Logic and Computation

CS245

Dr. Borzoo Bonakdarpour

University of Waterloo

(Fall 2012)

Resolution in First-order Predicate Logic

Agenda

- Resolution in Propositional Logic
- Prenex Normal Form
- Skolemization
- Ground Resolution in FOL
- Unification
- General Resolution

Literals and Clauses

A *literal* is a propositional variable or the negation of a propositional variable.

Two literals are said to be *complements* (or *conjugate*), if one is the negation of the other (e.g., p and $\neg p$)

A formula of the form $C_i = p_1 \vee p_2 \vee \cdots \vee p_n$, where each p_i is a literal is called a *clause*.

CNF

A formula in the *conjunctive normal form* (CNF) is a conjunction of clauses

For example, these formulas are in CNF:

$$(p \vee q) \wedge (\neg q \vee r \vee \neg m) \wedge (m \vee \neg n)$$

$$p \wedge q$$

It is possible to convert any formula into an equivalent formula in CNF.

CNF

The CNF equivalent of the following formulas:

$$(p \wedge q) \vee r$$

$$\neg(p \vee q)$$

are these:

$$(p \vee r) \wedge (q \vee r)$$

$$\neg p \wedge \neg q$$

Resolution Rule

$$\frac{p_1 \lor \cdots \lor p_i \lor \ldots p_n, \quad q_1 \lor \cdots \lor q_j \lor \ldots q_m}{p_1 \lor \cdots \lor p_{i-1} \lor p_{i+1} \lor \ldots p_n \lor q_1 \cdots \lor q_{j-1} \lor q_{j+1} \lor \cdots \lor q_m}$$

where $p_1 \ldots p_n, q_1 \ldots q_m$ are propositions and p_i and q_j are complements.

The clause produced by the resolution rule is called the *resolvent* of the two input clauses.

The upper side of the rull is in CNF and may have multiple clauses.

Proof by Resolution

The resolution rule can be used to develop a finite-step proof for propositional logic:

1- Transform the CNF formual into a set S of caulses. For example, for formula:

$$(p \vee q \vee r) \wedge (\neg r \vee \neg p \vee m) \wedge q$$

we have:

$$S = \{\{p, q, r\}, \{\neg r, \neg p, m\}, \{q\}\}$$

Proof by Resolution

2- The resolution rule is applied to all possible pairs of clauses that contain complementary literals. After each application of the resolution rule, the resulting sentence is simplified by

- Removing repeated literals.
- If the sentence contains complementary literals, it is removed (as a validity).
- If not, and if it is not yet present in the clause set S, then it is added to S, and is considered for further resolution inferences.

Proof by Resolution

Example:

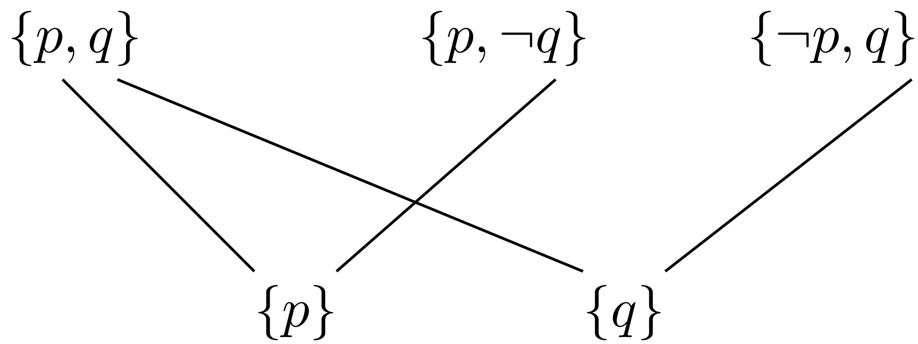
$$\frac{p \vee q, \quad \neg p \vee r}{q \vee r}$$

This is equal to (different syntax):

$$\frac{\{p, q\}, \{\neg p, r\}}{\{q, r\}}$$

Proof by Resolution

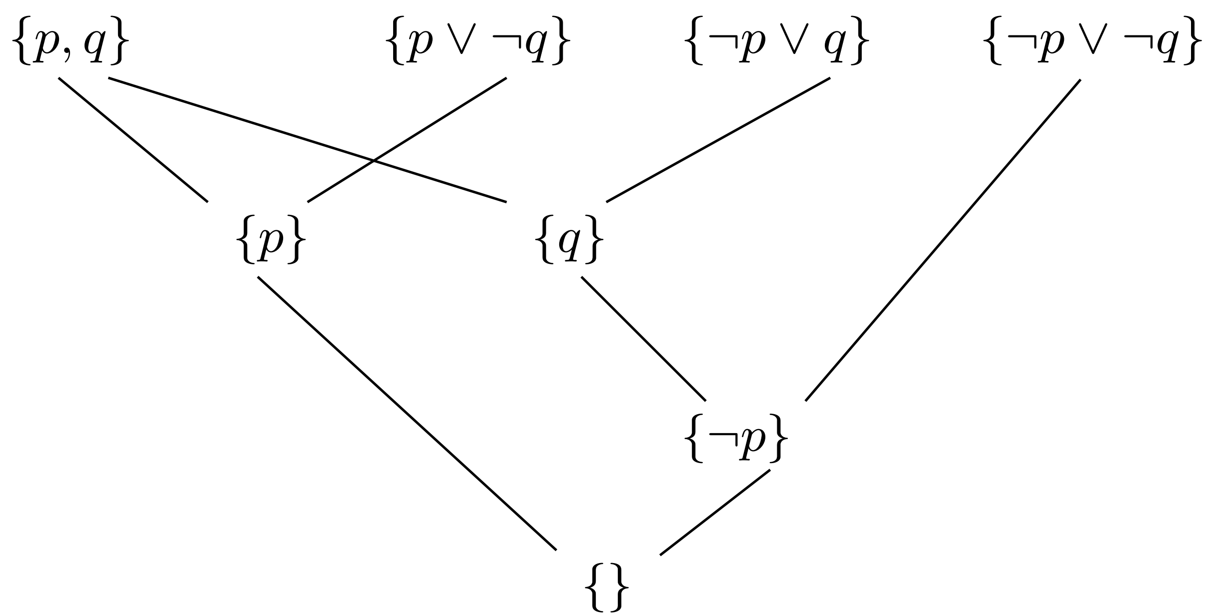
Example (in directed acyclic graph):



3- If the empty clause cannot be derived, and the resolution rule cannot be applied to derive any more new clauses, then the original formula is satisfiable.

Proof by Resolution

4- If after applying a resolution rule the empty clause is derived, the original formula is unsatisfiable (i.e., a contradiction).
Example:

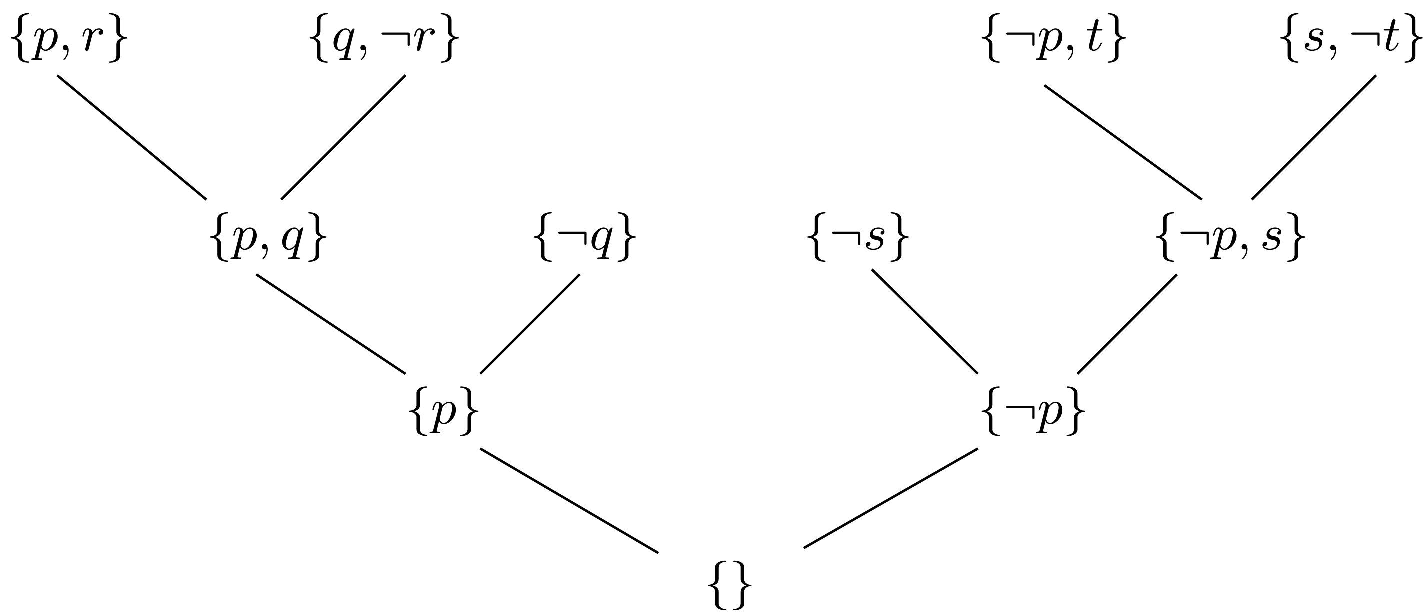


Example

$$S = (p \vee r) \wedge (r \Rightarrow q) \wedge \neg q \wedge (p \Rightarrow t) \wedge \neg s \wedge (t \Rightarrow s)$$

$$S = (p \vee r) \wedge (\neg r \vee q) \wedge \neg q \wedge (\neg p \vee t) \wedge \neg s \wedge (\neg t \vee s)$$

$$S = \{\{p, r\}, \{\neg r, q\}, \{\neg q\}, \{\neg p, t\}, \{\neg s\}, \{\neg t, s\}\}$$



Soundness and Completeness

Resolution for propositional logic is sound and complete.

Prenex Normal Form

A first-order formula is in *prenex normal form* (PNF), if it is written as a string of quantifiers followed by a quantifier-free part.

Every first-order formula has an equivalent formula in PNF. For example, formula

$$\forall x((\exists y A(y)) \vee ((\exists z B(z)) \to C(x)))$$

has the following PNF:

$$\forall x \exists y \forall z (A(y) \vee (B(z) \to C(x)))$$

Conversion to PNF

The rules for *conjunction* and *disjunction* say that

$(\forall x \phi) \wedge \psi$ is equivalent to $\forall x(\phi \wedge \psi)$
$(\forall x \phi) \vee \psi$ is equivalent to $\forall x(\phi \vee \psi)$

and

$(\exists x \phi) \wedge \psi$ is equivalent to $\exists x(\phi \wedge \psi)$
$(\exists x \phi) \vee \psi$ is equivalent to $\exists x(\phi \vee \psi)$

Conversion to PNF

The rules for *negation* say that

$\neg \exists x \phi$ is equivalent to $\forall x \neg \phi$

and

$\neg \forall x \phi$ is equivalent to $\exists x \neg \phi$

Conversion to PNF

The rules for *removing quantifiers* from the antecedent are:

$(\forall x \phi) \rightarrow \psi$ is equivalent to $\exists x(\phi \rightarrow \psi)$
$(\exists x \phi) \rightarrow \psi$ is equivalent to $\forall x(\phi \rightarrow \psi)$

The rules for *removing quantifiers* from the consequent are:

$\phi \rightarrow (\exists x \psi)$ is equivalent to $\exists x(\phi \rightarrow \psi)$
$\phi \rightarrow (\forall x \psi)$ is equivalent to $\forall x(\phi \rightarrow \psi)$

Example

Suppose that ϕ, ψ, and ρ are quantifier-free formulas and no two of these formulas share any free variable. The formula

$$(\phi \vee \exists x \psi) \rightarrow \forall z \rho$$

can be transformed into PNF as follows:

$$(\exists x(\phi \vee \psi)) \rightarrow \forall z \rho$$
$$\forall x((\phi \vee \psi) \rightarrow \forall z \rho)$$
$$\forall x(\forall z((\phi \vee \psi) \rightarrow \rho))$$
$$\forall x \forall z((\phi \vee \psi) \rightarrow \rho)$$

NNF

A formula is in *negation normal form* if negation occurs only immediately above propositions, and $\{\neg, \vee, \wedge\}$ are the only allowed Boolean connectives.

It is possible to convert any first-order formula to an equivalent formula in NNF:

$$\neg(\forall x.G) \rightarrow \exists x.\neg G$$
$$\neg(\exists x.G) \rightarrow \forall x.\neg G$$
$$\neg\neg G \rightarrow G$$
$$\neg(G_1 \wedge G_2) \rightarrow (\neg G_1) \vee (\neg G_2)$$
$$\neg(G_1 \vee G_2) \rightarrow (\neg G_1) \wedge (\neg G_2)$$

SNM: Skolemization

Reduction to *Skolem normal form* is a method for removing existential quantifiers from first-order formulas

A first-order formula is in SNF, if it is in conjunctive PNF with only universal first-order quantifiers.

Important note: Skolemization only preserves *satisfiability*.

Skolemization

Skolemization is performed by replacing every existentially quantified variable y with a term $f(x_1, \ldots, x_n)$ where function f does not occur anywhere else in the formula.

If the formula is in PNF, $x_1, \ldots, x_n$ are the variables that are universally quantified where quantifiers precede that of y. The function f is called a *Skolem function*.

Skolemization

In general,

$$\forall x_1 \ldots x_k \exists y . \varphi(x_1 \ldots x_k, y) \rightarrow \\ \forall x_1 \ldots x_k . \varphi(x_1 \ldots x_k, f(x_1 \ldots x_k))$$

For example, the formula

$$\forall x \exists y \forall z . P(x, y, z)$$

is not in SNF. Skolemization results in

$$\forall x \forall z . P(x, f(x), z)$$

Ground Clauses

A sentence A is in *clause form* iff it is a conjunction of (prenex) sentences of the form $\forall x_1 \ldots \forall x_m.C$, where C is a disjunction of literals, and the sets of bound variables $\{x_1, \ldots, x_m\}$ are disjoint for any two distinct clauses.

Each sentence $\forall x_1 \ldots \forall x_m.C$ is called a *clause*.

If a clause in A has no quantifiers and does not contain any variables, we say that it is a *ground clause*.

Ground Clauses

Lemma. For every sentence A, a sentence B in clause form such that A is valid iff B is unsatisfiable can be constructed.

Example

Let
$A = \neg\exists y.\forall z.(P(z,y) \Leftrightarrow \neg\exists x.(P(z,x) \wedge P(x,z)))$.

First, we negate A and eliminate $\Leftrightarrow$:

$$\exists y.\forall z.[(\neg P(z,y) \vee \neg\exists x.(P(z,x) \wedge P(x,z)))\wedge$$

$$(\exists x.(P(z,x) \wedge P(x,z)) \vee P(z,y))]$$

Example

Next, we put in this formula in NNF:

$$\exists y.\forall z.[(\neg P(z,y) \vee \forall x.(\neg P(z,x) \vee \neg P(x,z))) \wedge$$

$$(\exists x.(P(z,x) \wedge P(x,z)) \vee P(z,y))]$$

Next, we Skolemize:

$$\forall z.[(\neg P(z,a) \vee \forall x.(\neg P(z,x) \vee \neg P(x,z))) \wedge$$

$$((P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))]$$

Example

We now put in prenex form:

$$\forall z.\forall x.[(\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z)))\wedge$$

$$((P(z,f(z)) \wedge P(f(z),z)) \vee P(z,a))]$$

We put in CNF by distributing $\wedge$ over $\vee$:

$$\forall z.\forall x.[(\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z)))\wedge$$

$$(P(z,f(z)) \vee P(z,a)) \wedge (P(f(z),z)) \vee P(z,a))]$$

Example

Omitting universal quantifiers, we have the following three clauses:

$$C_1 = (\neg P(z_1, a) \vee (\neg P(z_1, x) \vee \neg P(x, z_1))$$

$$C_2 = (P(z_2, f(z_2)) \vee P(z_2, a))$$

$$C_3 = (P(f(z_3), z_3)) \vee P(z_3, a))]$$

Ground Resolution

Suppose, we want to prove (for the previous example) that $B = \neg A$ is unsatisfiable.

The *ground resolution* method is the resolution method applied to sets of ground clauses.

Ground Resolution

For example,

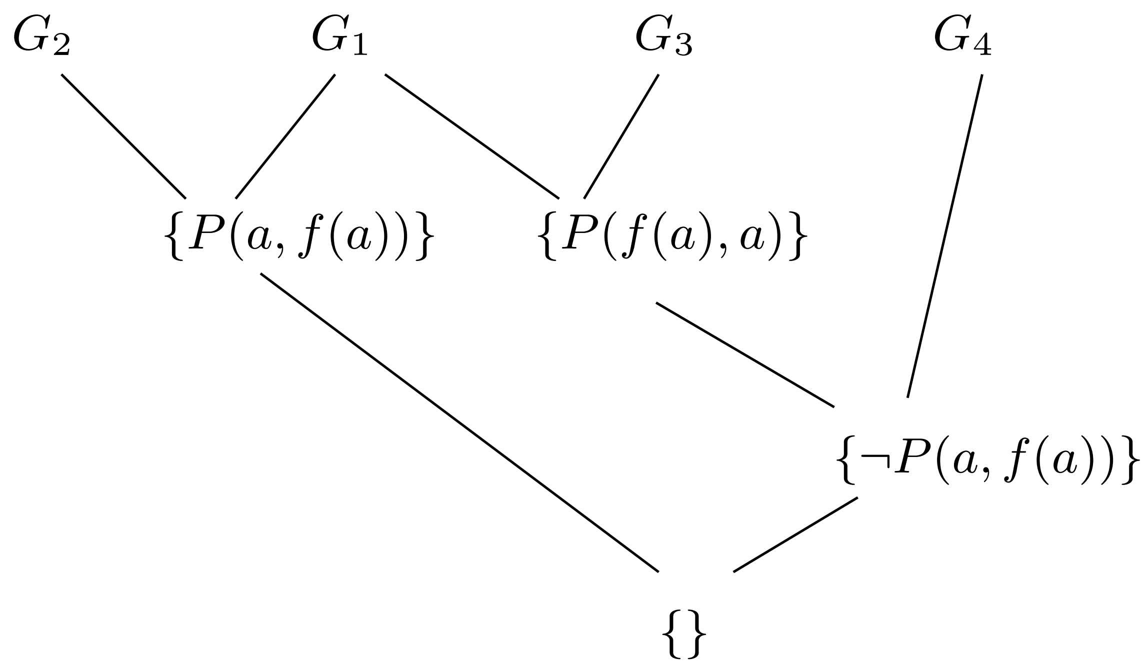
$G_1 = (\neg P(a, a))$
(from C_1, substituting a for x and z_1)

$G_2 = (P(a, f(a)) \vee P(a, a))$
(from C_2, substituting a for z_2)

$G_3 = (P(f(a), a)) \vee P(a, a))$
(from C_3, substituting a for z_3)

$G_4 = (\neg P(f(a), a) \vee \neg P(a, f(a)))$
(from C_1, substituting $f(a)$ for z_1 and a for x)

Example



Unification

To generalize ground resolution to arbitrary clauses, one is allowed to apply substitutions to the parent clauses.

For example, to obtain $\{P(a, f(a))\}$ from

$C_1 = (\neg P(z_1, a) \vee \neg P(z_1, x) \vee \neg P(x, z_1))$ and
$C_2 = (P(z_2, f(z_2)) \vee P(z_2, a))$,
first we substitute a for z_1, a for x, and a for z_2, obtaining

$$G_1 = (\neg P(a, a)) \text{ and } G_2 = (P(a, f(a)) \vee P(a, a))$$

and then we resolve on the literal $P(a, a)$.

Unification

Note that the two sets of literals $\{P(z_1, a), P(z_1, x), P(x, z_1)\}$ and $\{P(z_2, a)\}$ obtained by dropping the negation sign in C_1 have been *unified* by the substitution $(a/x, a/z1, a/z2)$.

Given two terms t and t' that do not share any variables, a substitution θ is called a *unifier* iff

$$\theta(t) = \theta(t')$$

Example

1. Let $t_1 = f(x, g(y))$ and $t_2 = f(g(u), g(z))$. The substitution $(g(u)/x, y/z)$ is a most general unifier yielding the most common instance $f(g(u), g(y))$.
2. However, $t_1 = f(x, g(y))$ and $t_2 = f(g(u), h(z))$ are not unifiable since this requires $g = h$.
3. Let $t_1 = f(x, g(x), x)$ and $t_2 = f(g(u), g(g(z)), z)$. To unify these two, we must have $x = g(u) = z$. But we also need $g(x) = g(g(z))$, that is, $x = g(z)$. This implies $z = g(z)$.

General Resolution

1. Find two clauses containing the same predicate, where it is negated in one clause but not in the other.
2. Perform a unification on the two predicates. (If the unification fails, you made a bad choice of predicates. Go back to the previous step and try again.)
3. If any unbound variables which were bound in the unified predicates also occur in other predicates in the two clauses, replace them with their bound values (terms) there as well.
4. Discard the unified predicates, and combine the remaining ones from the two clauses into a new clause, also joined by the $\vee$ operator.

Example

For clauses

$$A = (\neg P(z,a) \vee (\neg P(z,x) \vee \neg P(x,z))$$
$$B = (P(z,f(z)) \vee P(z,a))$$

We choose subsets $A' = A$ and $B' = \{P(z,a)\}$ and and unifier $(a/z, a/x)$, we obtain resolvent

$$C = \{P(a, f(a)\}$$

Example

$$C_1 = (\neg P(z_1, a) \vee (\neg P(z_1, x) \vee \neg P(x, z_1))$$

$$C_2 = (P(z_2, f(z_2)) \vee P(z_2, a))$$

$$C_3 = (P(f(z_3), z_3)) \vee P(z_3, a))]$$

Example