D\sigma\{s\sigma \mapsto t\sigma\}$. Therefore, PARAMODULATION is valid.

Resolution theorem proving : finishing disequality

If we have a disequality, we can eliminate it if both sides can be unified.

$$\textsc{Relexivity}\frac{t \neq u \vee C}{C\sigma}\sigma = mgu(t, u)$$

Example 19.11

The following derivation removes a literal from the clause.

$$\textsc{Relexivity}\frac{x \neq f(y) \vee P(x)}{P(f(y))}\sigma = mgu(x, f(y)) = \{x \mapsto f(y)\}$$

Example: a resolution proof

Example 19.12

Consider the following set of input clauses

1. $\neg Mother(x, y) \vee husbandOf(y) = fatherOf(x)$
2. $Mother(geoff, maggie)$
3. $bob = husbandOf(maggie)$
4. $fatherOf(geoff) \neq bob$ — *Assumption*
5. $husbandOf(maggie) = fatherOf(geoff)$ — *Resolution applied to 1 and 2*
6. $bob = fatherOf(geoff)$ — *Paramodulation applied to 3 and 5*
7. $\bot$ — *Resolution applied to 4 and 6*

Commentary: Example source http://www.cs.miami.edu/home/geoff/Courses/TPTPSYS/FirstOrder/Paramodulation.shtml

A resolution theorem prover: 5 rules to rule them all

$$\textsc{Assumption}\ \frac{}{C}\ C \in F$$

$$\textsc{Resolution}\ \frac{\neg A \vee C \quad B \vee D}{(C \vee D)\sigma}\ \sigma = mgu(A, B) \qquad \textsc{Factor}\ \frac{L_1 \vee .. \vee L_k \vee C}{(L_1 \vee C)\sigma}\ \sigma = mgu(L_1, .., L_k)$$

$$\textsc{Paramodulation}\ \frac{s = t \vee C \quad D(u)}{(C \vee D(t))\sigma}\ \sigma = mgu(s, u) \qquad \textsc{Relexivity}\ \frac{t \neq u \vee C}{C\sigma}\ \sigma = mgu(t, u)$$

CS433 : automated reasoning

- How to make sat solvers efficient?
- FOL + arithmetic + decision procedures
- Applications to program verification

Topic 19.4

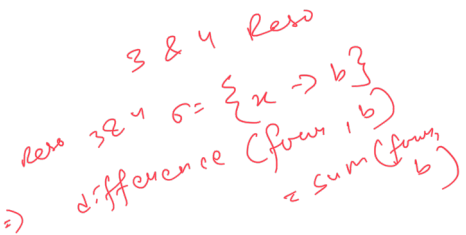
Problems

Exercise: prove the unsatisfiability

Exercise 19.2

Prove that the following set of clauses is unsatisfiable.

1. $Even(sum(twoSquared, b))$
2. $twoSquared = four$
3. $\neg Zero(x) \vee difference(four, x) = sum(four, x)$
4. $Zero(b)$
5. $\neg Even(difference(twoSquared, b))$



Exercise: prove the unsatisfiability

Exercise 19.3

Prove that the following set of clauses is unsatisfiable.

1. $P(f(a))$
2. $a = c$,
3. $\neg Q(x, x) \vee f(c) = f(d)$,
4. $Q(b, b)$
5. $\neg P(f(d))$

Proving transitivity

Exercise 19.4

Prove transitivity of equality using Paramodulation rule.

Exercise: merge factor and resolution

Exercise 19.5

a. Prove the following proof rule using Factor and Resolution rule

$$\textsc{ExtendedResolution}\frac{\neg A_1 \vee ... \vee \neg A_m \vee C \quad B_1 \vee \cdots \vee B_n \vee D}{(C \vee D)\sigma}\sigma = mgu(A_1, \ldots, A_m, B_1, \ldots, B_n)$$

b. Show that the above rule subsumes Factor and Resolution rule.

c. Show that the following rule also subsumes the Factor and Resolution rule.

$$\textsc{RestrictedExtendedResolution}\frac{\neg A \vee C \quad B_1 \vee \cdots \vee B_n \vee D}{(C \vee D)\sigma}\sigma = mgu(A, B_1, \ldots, B_n)$$

Topic 19.5

Extra slides: optimizations in resolution proof systems

Example: Redundancies due to equality reasoning

Example 19.13

Consider the following clauses

1. $a = c$ — *Assumption*
2. $b = d$ — *Assumption*
3. $P(a, b)$ — *Assumption*
4. $\neg P(c, d)$ — *Assumption*
5. $P(c, b)$ — PARAMODULATION *applied to 1 and 3*
6. $P(a, d)$ — PARAMODULATION *applied to 2 and 3*
7. $P(c, d)$ — PARAMODULATION *applied to 2 and 5*
8. $\bot$ — RESOLUTION *applied to and 7*

Redundant derivation

- Many clauses can be derived due to simple permutations
- Often derived clauses do not add new information
- A typical solver restricts application of the rules by imposing order

End of Lecture 19