

# FOL Resolution

HE Mingxin, Max

CS104: program07 @ yeah.net

CS108: mxhe1 @ yeah.net

I2ML(H) Spring 2023 (CS104|CS108)

## Exercises 13 : Reading and More

Record your time spent (in 0.1 hours) with brief tasks and durations in your learning log by hand writing!

- 1) Read [textl-ch03-3.2-NormalForms.pdf](#) (cont.)
- 2) Read [textl-ch03-3.4-Unification4FOL.pdf](#) (in 2 weeks)

## Topic 13B.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  $\Sigma$  as a set of clauses**.

## Recall: Refutation Proof System

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

We assume  $\Sigma$  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  $\Sigma'$ .

We apply the a **refutation proof method** on  $\Sigma'$ .

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

## Topic 13B.2

### Unification and Resolution

## Applying resolution in FOL

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

$$\text{RESOLUTION} \frac{F \vee C \quad \neg F \vee D}{C \vee D}$$

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

We can make two terms equal by substitutions.

### Example 13B.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 13B.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 if  $\Sigma$  is a set of sentences.

1.  $\Sigma \vdash \forall x, y. F(x, y)$
2.  $\Sigma \vdash F(t_1(x, y), t_2(x, y))$
3.  $\Sigma \vdash \forall x, y. F(t_1(x, y), t_2(x, y))$

Premise

$\forall$ -Elim

$\forall$ -Intro

Therefore the following derivations in our clauses are sound

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

### Example 13B.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 13B.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 13B.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:**

$$\text{ASSUMPTION} \frac{}{C} C \in F$$

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

## Example: resolution proof

### Example 13B.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.

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

Therefore,  $\neg(\exists x. \forall y R(x, y) \Rightarrow \forall y. \exists x R(x, y))$  is unsat, 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 13B.6

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

$x$ s *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,

$$\text{RESOLUTION} \frac{P(x, y) \vee Q(y) \quad \neg P(x, x) \vee R(f(x))}{Q(x) \vee R(f(x))} \sigma = \{x \mapsto x, y \mapsto x\}$$

## Why MGU? – not just any unifier

**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 maximum opportunity of finding the empty clause.

MGU keeps maximum generality in the consequence

### Example 13B.7

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

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

The above conclusion can always be derived from the mgu consequence

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

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.

$$\text{FACTOR} \frac{L_1 \vee \dots \vee L_k \vee C}{(L_1 \vee C)\sigma} \sigma = \text{mgu}(L_1, \dots, L_k)$$

### Example 13B.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

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

## 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  $\sigma$  is not an output of Robinson algorithm, since  $y$  and  $y$  do not occur in the unifying atoms. How do we understand this?

### Example 13B.9

1.  $P(x) \vee P(y)$
2.  $\neg P(x) \vee \neg P(y)$
3.  $P(x) \vee \neg P(y)$
4.  $P(x)$
5.  $\neg P(x)$
6.  $\perp$

No progress without  
FACTOR

*Assumption*

*Assumption*

RESOLUTION *applied to 1 and 2*

*Applied* FACTOR *applied to 1*

FACTOR *applied to 2*

RESOLUTION *applied to 4 and 5*

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

### Thinking Exercise 13B.1

*Why do we not need similar rule for propositional logic?*

## Resolution theorem proving : apply equality over clauses

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

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

### Example 13B.10

Consider clauses  $f(x) = d \vee \underbrace{P(x)}_C$  and  $\underbrace{Q(f(y))}_D$

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

**Commentary:** From some  $\sigma$  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**.

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

### Example 13B.11

*The following derivation removes a literal from the clause.*

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

## Example: a resolution proof

### Example 13B.12

Consider the following set of input clauses

1.  $\neg \text{Mother}(x, y) \vee \text{husbandOf}(y) = \text{fatherOf}(x)$
2.  $\text{Mother}(\text{geoff}, \text{maggie})$
3.  $\text{bob} = \text{husbandOf}(\text{maggie})$
4.  $\text{fatherOf}(\text{geoff}) \neq \text{bob}$
5.  $\text{husbandOf}(\text{maggie}) = \text{fatherOf}(\text{geoff})$
6.  $\text{bob} = \text{fatherOf}(\text{geoff})$
7.  $\perp$

*Assumption*

*Resolution applied to 1 and 2*

*Paramodulation applied to 3 and 5*

*Resolution applied to 4 and 6*

## A resolution theorem prover: 5 rules to rule them all

$$\text{ASSUMPTION} \frac{}{C} C \in F$$

$$\text{RESOLUTION} \frac{\neg A \vee C \quad B \vee D}{(C \vee D)\sigma} \sigma = \text{mgu}(A, B) \quad \text{FACTOR} \frac{L_1 \vee \dots \vee L_k \vee C}{(L_1 \vee C)\sigma} \sigma = \text{mgu}(L_1, \dots, L_k)$$

$$\text{PARAMODULATION} \frac{s = t \vee C \quad D(u)}{(C \vee D(t))\sigma} \sigma = \text{mgu}(s, u) \quad \text{RELEXIVITY} \frac{t \neq u \vee C}{C\sigma} \sigma = \text{mgu}(t, u)$$

Topic 13B.4

Problems

## Exercise: prove the unsatisfiability

### Thinking Exercise 13B.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

### Thinking Exercise 13B.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

### Thinking Exercise 13B.4

*Prove transitivity of equality using Paramodulation rule.*

## Exercise: merge factor and resolution

### Thinking Exercise 13B.5

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

$$\text{EXTENDEDRESOLUTION} \frac{\neg A_1 \vee \dots \vee \neg A_m \vee C \quad B_1 \vee \dots \vee B_n \vee D}{(C \vee D)\sigma} \sigma = \text{mgu}(A_1, \dots, A_m, B_1, \dots, B_n)$$

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

c. *Prove/disprove that we can replace both Resolution rule and Factor rule in the FOL resolution proof system by the following rule without losing completeness of deriving false clause.*

$$\text{RESTRICTEDEXTENDEDRESOLUTION} \frac{\neg A \vee C \quad B_1 \vee \dots \vee B_n \vee D}{(C \vee D)\sigma} \sigma = \text{mgu}(A, B_1, \dots, B_n)$$

# General Resolution

## Thinking Exercise 13B.6

*Consider the following general resolution rule which is more liberal in application in comparison to the standard FOL resolution. Show that if the following proof rule can prove  $\perp$  then the standard resolution can also do it.*

$$\text{GENERALRESOLUTION} \frac{\neg A \vee C \quad B \vee D}{(C\sigma_1 \vee D\sigma_2)} A\sigma_1 = B\sigma_2$$

## Topic 13B.5

Extra slides: optimizations in resolution proof systems

## Example: Redundancies due to equality reasoning

### Example 13B.13

Consider the following clauses

1.  $a = c$
2.  $b = d$
3.  $P(a, b)$
4.  $\neg P(c, d)$
5.  $P(c, b)$
6.  $P(a, d)$
7.  $P(c, d)$
8.  $\perp$

Redundant  
derivation

*Assumption*

PARAMODULATION applied to 1 and 3

PARAMODULATION applied to 2 and 3

PARAMODULATION applied to 2 and 5

RESOLUTION applied to 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

End of Lecture 13B