CS 461
Artificial Intelligence

Dr. Hashim Yasin

Resolution in First Order Logic

Resolution

- Here's the rule for first-order resolution.

$$\frac{\begin{array}{c}\alpha \vee \varphi \\ \neg \psi \vee \beta\end{array}}{(\alpha \vee \beta)\theta} \qquad MGU(\varphi, \psi) = \theta$$

- **Example 1:**

$$\frac{\begin{array}{l}P(x) \vee Q(x,y) \\ \neg P(A) \vee R(B,z)\end{array}}{(Q(x,y) \vee R(B,z))\theta} \qquad \theta = \{x/A\}$$

$$Q(A,y) \vee R(B,z)$$

Resolution

Example 2:

- In this example, there is ***x*** **in the two sentences** which are actually *different*.

$$P(x) \vee Q(x,y)$$

$$\neg P(A) \vee R(B,x)$$

- Furthermore, ***there is an implicit universal quantifier*** on the outside of each of these sentences.

$$\forall xy.\quad P(x) \vee Q(x,y)$$

$$\forall x.\quad \neg P(A) \vee R(B,x)$$

Resolution

Example 2:

$$\forall xy. \quad P(x) \vee Q(x,y)$$

$$\forall x. \quad \neg P(A) \vee R(B,x)$$

Scope of var is local to a clause.
Use renaming to keep vars distinct

- **Rename the variables (Standardizing apart)** in the two sentences so that they don't share any variables in common. **Standardizing apart** eliminates overlap of variables.

$$\forall x_1 y. \quad P(x_1) \vee Q(x_1,y)$$

$$\forall x_2. \quad \neg P(A) \vee R(B,x_2)$$

Resolution

$$\forall x_1 y. \quad P(x_1) \lor Q(x_1,y)$$
$$\forall x_2. \quad \neg P(A) \lor R(B,x_2)$$

$$\begin{array}{ll} \forall x_1 y. & \boxed{P(x_1)} \lor Q(x_1,y) \\ \forall x_2. & \boxed{\neg P(A)} \lor R(B,x_2) \\ \hline & (Q(x_1,y) \lor R(B,x_2))\theta \end{array}$$

$$\theta = \{x_1/A\}$$

$$\begin{array}{ll} \forall x_1 y. & P(x_1) \lor Q(x_1,y) \\ \forall x_2. & \neg P(A) \lor R(B,x_2) \\ \hline & (Q(x_1,y) \lor R(B,x_2))\theta \\ & Q(A,y) \lor R(B,x_2) \end{array}$$

Factoring

Factoring

- **<u>Factoring</u>**—the removal of redundant literals—to the first-order case.
- Propositional factoring reduces two literals to one if they are *identical*;
- First-order factoring reduces two literals to one if they are *unifiable*.
- The unifier must be applied to the entire clause.
- *The combination of binary resolution and factoring is complete*

Binary Resolution

- The version of the first-order resolution rule that we have worked with, is called ***binary resolution***.
- Binary Resolution involves **two literals**, one from each clause being resolved.
- This form of resolution is **not complete** for first-order logic.
  - There are sets of unsatisfiable clauses that will not generate a contradiction by successive applications of binary resolution.

Binary Resolution

The resolution rule alone is not complete.

For example, the set $\{\overbrace{P(x) \vee P(y)}^{\alpha}, \overbrace{\neg P(x) \vee \neg P(y)}^{\beta}\}$ is unsatisfiable.

But all we can do is start with

$$\frac{\overline{P(x)} \vee P(y) \qquad \neg \overline{P(x)} \vee \neg P(y)}{\underbrace{P(y) \vee \neg P(y)}_{\gamma}}$$

But then, resolving γ with α, or γ with β leads nowhere.

Binary Resolution

- **Can we get a contradiction using binary resolution?**

$$P(x) \vee P(y)$$
$$\neg P(v) \vee \neg P(w)$$

- We can get,

$$\frac{\begin{array}{c} P(x) \vee P(y) \\ \neg P(v) \vee \neg P(w) \end{array}}{P(x) \vee \neg P(w)}$$

Binary Resolution

$$P(x) \vee \neg P(w)$$

- *If we use binary resolution on this new clause with one of the parent clauses, we get back one of the parent clauses.* We do not get a contradiction.
- So, we have shown by counterexample that **Binary Resolution** is **NOT**, in fact, a complete strategy.

- There is a simple extension of binary resolution that is complete, called **factoring**.

Factoring

- In factoring,
  - **unify two literals** within a **single** clause,
    - **alpha** and **beta** in this case, with **unifier theta**,
  - **drop one of them** from the clause (it doesn't matter which one),
  - apply the unifier to the whole clause.

$$\frac{\alpha \vee \beta \vee \gamma}{(\alpha \vee \gamma)\theta} \qquad \theta = \mathrm{MGU}(\alpha, \beta)$$

- Binary Resolution, combined with factoring, is **complete**.

Factoring

Example:

$$\underline{Q(y) \vee P(x,y) \vee P(v,A)}$$

- we can apply factoring to this sentence, by unifying $\boldsymbol{P(x,y)}$ and $\boldsymbol{P(v,A)}$.

$$\frac{Q(y) \vee \boxed{P(x,y) \vee P(v,A)}}{(Q(y) \vee P(x,y))\{x/v, y/A\}}$$

$$Q(A) \vee P(v,A)$$

Factoring

Example:

$$\frac{Q(y) \vee \boxed{P(x,y) \vee P(v,A)}}{(Q(y) \vee P(x,y))\{x/v, y/A\}}$$

$$Q(A) \vee P(v,A)$$

- First, we get $Q(y) \ \vee \ P(x,y)$, and we have to apply the **substitution** $\{x/v, y/A\}$, which yields the result

$$Q(A) \ \vee \ P(v,A)$$

Factoring... Example

- Given

$$\neg p(x) \lor \neg q(y)$$
$$\neg p(x) \lor q(y)$$
$$p(x) \lor \neg q(y)$$
$$p(x) \lor q(y)$$

1	$\neg p(x) \lor \neg q(y)$	Given
2	$\neg p(x) \lor q(y)$	Given
3	$p(x) \lor \neg q(y)$	Given
4	$p(x) \lor q(y)$	Given

Factoring… Example

1	$\neg p(x) \lor \neg q(y)$	Given
2	$\neg p(x) \lor q(y)$	Given
3	$p(x) \lor \neg q(y)$	Given
4	$p(x) \lor q(y)$	Given
5	$\neg p(x) \lor \neg p(x)$	Resolution (1,2)
6	$\neg p(x)$	Factoring (5)
7	$p(x) \lor p(x)$	Resolution (3,4)
8	$p(x)$	Factoring (7)
9	.	Resolution (6,8)

Resolution in First Order Logic

Converting to Clausal Form

- Eliminate $\rightarrow, \leftrightarrow$

$$\alpha \rightarrow \beta \qquad\qquad \neg\alpha \lor \beta$$

- Drive in $\neg$

$$\neg(\alpha \lor \beta) \qquad\qquad \neg\alpha \land \neg\beta$$

$$\neg(\alpha \land \beta) \qquad\qquad \neg\alpha \lor \neg\beta$$

$$\neg\neg\alpha \qquad\qquad \alpha$$

$$\neg\exists x.\ P(x) \qquad\qquad \forall x. \neg P(x)$$

$$\neg\forall x. P(x) \qquad\qquad \exists x. \neg P(x)$$

Converting to Clausal Form

- Rename variables apart

$$\forall x. \exists y. (P(x) \rightarrow \forall x. Q(x, y))$$

$$\forall x_1. \exists y_2. (P(x_1) \rightarrow \forall x_3. Q(x_3, y_2))$$

- Skolemization
  - <u>***Skolem Constant:***</u> Substitute brand new name for each *existentially quantified variable*
  - $\exists\, x. P(x)$ $\qquad$ $P(Fred)$
  - $\exists\, x. P(x, y)$ $\qquad$ $P(X11, Y13)$
  - $\exists\, x. P(x) \wedge Q(x)$ $\qquad$ $P(Blue) \wedge Q(Blue)$

Converting to Clausal Form

- Skolemization
  - ***<u>Skolem Function:</u>*** Substitute a **new function** of all universally quantified variables in enclosing scopes for each *existentially quantified variable*.
  - **"There is someone who is loved by everyone"**

$$\exists\, y.\forall\, x. Loves(x, y)$$

$$\forall\, x. Loves(x, Englebert)$$

  - **"Everybody loves somebody"**

$$\forall\, x.\exists\, y. Loves(x, y)$$

$$\forall\, x. Loves(x, Beloved(x))$$

Converting to Clausal Form

- Prenex Form
- Drop universal quantifiers
- Convert to CNF
- Rename the variables in each clause, if necessary

Example: Converting to Clausal Form

1. Jack owns a dog.
2. Every dog owner is an animal lover.
3. No animal lover kills an animal.
4. Either Jack or Curiosity killed the cat, who is named Tuna.
5. Did Curiosity kill the cat?

Example: Converting to Clausal Form

1. **Jack owns a dog.**

$$\exists x. D(x) \wedge O(J,x), \quad \forall \boldsymbol{x}. \boldsymbol{D}(\boldsymbol{x}) \rightarrow \mathbf{A}(\boldsymbol{x})$$

$$D(Fido) \wedge O(J, Fido)$$

2. **Every dog owner is an animal lover.**

$$\forall x. \left[\underbrace{\left(\exists y. D(y) \wedge O(x,y) \right)}_{} \rightarrow L(x) \right]$$

Example: Converting to Clausal Form

$\neg \exists x.\ P(x) \ \equiv\ \forall x. \neg\, P(x)$

2. **Every dog owner is an animal lover.**

$$\forall x. [(\exists y. D(y) \land\ O(x,y)) \rightarrow L(x)]$$

$$\forall x. [\neg \exists y. (D(y)\ \land\ O(x,y))\ \lor\ L(x)]$$

$$\forall x. \forall y. \neg(D(y)\ \land\ O(x,y))\ \lor\ L(x)$$

$$\forall x. \forall y. \neg\, D(y) \lor\ \neg\, O(x,y) \lor\ L(x)$$

Quantifiers can be dropped since all variables can be assumed to be universally quantified by default.

$$\neg\, D(y)\ \lor\ \neg\, O(x,y)\ \lor\ L(x)$$

Example: Converting to Clausal Form

3. **No animal lover kills an animal.**

$$\forall x.\ L(x) \rightarrow (\forall y. A(y) \rightarrow \neg K(x,y))$$

$$\forall x.\ L(x) \rightarrow (\forall y. A(y) \rightarrow \neg K(x,y))$$

$$\forall x. \neg L(x) \ \vee \ (\forall y. A(y) \ \rightarrow \neg K(x,y))$$

$$\forall x. \neg L(x) \ \vee \ (\forall y. \neg A(y) \ \vee \ \neg K(x,y))$$

$$\forall x, y. \neg L(x) \ \vee \ \neg A(y) \ \vee \ \neg K(x,y)$$

$$\neg L(x) \ \vee \ \neg A(y) \ \vee \ \neg K(x,y)$$

Example: Converting to Clausal Form

4. **Either Jack or Curiosity killed the cat, who is named Tuna**

$$K(J,T) \lor K(C,T)$$

$$C(T)$$

$$\forall x. C(x) \rightarrow \mathrm{A}(\mathrm{x})$$
$$\forall x. \neg C(x) \lor \mathrm{A}(\mathrm{x})$$

5. **Did Curiosity kill the cat?**

$$K(C,T)$$

Resolution

1	D(Fido)	a
2	O(J,Fido)	a
3	¬ D(y) v ¬ O(x,y) v L(x)	b
4	¬ L(x) v ¬ A(y) v ¬ K(x,y)	c
5	K(J,T) v K(C,T)	d
6	C(T)	e
7	¬ C(x) v A(x)	f
8	¬ K(C,T)	Neg

Resolution

1	D(Fido)	a
2	O(J,Fido)	a
3	¬ D(y) v ¬ O(x,y) v L(x)	b
4	¬ L(x) v ¬ A(y) v ¬ K(x,y)	c
5	K(J,T) v K(C,T)	d
6	C(T)	e
7	¬ C(x) v A(x)	f
8	¬ K(C,T)	Neg
9	K(J,T)	5,8
10	A(T)	6,7 {x/T}
11	¬ L(J) v ¬ A(T)	4,9 {x/J, y/T}
12	¬ L(J)	10,11
13	¬ D(y) v ¬ O(J,y)	3,12 {x/J}
14	¬ D(Fido)	13,2 {y/Fido}
15	•	14,1

16	¬L(x) ∨ ¬K(x, T)	4,10{y/T}
17	¬D(y) ∨ ¬O(x, y) ∨ ¬K(x, T)	3,16
18	¬O(x, Fido) ∨ ¬K(x, T)	1,17 {y/Fido}
19	¬K(J, T)	2,18 {x/J}
20	K(C, T)	5,19
21	.	8, 20

Green Trick

Green's Trick

- We **can ask for an answer** to a question with resolution.
- If the desired conclusion is that there exists an x such that $P(x)$, we'll figure out what value of x makes $P(x)$ true.
- **Green's trick,** named after **Cordell Green**, who pioneered the use of logic.
- “There exists a sequence of actions such that, if I do them in my initial state, my goal will be true at the end.”

Green's Trick

Example:

All men are mortal and Socrates is a man.

- We want to know whether there are *any mortal available in the knowledge base*.
- Use resolution to get answers.

- The desired conclusion, negated and turned into clausal form would be

$$\neg Mortal\ (x).$$

- **Green's trick will be to add a special extra literal onto that clause, of the form $Answer(x)$.**

Green's Trick

Example:

All men are mortal and Socrates is a man.

1.	$\neg Man(x) \vee Mortal(x)$	
2.	$Man(Socrates)$	
3.	$\neg Mortal(x) \vee Answer(x)$	
4.		
5.		

Green's Trick

Example:

All men are mortal and that Socrates is a man.

1.	$\neg Man(x) \vee Mortal(x)$	
2.	$Man(Socrates)$	
3.	$\neg Mortal(x) \vee Answer(x)$	
4.	$Mortal(Socrates)$	1,2
5.	$Answer(Socrates)$	3,4

We can resolve lines (3 and 4), substituting $Socrates$ for x, and get $Answer$.

Equality

- In case of first-order logic, there is a special equality predicate.
- Treat equality almost like any other predicate, but to constrain its semantics via axioms.
- Equals has ***three important sets of properties***.
  - **First,** it's **reflexive**: every x is equal to itself.
  - **Second,** it's **symmetric**: If x is equal to y then y is equal to x.
  - **Third,** it's **transitive**. That means that if x equals y and y equals z, then x equals z.

Equality

- **First,** it's **reflexive**: every x is equal to itself.

$$\forall x.\mathrm{Eq}(x, x)$$

- **Second,** it's **symmetric**. If x is equal to y then y is equal to x.

$$\forall x, y.\mathrm{Eq}(x, y) \rightarrow \mathrm{Eq}(y, x)$$

- **Third,** it's **transitive**. That means that if x equals y and y equals z, then x equals z.

$$\forall x, y, z.\mathrm{Eq}(x, y) \wedge \mathrm{Eq}(y, z) \rightarrow \mathrm{Eq}(x, z)$$

Equality

- The other thing we need is the ability to **"substitute equals for equals"** into any place in any predicate.

- That means that, for each place in each predicate, we'll need an axiom that looks like this:

for all x and y, if x equals y, then if P holds of x, it holds of y.

$$\forall x, y. Eq(x, y) \rightarrow (P(x) \leftrightarrow P(y))$$

Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig

  - **Chapter 8 & 9.**