COMP340-08B Reasoning About Programs

19. Resolution in Predicate Logic

Robi Malik

THE UNIVERSITY OF WAIKATO *Te Whare Wānanga o Waikato*

DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

Resolution and Predicate Logic

So far …

- Resolution of clauses
- Translation of propositional connectives into clausal form

But …

- What about predicate logic?
- What about quantifiers?

Using the Skolem Normal Form

Remember …

- Predicate logic formulas can be transformed into Skolem Normal Form without existential quantifiers that is **satisfiability equivalent**.

Thus …

- A formula A is **unsatisfiable** if and only if its Skolem Normal Form is unsatisfiable.

Example

Prove from the premises …

1. $\forall x\,(\mathrm{man}(x) \rightarrow \mathrm{mortal}(x))$
2. man(socrates)
3. man(platon)
4. man(euclid)

…

the conclusion …

mortal(socrates)

Step 1 — Translation into Clauses

Universal quantifier can be dropped

1.	$\forall x\ \{\neg\mathrm{man}(x), \mathrm{mortal}(x)\}$	Premise 1
2.	$\{\mathrm{man}(\mathrm{socrates})\}$	Premise 2
3.	$\{\mathrm{man}(\mathrm{platon})\}$	Premise 3
4.	$\{\mathrm{man}(\mathrm{euclid})\}$	Premise 4
5.	$\{\neg\mathrm{mortal}(\mathrm{socrates})\}$	Negated conclusion

Solving the Example

1.	$\{\neg\mathrm{man}(\mathrm{socrates}), \mathrm{mortal}(\mathrm{socrates})\}$	Instance of Premise 1
2.	$\{\neg\mathrm{man}(\mathrm{platon}), \mathrm{mortal}(\mathrm{platon})\}$	Instance of Premise 1
3.	$\{\neg\mathrm{man}(\mathrm{euclid}), \mathrm{mortal}(\mathrm{euclid})\}$	Instance of Premise 1
4.	$\{\mathrm{man}(\mathrm{socrates})\}$	Premise 2
5.	$\{\mathrm{man}(\mathrm{platon})\}$	Premise 3
6.	$\{\mathrm{man}(\mathrm{euclid})\}$	Premise 4
7.	$\{\neg\mathrm{mortal}(\mathrm{socrates})\}$	Negated Conclusion
8.	$\{\mathrm{mortal}(\mathrm{socrates})\}$	Res: 4,1
9.	$\square$	Res: 8,7

Using the Herbrand Base

Idea

Instantiate all clauses in all possible ways using **ground terms**, and apply resolution to the resultant set of ground clauses.

ground term = term without variables

Problems?

Another Way of Solving the Example

1.	$\{\neg \text{man}(x), \text{mortal}(x)\}$	Premise 1
2.	$\{\text{man(socrates)}\}$	Premise 2
3.	$\{\text{man(platon)}\}$	Premise 3
4.	$\{\text{man(euclid)}\}$	Premise 4
5.	$\{\neg\text{mortal(socrates)}\}$	Negated Conclusion
6.	$\{\text{mortal(socrates)}\}$	Res: 2,1 $\theta = [x/\text{socrates}]$
7.	$\square$	Res: 6,5

Apply Substitution

Substitutions

Definition

A **substitution** is a list of pairs of variables and terms.

$$\theta = [x_1/t_1, x_2/t_2, \ldots, x_n/t_n]$$

Applying Substitutions

Definition

Let

$$\theta = [x_1/t_1, x_2/t_2, \ldots, x_n/t_n]$$

be a substitution, and let t be a term.

The result of applying θ to t, written $t\theta$, is obtained **by simultaneously replacing** each occurrence of x_i in t by t_i.

The result of applying θ to a formula A, written $A\theta$, is defined analogously.

Examples

- $A = \text{man}(x);\ \ \theta = [x/\text{socrates}]$

  $A\theta =$

- $A = \text{parent}(x,y);\ \ \theta = [x/\text{father}(y), y/\text{mary}]$

  $A\theta =$

- $A = x < \text{s}(x);\ \ \theta = [x/\text{s}(\text{s}(0))]$

  $A\theta =$

Concatenating Substitutions

Definition

The **concatenation** of two substitutions

$$\theta = [x_1/t_1, x_2/t_2, \ldots, x_n/t_n]$$

$$\sigma = [y_1/u_1, y_2/u_2, \ldots, y_m/u_m]$$

is defined as

$$\theta\sigma = [x_1/t_1\sigma, \ldots, x_n/t_n\sigma, y_{i_1}/u_{i_1}, \ldots, y_{i_k}/u_{i_k}]$$

where $y_{i_1},\ldots, y_{i_k}$ are those variables out of $\{y_1,\ldots, y_m\}$ that do not occur in $\{x_1,\ldots, x_n\}$.

Resolution Procedure in Predicate Logic

To check whether a set Φ of formulas is unsatisfiable ...

1. Transform all formulas into **Skolem Normal Form**.
2. Drop universal quantifiers and extract clauses.
3. Apply resolution.
4. If the empty clause is obtained, then Φ is unsatisfiable.

Unification Examples

1. Constants unify if they are equal.

   $a = a$ ✓ $a = b$ ✗

2. A variable x unifies with any term that does not contain x (**occurs check**).

   $x = f(x)$ ✗ $g(x,b) = y$ ✓ $[y/g(x,b)]$

3. Composed terms unify if they have the same function symbol and all their components unify.

   $date(x,y,2003) = date(7,mar,2003)$ ✓ $[x/7, y/mar]$

Resolution Rule in Predicate Logic

(Res)
$$\frac{C_1 \cup \{L_1\} \quad C_2 \cup \{\neg L_2\}}{(C_1 \cup C_2)\theta}$$

Provided that θ is a substitution such that $L_1\theta = L_2\theta$.

Unification Algorithm

- Takes as input two terms (or formulas) A and B.
- Determines whether A and B are unifiable.
- If A and B are unifiable, produces a **most general unifier** θ such that

$$A\theta = B\theta.$$

Unification

Definition

Let A and B be two formulas (or terms). A substitution θ is called a **unifier** for A and B if $A\theta = B\theta$.

A and B are called **unifiable**, if there exists a unifier for A and B.

Unification Algorithm

```
mgu := [];  /* empty substitution */
while A ≠ B do
  let t_A and t_B be the first differing subterms of A and B;
  if t_A = x is a variable that does not occur in t_B then
    mgu := mgu[x/t_B]; A := A[x/t_B]; B := B[x/t_B];
  else if t_B = y is a variable not in t_A then
    mgu := mgu[y/t_A]; A := A[y/t_A]; B := B[y/t_A];
  else
    return "not unifiable";
  end if
end while
return mgu;
```

Example

1. $\{\mathrm{mimsy}(x)\}$
2. $\{\neg\mathrm{brillig}(x), \mathrm{gyre}(x, \mathrm{f}(x))\}$
3. $\{\neg\mathrm{brillig}(x), \mathrm{slithy}(\mathrm{f}(x))\}$
4. $\{\neg\mathrm{tove}(x), \mathrm{gyre}(x, \mathrm{f}(x))\}$
5. $\{\neg\mathrm{tove}(x), \mathrm{slithy}(\mathrm{f}(x))\}$
6. $\{\neg\mathrm{mimsy}(x), \mathrm{brillig}(x)\}$
7. $\{\neg\mathrm{gyre}(x, y), \neg\mathrm{brillig}(y)\}$

One Final Point

- Clauses are implicitly combined by “and”, and universally quantified.
- Note that

$$\forall x\, (A(x) \wedge B(x))$$

is logically equivalent to

$$\forall x\, A(x) \wedge \forall y\, B(y)$$

- Therefore, we can **rename variables** in clauses **initially** and **after each step**.

Solving the Example

1.	$\{\mathrm{mimsy}(x_1)\}$	Premise
2.	$\{\neg\mathrm{brillig}(x_2), \mathrm{gyre}(x_2,\mathrm{f}(x_2))\}$	Premise
3.	$\{\neg\mathrm{brillig}(x_3), \mathrm{slithy}(\mathrm{f}(x_3))\}$	Premise
4.	$\{\neg\mathrm{tove}(x_4), \mathrm{gyre}(x, \mathrm{f}(x_4))\}$	Premise
5.	$\{\neg\mathrm{tove}(x_5), \mathrm{slithy}(\mathrm{f}(x_5))\}$	Premise
6.	$\{\neg\mathrm{mimsy}(x_6), \mathrm{brillig}(x_6)\}$	Premise
7.	$\{\neg\mathrm{gyre}(x_7, y_7), \neg\mathrm{brillig}(y_7)\}$	Premise
8.	$\{\mathrm{brillig}(x_8)\}$	Res: 1,6; $\theta = [x_1/x_8, x_6/x_8]$
9.	$\{\mathrm{gyre}(x_9, \mathrm{f}(x_9))\}$	Res: 8,2; $\theta = [x_2/x_9, x_8/x_9]$
10.	$\{\neg\mathrm{brillig}(\mathrm{f}(x_{10}))\}$	Res: 9,7; $\theta = [x_7/x_{10}, x_9/x_{10}, y_7/\mathrm{f}(x_{10})]$
11.	$\{\neg\mathrm{mimsy}(\mathrm{f}(x_{11}))\}$	Res: 6,10; $\theta = [x_6/\mathrm{f}(x_{11}), x_{10}/x_{11}]$
12.	$\square$	Res: 1,11; $\theta = [x_1/\mathrm{f}(x_{11})]$

Another Example

1.	$\{\neg r(x_1,y_1), r(y_1,x_1)\}$	Premise
2.	$\{\neg r(x_2,y_2), \neg r(y_2,z_2), r(x_2,z_2)\}$	Premise
3.	$\{r(b,a)\}$	Premise
4.	$\{r(c,a)\}$	Premise
5.	$\{r(b,d)\}$	Premise
6.	$\{\neg r(c,d)\}$	Premise
7.	$\{r(a,b)\}$	Res: 3,1; $\theta = [x_1/b, y_1/a]$
8.	$\{\neg r(a,z_3), r(c,z_3)\}$	Res: 4,2; $\theta = [x_2/c, y_2/a, z_2/z_3]$
9.	$\{r(c,b)\}$	Res: 7,8; $\theta = [z_3/b]$
10.	$\{\neg r(b,z_4), r(c,z_4)\}$	Res: 9,2; $\theta = [x_2/c, y_2/b, z_2/z_4]$
11.	$\{r(c,d)\}$	Res: 5,10; $\theta = [z_4/d]$
12.	$\square$	Res: 11,6

Summary

To determine whether a set Φ of predicate logic formulas is unsatisfiable using resolution …

1. Rewrite all formulas into Skolem Normal Form.
2. Transform the results into clauses, dropping universal quantifiers and renaming variables.
3. Apply resolution.
   After each step, rename variables.
4. If the empty clause $\square$ is found, then Φ is unsatisfiable.

It's Sound and Complete!

Theorem

Let Ψ be a set of predicate logic clauses.
If the empty clause $\square$ can be proven from Ψ by resolution, then Ψ is **unsatisfiable**.

Theorem

Let Ψ be a set of predicate logic clauses.
If Ψ is **unsatisfiable**, then the empty clause $\square$ can be proven from Ψ by resolution.