CS228 Logic for Computer Science 2020

Lecture 20: FOL Resolution

Instructor: Ashutosh Gupta

IITB, India

Compile date: 2020-03-28

Topic 20.1

Refutation proof systems

Recall: Clauses as sets and CNF formulas as set of sets

Definition 20.1 (clause redefined)

A clause is a finite set of literals $\{\ell_1, \dots, \ell_n\}$ *and interpreted as* $\ell_1 \vee .. \vee \ell_n$.

For a clause C and a literal ℓ, we will write $\ell \cup C$ to denote $\{\ell\} \cup C$.

Definition 20.2 (CNF formula redefined)

A CNF formula is a finite set of clauses $\{C_1, \dots, C_n\}$ *and interpreted as* $C_1 \wedge .. \wedge C_n$.

Recall: derivations starting from CNF

We assumed that we have a set of formulas in the lhs, which was treated as 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 $\bigwedge \Sigma \land \neg F$ 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 20.2

Unification and resolution

Applying resolution in FOL

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

$$\textsc{Res}\frac{F \lor C \quad \neg F \lor D}{C \lor D}$$

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

We can make two terms equal by substitutions.

Example 20.1

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

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 20.2

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

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

Issue 2: deriving from substitution?

We know that the following derivation is valid

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

Therefore the following derivations in our clauses are sound

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

Example 20.3

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 20.4

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

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

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 20.3

Resolution theorem proving

Resolution theorem proving

Input: a set of FOL clauses F

Inference rules:

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

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

Example: resolution proof

Example 20.5

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.

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

Therefore, $\neg(\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ *is unsat.*

Therefore, $\emptyset \vdash (\exists x.\forall y R(x,y) \Rightarrow \forall y.\exists x R(x,y))$ *holds.*

Example: resolution with unification

Example 20.6

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 mgu* $\{x \mapsto x, y \mapsto x\}$.

Therefore,

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

Why mgu? – Not just any unifier

Keeps maximum amount of generality in the consequence

Example 20.7

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

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

The above conclusion can always be derived from the mgu consequence

$$\cfrac{\cfrac{P(x,y) \lor Q(y) \qquad \neg P(x,x) \lor R(f(x))}{Q(x) \lor R(f(x))}\sigma = \{x \mapsto x, y \mapsto x\}}{Q(d) \lor R(f(d))}\sigma = \{x \mapsto d\}$$

Once a clause becomes specific, we can not go back.

Resolution theorem proving : factoring

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

We need a rule allows us to simplify clauses.

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

Example 20.8

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

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

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

Example: why FACTOR rule?

Example 20.9

1. $P(x) \vee P(y)$ // *Intro*
2. $\neg P(x) \vee \neg P(y)$ // *Intro*
3. $P(x) \vee \neg P(y)$ //*Applied* RESOLUTION *on 1 and 2*
4. $P(x)$ //*Applied* FACTOR *on 1*
5. $\neg P(x)$ //FACTOR *on 2*
6. $\perp$ //RES *on 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 20.1

a. Do we really need factoring?

b. Why do we not need similar rule for propositional logic?

Resolution theorem proving : apply equality over clauses

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

Example 20.10

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.

Example: Redundancies due to equality reasoning

Example 20.11

Consider the following clauses

1. $a = c$
2. $b = d$
3. $P(a, b)$
4. $\neg P(c, d)$ *//input*
5. $P(c, b)$ *//*PARAMODULATION *on 1 and 3*
6. $P(a, d)$ Redundant derivation *//*PARAMODULATION *on 2 and 3*
7. $P(c, d)$ *//*PARAMODULATION *on 2 and 5*
8. $\perp$ *//*RES *on 4 and 7*

- 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

Attendance quiz

Which of the following are correct applications of paramodulation?

$P(x) \vee x = c, Q(z)/P(c) \vee Q(c)$

$P(x) \vee x = c, Q(z)/P(x) \vee Q(c)$

$P(x) \vee x = c, Q(z)/P(x) \vee Q(x)$

$P(x) \vee x = c, Q(z)/P(c) \vee Q(x)$

$P(x) \ \ x = c, Q(z)/P(c) \ \ Q(c)$

$P(x) \ \ x = c, Q(z)/P(x) \ \ Q(c)$ $P(x) \ \ x = c, Q(z)/P(x) \ \ Q(x)$ $P(x) \ \ x = c,$ $Q(z)/P(c) \ \ Q(x)$

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 20.12

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 20.13

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. $fatherOf(geoff) \neq bob$ *Input*
5. $husbandOf(maggie) = fatherOf(geoff)$ *Resolution applied to 1 and 2*
6. $bob = fatherOf(geoff)$ *Paramodulation applied to 3 and 5*
7. $\perp$ *Resolution applied to 4 and 6*

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

A resolution theorem prover: 5 rules

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

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

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

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

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

Next semester

CS433 : automated reasoning

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

End of Lecture 20