Mathematical Logics

17 Resolution and Unification

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

November 21, 2013

The rule of Propositional Resolution

$$\textbf{RES} \qquad \frac{A \vee C, \qquad B \vee \neg C}{A \vee B}$$

The formula $A \vee B$ is called a resolvent of $A \vee C$ and $B \vee \neg C$, denoted $Res(A \vee C, B \vee \neg C)$.

Exercize

Show that the Resolution rule is logically sound; i.e., that the conclusion is a logical consequence of the premise

RES inference rules assumes that the formulas are in normal form (CNF)

Clausal normal forms - (CNF)

- A clause is essentially an elementary disjunction $l_1 \vee \cdots \vee l_n$ but written as a (possibly empty) set of literals $\{l_1, \dots, l_n\}$.
- The empty clause $\{\}$ is a clause containing no literals. and therefore it is not satisfiable
- A unit clause is a clause containing only one literal.
- A clausal form is a (possibly empty) set of clauses, written as a list: $C_1 \dots C_k$ it represents the conjunction of these clauses.

Every formula in CNF can be re-written in a clausal form, and therefore every propositional formula is equivalent to one in a clausal form.

Example (Clausal form)

the clausal form of the CNF-formula $(p \vee \neg q \vee \neg r) \wedge \neg p \wedge (\neg q \vee r)$ is $\{p, \neg q, \neg r\}, \{\neg p\}, \{\neg q, r\}$
Note that the empty clause $\{\}$ (sometimes denoted by $\Box$) is not satisfiable (being an empty disjunction)

Clausal Propositional Resolution rule

The Propositional Resolution rule can be rewritten for clauses:

$$RES\,\frac{A_1, \dots, C, \dots, A_m\} \quad \{B_1, \dots, \neg C, \dots, B_n\}}{\{A1, \dots, A_m, B_1, \dots, B_n\}}$$

- The clause $\{A_1, \dots, A_m, B_1, \dots, B_n\}$ is called a resolvent of the clauses $\{A_1, \dots, C, \dots, A_m\}$ and $\{B_1, \dots, \neg C, \dots, B_n\}$.

Example (Applications of RES rule)

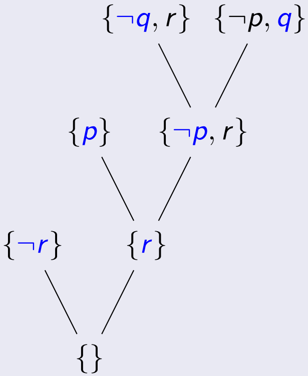
$$\frac{\{p, q, \neg r\} \quad \{\neg q, \neg r\}}{\{p, \neg r, \neg r\}} \qquad \frac{\{\neg p, q, \neg r\} \quad \{r\}}{\{\neg p, q\}} \qquad \frac{\{\neg p\} \quad \{p\}}{\{\}}$$

The rule of Propositional Resolution

Example

Try to apply the rule **RES** to the following two set of clauses
$\{\{\neg p, q\}, \{\neg q, r\}, \{p\}, \{\neg r\}\}$

Solution



Some remarks

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

- Note that two clauses can have more than one resolvent, e.g.:

$$\frac{\{p, \neg q\} \quad \{\neg p, q\}}{\{\neg q, q\}} \qquad \frac{\{\neg p, q\} \quad \{p, \neg p\}}{\{\neg p, p\}}$$

However, it is wrong to apply the Propositional Resolution rule for both pairs of complementary literals simultaneously as follows:

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

Sometimes, the resolvent can (and should) be simplified, by removing duplicated literals on the fly:

$$\{A_1, \ldots, C, C, \ldots, A_m\} \Rightarrow \{A_1, \ldots, C, \ldots, A_m\}.$$

For instance:

$$\frac{\{p, \neg q, \neg r\} \quad \{q, \neg r\}}{\{p, \neg r\}} \quad \text{instead of} \quad \frac{\{p, \neg q, \neg r\} \quad \{q, \neg r\}}{\{p, \neg r, \neg r\}}$$

Propositional resolution as a refutation system

- The underlying idea of Propositional Resolution is like the one of Semantic Tableau: in order to prove the validity of a logical consequence $A_1, \ldots, A_n \vdash B$, show that the set of formulas $\{A_1, \ldots, A_n, \neg B\}$ is Unsatisfiable
- That is done by transforming the formulae $A_1, \ldots, A_n$ and $\neg B$ into a clausal form, and then using repeatedly the Propositional Resolution rule in attempt to derive the empty clause $\{\}$.
- Since $\{\}$ is not satisfiable, its derivation means that $\{A_1, \ldots, A_n, \neg B\}$ cannot be satisfied together. Then, the logical consequence $A_1, \ldots, A_n \vdash B$ holds.
- Alternatively, after finitely many applications of the Propositional Resolution rule, no new applications of the rule remain possible. If the empty clause is not derived by then, it cannot be derived at all, and hence the $\{A_1, \ldots, A_n, \neg B\}$ can be satisfied together, so the logical consequence $A_1, \ldots, A_n \vdash B$ does not hold.

Propositional resolution - Examples

Example

- Check whether $(\neg p \supset q), \neg r \vdash p \vee (\neg q \wedge \neg r)$ holds.
- Check whether $p \supset q, q \supset r \models p \supset r$ holds.
- Show that the following set of clauses is unsatisfiable $\{\{A, B, \neg D\}, \{A, B, C, D\}, \{\neg B, C\}, \{\neg A\}, \{\neg C\}\}$

First-order resolution

- The Propositional Resolution rule in clausal form extended to first-order logic:

$$\frac{\{A_1, \ldots, Q(s_1, \ldots, s_n), \ldots, A_m\} \quad \{B_1, \ldots, \neg Q(s_1, \ldots, s_n), \ldots, B_n\}}{\{a_1, \ldots, a_m, b_1, \ldots, b_n\}}$$

  this rule, however, is not strong enough.
- **example:** consider the clause set

$$\{\{p(x)\}, \{\neg p(f(y))\}\}$$

  is not satisfiable, as it corresponds to the unsatisfiable formula

$$\forall x \forall y.(p(x) \land \neg p(f(y)))$$

- however, the resolution rule above cannot derive an empty clause from that clause set, because it cannot unify the two clauses in order to resolve them.
- so, we need a stronger resolution rule, i.e., a rule capable to understand that x and $f(y)$ can be instantiated to the same ground term $f(a)$.

Unification

Finding a common instance of two terms.

Intuition in combination with Resolution

$$S = \left\{ \begin{array}{c} friend(x, y) \supset friend(y, x) \\ friend(x, y) \supset knows(x, mother(y)) \\ friend(Mary, John) \\ \neg knows(John, mother(Mary)) \end{array} \right\}$$

$$cnf(S) = \left\{ \begin{array}{c} \neg friend(x, y) \vee friend(y, x) \\ \neg friend(x, y) \vee knows(x, mother(y)) \\ friend(Mary, John) \\ \neg knows(John, mother(Mary)) \end{array} \right\}$$

Is $cnf(S)$ satisfiable or unsatisfiable?
The key point here is to apply the right substitutions

Substitutions: A Mathematical Treatment

A substitution is a finite set of replacements

$$\sigma = [t_1/x_1, \ldots, t_k/x_k]$$

where $x_1, \ldots, x_k$ are distinct variables and $t_i \neq x_i$.

$t\sigma$ represents the result of the substitution σ applied to t.

$c\sigma = c$	(non) substitution of constants
$x[t_1/x_1, \ldots t_n/x_n] = t_i$ if $x = x_i$ for some i	substitution of variables
$x[t_1/x_1, \ldots t_n/x_n] = x$ if $x \neq x_i$ for all i	(non) substitution of variables
$f(t, u)\sigma = f(t\sigma, u\sigma)$	substitution in terms
$P(t, u)\sigma = P(t\sigma, u\sigma)$	...in literals
$\{L_1, \ldots, L_m\}\sigma = \{L_1\sigma, \ldots, L_m\sigma\}$	...in clauses

Composing Substitutions

Composition of σ and θ written $\sigma \circ \theta$, satisfies for all terms t

$$t(\sigma \circ \theta) = (t\theta)\sigma$$

If $\sigma = [t_1/x_1, \dots t_n/x_n]$ and $\theta = [u_1/x_1, \dots u_n/x_n]$, then

$$\sigma \circ \theta = [t_1\theta/x_1, \dots t_n\theta/x_n]$$

Identity substitution

$$[x/x, t_1/x_1, \dots t_n/x_n] = [t_1/x_1, \dots t_n/x_n]$$

$$\sigma \circ [] = \sigma$$

Associativity

$$\sigma \circ (\theta \circ \phi) = (\sigma \circ \theta) \circ \phi = \sigma \circ \theta \circ \phi =$$

Non commutativity, in general we have that

$$\sigma\theta \neq \theta\sigma$$

Composition of substitutions - example

$$
\begin{aligned}
f(g(x), f(y,x))[f(x,y)/x][g(a)/x, x/y] &= \\
f(g(f(x,y)), f(y, f(x,y)))[g(a)/x, x/y] &= \\
f(g(f(g(a), x)), f(x, f(g(a), x)))&
\end{aligned}
$$

$$
\begin{aligned}
f(g(x), f(y,x))[g(a)/x, x/y][f(x,y)/x] &= \\
f(g(g(a)), f(x, g(a)))[f(x,y)/x] &= \\
f(g(g(a)), f(f(x,y), g(a)))&
\end{aligned}
$$

Computing the composition of substitutions

The composition of two substitutions $\tau = [t_1/x_1, \dots, t_k/x_k]$ and σ

1. Extend the replaced variables of τ with the variables that are replaced in σ but not in τ with the identity substitution x/x
2. Apply the substitution simultaneously to all terms $[t_1, \dots, t_k]$ to obtaining the substitution $[t_1\sigma/x_1, \dots, t_k\sigma/x_k]$.
3. Remove from the result all cases x_i/x_i , if any.

Example

$$\begin{aligned}
[f(x,y)/x, x/y][y/x, a/y, g(y)/z] &= \\
[f(x,y)/x, x/y, z/z][y/x, a/y, g(y)/z] &= \\
[f(y,a)/x, y/y, g(y)/z] &= \\
[f(y,a)/x, g(y)/z]&
\end{aligned}$$

Unifiers and Most General Unifiers

σ is a unifier of terms t and u if $t\sigma = u\sigma$.
For instance

- the substitution $[f(y)/x]$ unifies the terms x and $f(y)$
- the substitution $[f(c)/x, c/y, c/z]$ unifies the terms $g(x, f(f(z)))$ and $g(f(y), f(x))$
- There is no unifier for the pair of terms $f(x)$ and $g(y)$, nor for the pair of terms $f(x)$ and x.

σ is more general than θ if $\theta = \sigma \circ \phi$ for some substitution ϕ.

σ is a most general unifier for two terms t and u if it a unifier for t and u and it is more general of all the unifiers of t and u.

If σ unifies t and u then so does $\sigma \circ \theta$ for any θ.

A most general unifier of $f(a, x)$ and $f(y, g(z))$ is $\sigma = [a/y, g(z)/x]$. The common instance is

$$f(a, x)\sigma = f(a, g(z)) = f(y, g(z))\sigma$$

Unifier

Example

The substitution $[3/x, g(3)/y]$ unifies the terms $g(g(x))$ and $g(y)$. The common instance is g(g(3)). This is not however the most general unifier for these two terms. Indeed, these terms have many other unifiers, including the following:

unifying substitution	common instance
$[f(u)/x, g(f(u))/y]$	$g(g(f(u)))$
$[z/x, g(z)/y]$	$g(g(z))$
$[g(x)/y]$	$g(g(x))$

$[g(x)/y]$ is also the most general unifier.

Examples of most general unifier

Notation: $x, y, z \ldots$ are variables, $a, b, c, \ldots$ are constants $f, g, h, \ldots$ are functions $p, q, r, \ldots$ are predicates.

terms	**MGU**	**result of the substitution**
$p(a, b, c)$ $p(x, y, z)$	$[a/x, b/y, c/z]$	$p(a, b, c)$
$p(x, x)$ $p(a, b)$	*None*	
$p(f(g(x, a), x)$ $p(z, b)$	$[b/x, f(g(b, a))/z]$	$p(f(g(b, a), b)$
$p(f(x, y), z)$ $p(z, f(a, y))$	$[f(a, y)/z, a/x]$	$p(f(a, y), f(a, y))$

Unification Algorithm: Preparation

We shall formulate a unification algorithm for literals only, but it can easily be adapted to work with formulas and terms.

Sub expressions Let L be a literal. We refer to formulas and terms appearing within L as the *subexpressions* of L. If there is a subexpression in L starting at position i we call it $L^{(i)}$ (otherwise (i) is undefined.

Disagreement pairs. Let L_1 and L_2 be literals with $L_1 \neq L_2$. The disagreement pair of L_1 and L_2 is the pair $(L_1^{(i)}, L_2^{(i)})$ of subexpressions of L_1 and L_2 respectively, where i is the smallest number such that $L_1^{(i)} \neq L_2^{(i)}$).

Example The disagreement pair of

$$P(g(c), f(a, g(x), h(a, g(b))))$$
$$P(g(c), f(a, g(x), h(k(x, y), z)))$$

is $(a, k(x, y))$

Robinson's Unification Algorithm

Imput: a set of literals Δ
Output: $\sigma = MGU(\Delta$ or Undefined!

$\sigma := []$
while $|\Delta\sigma| > 1$ **do**
    pick a disagreement pair p in $\Delta\sigma$'
    if no variable in p **then**
        return 'not unifiable';
    else
        let $p = (x, t)$ with x being a variable;
        if x occurs in t **then**
            return 'not unifiable';
        else $\sigma := \sigma \circ [t/x]$;
return σ

Theorem-Proving Example

$$(\exists y \forall x R(x, y)) \supset (\forall x \exists y R(x, y))$$

Negate $\neg((\exists y \forall x R(x, y)) \supset (\forall x \exists y R(x, y)))$

NNF $\exists y \forall x R(x, y), \quad \exists x \forall y \neg R(x, y)$

Skolemize $R(x, b), \quad \neg R(a, y)$

Unify $MGU(R(x, b), R(a, y)) = [b/x, a/y]$

Contrad.: We have the contradiction $R(b, a), \neg R(b, a)$, so the formula is valid

Theorem-Proving Example

$$(\forall x \exists y R(x, y)) \supset (\exists y \forall x R(x, y))$$

Negate $\neg((\forall x \exists y R(x, y)) \supset (\exists y \forall x R(x, y)))$

NNF $\forall x \exists y R(x, y), \quad \forall y \exists x \neg R(x, y)$

Skolemize $R(x, f(x)), \quad \neg R(g(y), y)$

Unify $MGU(R(x, f(x)), \quad \neg R(g(y), y)) = \mathit{Undefined}$

Contrad.: We do not have the contradiction, so the formula is not valid.

Resolution for first order logic

The resolution rule for Propositional logic is

$$\frac{\{l_1, \dots, l_n, p\} \qquad \{\neg p, l_{n+1}, \dots, l_m\}}{\{l_1, \dots l_m\}}$$

The binary resolution rule

In first order logic each l_i and p are formulas of the form $P(t_1, \dots, t_n)$ or $\neg P(t_1, \dots, t_n)$.

When two opposite literals of the form $P(t_1, \dots, t_n)$ and $P(u_1, \dots, u_n)$ occur in the clauses C_1 and C_2 respectively, we have to find a way to partially instantiate them, by a substitution σ, in such a way the resolution rule can be applied, to to $C_1\sigma$ and $C_2\sigma$, i.e., such that $P(t_1, \dots, t_n)\sigma = P(u_1, \dots, u_n)\sigma$.

$$\frac{\{l_1, \dots, l_n, P(t_1, \dots, t_n)\}\{\neg P(u_1, \dots, u_n), l_{n+1}, \dots, l_m\}}{\{l_1, \dots l_m\}\sigma}$$

where σ is the $MGU(P(t_1, \dots, t_n), P(u_1, \dots, u_n))$.

The factoring rule

$$\frac{\{l_1, \ldots, l_n, l_{n+1}, \ldots, l_m\}}{\{l_1, l_{n+1}, \ldots l_m\}\sigma} \quad \text{If } l_1\sigma = \cdots = l_n\sigma$$

Example

Prove $\forall x \exists y \neg(P(y,x) \equiv \neg P(y,y))$

Clausal form $\{\neg P(y,a),\ \neg P(y,y)\},\ \{P(y,y),\ P(y,a)\}$

Factoring yields $\{\neg P(a,a)\},\ \{P(a,a)\}$

By resolution rule we obtain the empty clauses $\square$

A Non-Trivial Proof

$$\exists x[P \supset Q(x)] \wedge \exists x[Q(x) \supset P] \supset \exists x[P \equiv Q(x)]$$

Clauses are $\{P, \neg Q(b)\},\ \{P, Q(x)\},\ \{\neg P, \neg Q(x)\},\ \{\neg P, Q(a)\}$

Apply resolution

$$\begin{tikzcd}[column sep=small]
\{\neg P, Q(a)\} \arrow[dr, no head] & & \{\neg P, \neg Q(x)\} \arrow[dl, no head] & & \{P, Q(x)\} \arrow[dr, no head] & & \{P, \neg Q(b)\} \arrow[dl, no head] \\
& \{\neg P\} \arrow[drr, no head] & & & & \{P\} \arrow[dll, no head] & \\
& & & \{\} & & &
\end{tikzcd}$$

Equality

In theory, it's enough to add the equality axioms:

- The reflexive, symmetric and transitive laws $\{x = x\}$, $\{x \neq y, y = x\}$, $\{x \neq y, y \neq z, x = z\}$.
- Substitution laws like $\{x_1 \neq y_1, \dots, x_n \neq y_n, f(x_1, \dots, x_n) = f(y_1, \dots, y_n)\}$ for each f with arity equal to n
- Substitution laws like $\{x_1 \neq y_1, \dots, x_n \neq y_n, \neg P(x_1, \dots, x_n),\ P(y_1, \dots, y_n)\}$ for each P with arity equal to n

In practice, we need something special: the paramodulation rule

$$\frac{\{P(t), l_1, \dots l_n\} \quad \{u = v, l_{n+1}, \dots, l_m\}}{P(v), l_1, \dots, l_m\}\sigma} \quad \text{provides that } t\sigma = u\sigma$$