Knowledge Representation and Learning

11. Resolution and Unification

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

May 22, 2023

The rule of Propositional Resolution

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

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

Exercise 1:

Show that the Resolution rule is logically sound; i.e., that the conclusion is a logical consequence of the premises. In other words shaow that

$$A \vee C, B \vee \neg C \models A \vee B$$

RES allows to infer new (true) clauses from other clauses. To apply **RES** to a set of formulas we firsty have to transform them in CNF (set of clauses).

Soundness of Propositional Resolution

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

To prove soundness of the **RES** rule we show that the following logical consequence holds:

$$(A \vee C) \wedge (\neg C \vee B) \models A \vee B$$

i.e., we have to show that, for every interpretation $\mathcal{I}$,

$$\text{if } \mathcal{I} \models (A \vee C) \wedge (\neg C \vee B)\text{, then } \mathcal{I} \models A \vee B$$

- Suppose that $\mathcal{I} \models (A \vee C) \wedge (\neg C \vee B)$, then $\mathcal{I} \models (A \vee C)$ and $\mathcal{I} \neg C \vee B)$
- This implies that $\mathcal{I} \models A \vee C$, and therefore that either $\mathcal{I} \models A$ or $\mathcal{I} \models C$
    - If $\mathcal{I} \models A$, then $\mathcal{I} \models A \vee B$
    - If $\mathcal{I} \models C$, then from the fact that $\mathcal{I} \models \neg C \vee B$ we have that $\mathcal{I} \models B$. Which implies that $\mathcal{I} \models A \vee B$.

Generality of Propositional Resolution

The propositional resolution inference rule implements a very general inference pattern, that includes many inference rules of propositional logics once the formulas are transformed in CNF.

Rule Name	Original form	CNF form
Modus Ponens	$\dfrac{p \quad p \rightarrow q}{q}$	$\dfrac{\{p\} \quad \{\neg p, q\}}{\{q\}}$
Modus tollens	$\dfrac{\neg q \quad p \rightarrow q}{\neg p}$	$\dfrac{\{\neg q\} \quad \{\neg p, q\}}{\{\neg p\}}$
Chaining	$\dfrac{p \rightarrow q \quad q \rightarrow r}{p \rightarrow r}$	$\dfrac{\{\neg p, q\} \quad \{\neg q, r\}}{\{\neg p, r\}}$
Reductio ad absurdum	$\dfrac{p \rightarrow q \quad p \rightarrow \neg q}{\neg p}$	$\dfrac{\{\neg p, q\} \quad \{\neg p, \neg q\}}{\{\neg p\}}$
Reasoning by case	$\dfrac{p \vee q \quad p \rightarrow r \quad q \rightarrow r}{r}$	$\dfrac{\dfrac{\{p, q\} \quad \{\neg p, r\}}{\{q, r\}} \qquad \{\neg q, r\}}{\{r\}}$
Tertium non datur	$\dfrac{p \quad \neg p}{\bot}$	$\dfrac{\{p\} \quad \{\neg p\}}{\{\}}$

Propositional Resolution rule

The Propositional Resolution rule is the general form of the rules presented in the previous slides. Using the setwise notation it can be written as:

$$\textbf{RES:}\ \frac{A_1, \ldots, C, \ldots, A_m\} \quad \{B_1, \ldots, \neg C, \ldots, B_n\}}{\{A1, \ldots, A_m, B_1, \ldots, B_n\}}$$

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

Example (Applications of RES rule)

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

Propositinal resolution: Decision Procedure

- Using **RES** it is possible to build a decision procedure that decides if a set of formulas are satisfiable.
- To check if a set of propositional formulas Γ is satisfiable, you have transform Γ conjunctive normal and apply PROPOSITIONALRESOCUTION algorithm.

Propositional resolution

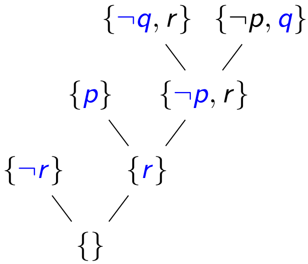
function PROPOSITIONALRESOLUTION(Γ:CNF)
    while no new clauses are derivable **do**
        $C_1, C_2, p \leftarrow$ select two clauses and an atom from Γ such that $p \in C_1$ and $\neg p \in C_2$, and such that (C_1, C_2, p) has not previously selected
        $\Gamma \leftarrow \Gamma \cup \{(C_1 \cup C_2) \setminus \{p, \neg p\}\}$
        if $\{\} \in \Gamma$ **then**
            return Unsat
        end if
    end while
    return Sat
end function

- This simple algorithm terminates, since the number of clauses that can be build using the propositional variables occurring in Γ are finite.
- Differently from DPLL this decision procedure, if the set of formulas Γ are satisfiable, does not necessarily provide a model for it. The proceduire provides only yes/no anser.

Example

Decide if the following set of clauses are satisfiable using PropositionalResolution.

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



Solution □

Propositional Resolution - Examples

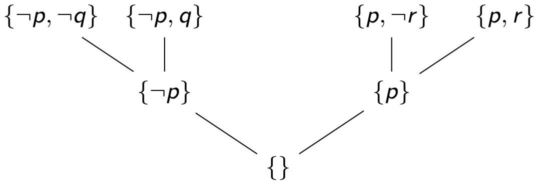
Example

Show that the following set of formulas are not satisfiable by PROPOSITIONALRESOLUTION.

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

Solution We first transform the formulas in clauses obtaining:

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



□

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:

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

For instance:

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

Deciding Validity and logical consequence with Propositional resolution

- Propositional Resolution, like DPLL,can be used to prove the validity of a formula and the logical consequence of a formula from a set of formulas.
- to check that $\models \phi$, (i.e., that ϕ is valid you can check that $\neg\phi$ is not satisfiable by transforming $\neg\phi$ in CNF and apply PropositionalResolution.
- To check if $\phi_1, \ldots, \phi_n \models \phi$, you have to check if the set of formulas $\{\phi_1, \phi_2 \ldots, \phi_n, \neg\phi\}$ is not satisfiable by applying Propositionalresolution to the CNF conversion of ϕ_i and $\neg\phi$.

Propositional resolution - Exercizes

Exercizes

Check the following facts via propositional resolution

1. $(\neg p \rightarrow q), \neg r \models p \vee (\neg q \wedge \neg r)$
2. $p \rightarrow q, q \rightarrow r \models p \rightarrow r$
3. The set of clauses $\{\{A, B, \neg D\}, \{A, B, C, D\}, \{\neg B, C\}, \{\neg A\}, \{\neg C\}\}$ is unsatisfiable

First-order resolution

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

$$\frac{\{A_1, \dots, Q(s_1, \dots, s_n), \dots, A_m\} \quad \{B_1, \dots, \neg Q(s_1, \dots, s_n), \dots, B_n\}}{\{A_1, \dots, A_m, B_1, \dots, B_n\}}$$

  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

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

- 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

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

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

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

First order logic Resolution

- Let Γ a set of first order clauses, i.e., formulas of the form

$$\forall x_1 \ldots x_n \phi(x_1, \ldots, x_n)$$

  where $\phi(x_1, \ldots, x_n)$ is a disjunction of literals not containing quantifiers.
- let H be Herbrand universe of Γ, i.e., the set of ground terms that can be builded with the signature of Γ.
- let Γ_H be the set of clauses $\phi(t_1, \ldots, t_n)$ obbtained by grounding the clauses in Γ with all the possible n-tuple of terms of the Herbrand universe.
- Γ_H can be infinite. but Herbrand theorem guarantees that if Γ is unsat, then there is a funite subset of Γ_H that is unsat.
- theoretically, if Γ is unsat, then by applying $\textsc{PropositionalResolution}$ to Γ_H we eventually derive the empty clause.

Substitutions: A Mathematical Treatment

A substitution is a finite set of replacements

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

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
$x[x_1/t_1, \ldots x_n/t_n] = t_i$ if $x = x_i$ for some i	substitution of variables
$x[x_1/t_1, \ldots x_n/t_n] = x$ 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

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

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

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

Identity substitution

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

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

Associativity

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

Non commutativity, in general we have that

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

Composition of substitutions - examples

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

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

Computing the composition of substitutions

The composition of two substitutions $\tau = [t_1/x_1, \ldots, t_k/x_k]$ 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 σ simultaneously to all terms $[t_1, \ldots, t_k]$ to obtaining the substitution $[x_1/t_1\sigma, \ldots, x_k/t_k\sigma/]$.
3. Remove from the result all cases x_i/x_i , if any.

Example

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

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

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

- The one marked in red are MGU
- **Exercize:** Show that the first substitution can be obtained by composing a MGU with another substitution

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
$p(a, b, c)$ $p(x, y, z)$	$[x/a, y/b, z/c]$	$p(a, b, c)$
$p(x, x)$ $p(a, b)$	*None*	
$p(f(g(x, a), x)$ $p(z, b)$	$[x/b, z/f(g(b, a))]$	$p(f(g(b, a), b)$
$p(f(x, y), z)$ $p(z, f(a, y))$	$[z/f(a, y), x/a]$	$p(f(a, y), f(a, y))$

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 $L^{(i)}$ 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

$$
\begin{array}{c}
P(g(c), f(a, g(x), h(a, g(b))))\\
P(g(c), f(a, g(x), h(k(x, y), z)))\\
\uparrow
\end{array}
$$

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

Input: a set of terms 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 $\sigma := \sigma \circ [x/t]$;
return σ

Substitution

Exercise 2:

Let $\sigma = [x/a, y/f(b), z/c]$ and $\theta = [v/f(f(a)), z/x, x/g(y)]$

- compute $\sigma \circ \theta$ and $\theta \circ \sigma$
- For every of the following formulæ, compute (i) $\phi\sigma$; (ii) $\phi\theta$; (iii) $\phi\sigma \circ \theta$; and (iv) $\phi\theta \circ \sigma$
  1. $\phi = p(x, y, z)$
  2. $\phi = p(h(v)) \vee \neg q(z, x)$
  3. $\phi = q(x, z, v) \vee \neg q(g(y), x, f(f(a)))$
- are σ and θ and their compositions idempotent?

Definition

A function $f : X \longrightarrow X$ on a set X is idempotent if and only if $f(x) = f(f(x))$

An example of idempotent function are $round(\cdot) : \mathbb{R} \longrightarrow \mathbb{R}$, that returns the closer integer $round(x)$ to a real number x.

Unification

Exercise 3:

For every C_1, C_2 and σ, decide whether (i) σ is a unifier of C_1 and C_2; and (ii) σ is the MGU of C_1 and C_2

C_1	C_2	σ
$P(a, f(y), z)$	$Q(x, f(f(v)), b)$	$[x/a, y/f(b), z/b]$
$Q(x, h(a, z), f(x))$	$Q(g(g(v)), y, f(w))$	$[x/g(g(v)), y/h(a, z), w/x]$
$Q(x, h(a, z), f(x))$	$Q(g(g(v)), y, f(w))$	$[x/g(g(v)), y/h(a, z), w/g(g(v))]$
$R(f(x), g(y))$	$R(z, g(v))$	$[x/a, z/f(a), y/v]$

Unification

Exercise 4:

Consider the signature $\Sigma = \langle a, b, f(\cdot,\cdot), g(\cdot,\cdot), P(\cdot,\cdot,\cdot)\rangle$ Use the algorithm from the previous lecture to decide whether the following clauses are unifiable.

1. $\{P(f(x,a), g(y,y), z), P(f(g(a,b),z), x, a)\}$
2. $\{P(x,x,z), P(f(a,a), y, y)\}$
3. $\{P(x, f(y,z), b), P(g(a,y), f(z, g(a,x)), b)\}$
4. $\{P(a,y,U), P(x, f(x,U), g(z,b))\}$

Unification of $P(f(x, a), g(y, y), z)$, $P(f(g(a, b), z), x, a)$

- $\{P(f(x, a), g(y, y), Z), P(f(g(a, b), z), x, a)\}$
- $\sigma = [x/g(a, b)]$
- $\{P(f(x, a), g(y, y), Z), P(f(g(a, b), z), x, a)\}\sigma =$
  $\{P(f(g(a, b), a), g(y, y), z), P(f(g(a, b), z), g(a, b), a)\}.$
- $\{P(f(g(a, b), a), g(y, y), z), P(f(g(a, b), z), g(a, b), a)\}.$
- $\sigma = [x/g(a, b), z/a]$
- $\{P(f(g(a, b), a), g(y, y), z), P(f(g(a, b), z), g(a, b), a)\}\sigma =$
  $\{P(f(g(a, b), a), g(y, y), a), P(f(g(a, b), a), g(a, b), a)\}$
- $\{P(f(g(a, b), a), g(y, y), a), P(f(g(a, b), a), g(a, b), a)\}$
- $\sigma = [x/g(a, b), z/, y/a]$
- $\{P(f(g(a, b), a), g(y, y), a), P(f(g(a, b), a), g(a, b), a)\}\sigma =$
  $\{P(f(g(a, b), a), g(a, a), a), P(f(g(a, b), a), g(a, b), a)\}$
- $\{P(f(g(a, b), a), g(a, a), a), P(f(g(a, b), a), g(a, b), a)\}$
- a and b are two constants and they are not unificable. So the algorithm returns that the set of clauses are not unifiable.

Unification of $\{P(x,x,z), P(f(a,a),y,y)\}$

- $\{P(x,x,z), P(f(a,a),y,y)\}$
- $\sigma = [x/f(a,a)]$
- $\{P(x,x,z), P(f(a,a),y,y)\}\sigma =$
  $\{P(f(a,a),f(a,a),z), P(f(a,a),y,y)\}$
- $\{P(f(a,a),f(a,a),z), P(f(a,a),y,y)\}$
- $\sigma = [x/f(a,a), y/f(a,a)]$
- $\{P(f(a,a),f(a,a),z), P(f(a,a),y,y)\}\sigma =$
  $\{P(f(a,a),f(a,a),z), P(f(a,a),f(a,a),f(a,a))\}$
- $\{P(f(a,a),f(a,a),z), P(f(a,a),f(a,a),f(a,a))\}$
- $\sigma = [x/f(a,a), y/f(a,a), z/f(a,a)]$
- $\{P(f(a,a),f(a,a),z), P(f(a,a),f(a,a),f(a,a))\}\sigma =$
  $\{P(f(a,a),f(a,a),f(a,a)), P(f(a,a),f(a,a),f(a,a))\}$
- the two terms are equal, so the initial terms are unifiable with the mgu equal to $\sigma = [x/f(a,a), y/f(a,a), z/f(a,a)]$

Unification

Exercise 5:

Find, when possible, the MGU of the following pairs of clauses.

1. $\{q(a), q(b)\}$
2. $\{q(a, x), q(a, a)\}$
3. $\{q(a, x, f(x)), q(a, y, y,)\}$
4. $\{q(x, y, z), q(u, h(v, v), u)\}$
5. $\left\{ \begin{array}{l} p(x_1, g(x_1), x_2, h(x_1, x_2), x_3, k(x_1, x_2, x_3)), \\ p(y_1, y_2, e(y_2), y_3, f(y_2, y_3), y_4) \end{array} \right\}$

Theorem-Proving Example

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

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

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

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

Theorem-Proving Example

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

Negate	$\neg((\forall x \exists y R(x,y)) \rightarrow (\exists y \forall x R(x,y)))$
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	$MGU(R(x, f(x)), \quad R(g(y), y)) = \mathit{Undefined}$
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 $P(t_1, \ldots, t_n)$ or $\neg P(t_1, \ldots, t_n)$.

When two opposite literals of the form $P(t_1, \ldots, t_n)$ and $P(u_1, \ldots, u_n)$ 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 $P(t_1, \ldots, t_n)\sigma = P(u_1, \ldots, u_n)\sigma$.

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

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

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

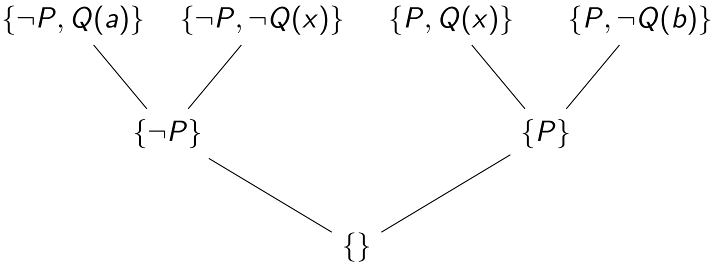
By resolution rule we obtain the empty clauses $\square$

A Non-Trivial Proof

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

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

Apply resolution



Example

Assumptions:

- $\forall x(P(x) \rightarrow P(f(x)))$
- $\forall x, y(Q(a,y) \wedge R(y,x) \rightarrow P(x))$
- $\forall z R(b, g(a,z))$
- $Q(a,b)$

Goal $= P(f(g(a,c)))$

1. clausify the assumptions
2. negate and clausify the goal
3. $mgu(Q(a,y), Q(a,b)) = [y/b]$
4. $mgu(R(b,g(a,z)), R(b,x)) = [x/g(a,z)]$
5. $mgu(P(x), P(g(a,z)) = [x/g(a,z)]$
6. $mgu(P(f(g(a,z))), P(f(g(a,c)))) = [z/c]$

Inference

1. $\neg P(x), P(f(x))$
2. $\neg Q(a,y), \neg R(y,x), P(x)$
3. $R(b, g(a,z))$
4. $Q(a,b)$
5. $\neg P(f(g(a,c)))$
6. $\neg R(b,x), P(x)$
7. $P(g(a,z))$
8. $P(f(g(a,z)))$
9. $\bot$

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 $\{x_1 \neq y_1, \ldots, x_n \neq y_n, f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)\}$ for each f with arity equal to n
- Substitution laws like $\{x_1 \neq y_1, \ldots, x_n \neq y_n, \neg P(x_1, \ldots, x_n),\ P(y_1, \ldots, y_n)\}$ for each P with arity equal to n

In practice, we need something special: the paramodulation rule

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

Exercise 6:

Find the possible resolvents of the following pairs of clauses.

C	D
$\neg p(x) \vee q(x,b)$	$p(a) \vee q(a,b)$
$\neg p(x) \vee q(x,x)$	$\neg q(a, f(a))$
$\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$	$p(g(x,y),x,y)$
$\neg p(v,z,v) \vee p(w,z,w)$	$p(w, h(x,x), w)$

Solution

C	D	σ
$\neg p(x) \vee q(x,b)$	$p(a) \vee q(a,b)$	$[x/a]$
$\neg p(x) \vee q(x,x)$	$\neg q(a, f(a))$	NO
$\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$	$p(g(x',y'),x',y')$	
$\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$	$p(g(x',y'),x',y')$	
$\neg p(x,y,u) \vee \neg p(y,z,v) \vee \neg p(x,v,w) \vee p(u,z,w)$	$p(g(x',y'),x',y')$	
$\neg p(v,z,v) \vee p(w,z,w)$	$p(w', h(x',x'), w')$	

Exercise 7:

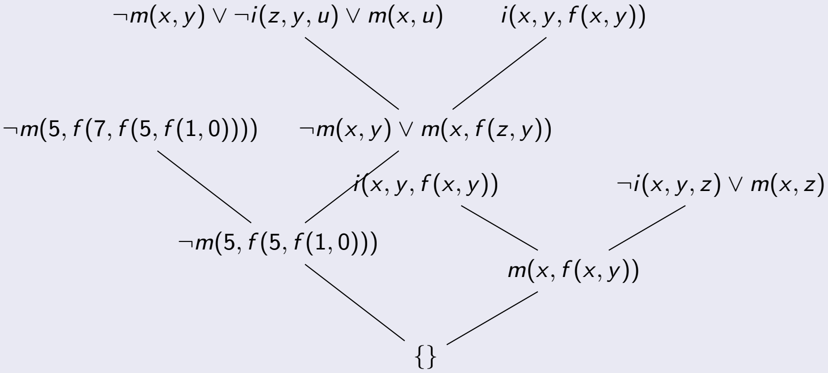
Apply resolution (with refutation) to prove that the following formula

$$\phi_5 \quad m(5, f(7, f(5, f(1, 0))))$$

is a consequence of the set

$$\begin{array}{ll} \phi_1 & \neg m(x, 0) \\ \phi_2 & \neg i(x, y, z) \vee m(x, z) \\ \phi_3 & \neg m(x, z) \vee \neg i(v, z, y) \vee m(x, y) \\ \phi_4 & i(x, y, f(x, y)) \end{array}$$

Solution



Notice that variables in clauses can be renamed in any way to facilitate unification. So for instance in ϕ_3 we rename variables in order to unify with ϕ_4.

Resolution and Unification - Exercize

Exercise

Show that the following set of formulas are not satisfiable:

1. $\forall x(P(x) \wedge \neg Q(x) \rightarrow \exists y(R(x, y) \wedge S(y)))$
2. $\exists x(P(x) \wedge T(x))$
3. $\forall x(\exists y R(y, x) \rightarrow T(x))$
4. $\forall x(T(x) \rightarrow \neg(Q(x) \vee S(x)))$

Resolution and Unification - Exercize

Solution we first transform the formula in first order clausal form, and rename variables.

- $\{\neg P(x), Q(x), R(x, f(x))\}$ (from formula 1. we introduce the skolem function f)
- $\{\neg P(y), Q(y), S(f(xy))\}$ (from formula 1.)
- $\{T(a)\}$ (from formula 2.we introduce the Skolem constant a)
- $\{P(a)\}$ (from formula 2.we introduce the Skolem constant a)
- $\{\neg R(z, w), T(z)\}$ (from formula 3.)
- $\{\neg T(v), \neg Q(v)\}$ (from formula 4.)
- $\{\neg T(u), \neg S(u)\}$ (from formula 4.)

□