Knowledge Representation and Learning

11. Resolution and Unification

Luciano Serafini

Fondazione Bruno Kessler, Trento, Italy

May 22, 2023
The rule of Propositional Resolution

\[
\text{RES} \quad \frac{A \lor C, \quad \neg C \lor B}{A \lor B}
\]

The formula \(A \lor B\) is called a resolvent of \(A \lor C\) and \(B \lor \neg C\), denoted \(\text{Res}(A \lor C, B \lor \neg C)\).

**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 show that

\[A \lor C, B \lor \neg C \models A \lor B\]

\textbf{RES} allows to infer new (true) clauses from other clauses. To apply \textbf{RES} to a set of formulas we firstly have to transform them in CNF (set of clauses).
Soundness of Propositional Resolution

\[
\text{RES} \quad \frac{A \lor C, \quad \lnot C \lor B}{A \lor B}
\]

To prove soundness of the \text{RES} rule we show that the following logical consequence holds:

\[(A \lor C) \land (\lnot C \lor B) \models A \lor B\]

i.e., we have to show that, for every interpretation \(I\),

\[
\text{if } I \models (A \lor C) \land (\lnot C \lor B), \text{ then } I \models A \lor B
\]

- Suppose that \(I \models (A \lor C) \land (\lnot C \lor B)\), then \(I \models (A \lor C)\) and \(I \lnot C \lor B\)
- This implies that \(I \models A \lor C\), and therefore that either \(I \models A\) or \(I \models C\)
  - If \(I \models A\), then \(I \models A \lor B\)
  - If \(I \models C\), then from the fact that \(I \models \lnot C \lor B\) we have that \(I \models B\). Which implies that \(I \models A \lor B\).
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.

<table>
<thead>
<tr>
<th>Rule Name</th>
<th>Original form</th>
<th>CNF form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modus Ponens</td>
<td>[ p \ p \rightarrow q ] [ q ]</td>
<td>[ {p} {\neg p, q}] [ {q}]</td>
</tr>
<tr>
<td>Modus tollens</td>
<td>[ \neg q \ p \rightarrow q ] [ \neg p ]</td>
<td>[ {\neg q} {\neg p, q}] [ {\neg p}]</td>
</tr>
<tr>
<td>Chaining</td>
<td>[ p \rightarrow q \ q \rightarrow r ] [ p \rightarrow r ]</td>
<td>[ {\neg p, q} {\neg q, r}] [ {\neg p, r}]</td>
</tr>
<tr>
<td>Reductio ad absurdum</td>
<td>[ p \rightarrow q \ p \rightarrow \neg q ] [ \neg p ]</td>
<td>[ {\neg p, q} {\neg p, \neg q}] [ {\neg p}]</td>
</tr>
<tr>
<td>Reasoning by case</td>
<td>[ p \lor q \ p \rightarrow r \ q \rightarrow r ] [ r ]</td>
<td>[ {p, q} {\neg p, r}] [ {q, r} {\neg q, r}] [ {r}]</td>
</tr>
<tr>
<td>Tertium non datur</td>
<td>[ p \neg p ] [ \bot ]</td>
<td>[ {p} {\neg p}] [ {}]</td>
</tr>
</tbody>
</table>
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:

\[
\text{RES: } \frac{A_1, \ldots, C, \ldots, A_m}{\{B_1, \ldots, \neg C, \ldots, B_n\}} \frac{\{A_1, \ldots, A_m, B_1, \ldots, B_n\}}{\{A_1, \ldots, A_m, B_1, \ldots, B_n\}}
\]

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)**

\[
\begin{align*}
\{p, q, \neg r\} & \quad \{\neg q, \neg r\} & \quad \{\neg p, q, \neg r\} & \quad \{r\} & \quad \{\neg p\} & \quad \{p\} \\
\{p, \neg r, \neg r\} & \quad & \{\neg p, q\} & \quad & \{} & \quad
\end{align*}
\]
Propositional 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 $\Gamma$ is satisfiable, you have transform $\Gamma$ conjunctive normal and apply PropositionalResolution algorithm.

### Propositional resolution

```plaintext
1: function PropositionalResolution($\Gamma$:CNF)
2:   while no new clauses are derivable do
3:     $C_1, C_2, p \leftarrow$ select two clauses and an atom from $\Gamma$ such that $p \in C_1$ and $\neg p \in C_2$, and such that $(C_1, C_2, p)$ has not previously selected
4:     $\Gamma \leftarrow \Gamma \cup \{(C_1 \cup C_2) \setminus \{p, \neg p\}\}$
5:     if $\{\} \in \Gamma$ then
6:       return Unsat
7:     end if
8:   end while
9:  return Sat
10: end function
```

- This simple algorithm terminates, since the number of clauses that can be build using the propositional variables occurring in $\Gamma$ are finite.

- Differently from DPLL this decision procedure, if the set of formulas $\Gamma$ are satisfiable, does not necessarily provide a model for it. The procedure provides only yes/no answer.
Propositional Resolution - Examples

Example

Decide if the following set of clauses are satisfiable using Propositional Resolution.

\[
\{\{\neg p, q\}, \{\neg q, r\}, \{p\}, \{\neg r\}\}
\]

Solution

\[
\{\}
\]

\[\square\]
Example

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

\[ \{ p \rightarrow q, p \rightarrow \neg q, \neg p \rightarrow r, \neg p \rightarrow \neg r \} \]

Solution We first transform the formulas in clauses obtaining:

\[ \{ \neg p, q \}, \{ \neg p, \neg q \}, \{ p, r \}, \{ p, \neg r \} \]

\[
\begin{array}{ccc}
\{ \neg p, \neg q \} & | & \{ \neg p, q \} \\
\{ \neg p \} & | & \{ p, \neg r \} \\
\{ \} & | & \{ p \} \\
\{ \} & | & \{ p, r \}
\end{array}
\]
Some remarks

\[
\begin{array}{c}
\{p, q, \neg r\} \quad \{\neg q, \neg r\} \\
\{p, \neg r, \neg r\}
\end{array}
\quad
\begin{array}{c}
\{\neg p, q, \neg r\} \quad \{r\} \\
\{\neg p, q\}
\end{array}
\quad
\begin{array}{c}
\{\neg p\} \quad \{p\} \\
\{\}
\end{array}
\]

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

\[
\begin{array}{c}
\{p, \neg q\} \quad \{\neg p, q\} \\
\{\neg q, q\}
\end{array}
\quad
\begin{array}{c}
\{\neg p, q\} \quad \{p, \neg p\} \\
\{\neg p, p\}
\end{array}
\]

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

\[
\begin{array}{c}
\{p, \neg q\} \quad \{\neg p, q\} \\
\{\}
\end{array}
\]

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:

\[
\begin{array}{c}
\{p, \neg q, \neg r\} \quad \{q, \neg r\} \\
\{p, \neg r\}
\end{array}
\quad
\begin{array}{c}
\{p, \neg q, \neg r\} \quad \{q, \neg r\} \\
\{p, \neg r, \neg r\}
\end{array}
\]

Luciano Serafini (Fondazione Bruno Kessler, Trento, Italy)
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 $\phi$ 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 $\phi_i$ and $\neg \phi$. 
Exercises

Check the following facts via propositional resolution

1. \((\neg p \rightarrow q), \neg r \models p \lor (\neg q \land \neg r)\)

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:

\[
\begin{align*}
\{A_1, \ldots, Q(s_1, \ldots, s_n), \ldots, A_m\} & \quad \{B_1, \ldots, \neg Q(s_1, \ldots, s_n), \ldots, B_n\} \\
\{A_1, \ldots, A_m, B_1, \ldots, B_n\}
\end{align*}
\]

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

\[
S = \left\{ 
\begin{array}{l}
\text{friend}(x, y) \rightarrow \text{friend}(y, x) \\
\text{friend}(x, y) \rightarrow \text{knows}(x, \text{mother}(y)) \\
\text{friend}(\text{Mary}, \text{John}) \\
\neg\text{knows}(\text{John}, \text{mother}(\text{Mary}))
\end{array}
\right.
\]

\[
cnf(S) = \left\{ 
\begin{array}{l}
\neg\text{friend}(x, y) \lor \text{friend}(y, x) \\
\neg\text{friend}(x, y) \lor \text{knows}(x, \text{mother}(y)) \\
\text{friend}(\text{Mary}, \text{John}) \\
\neg\text{knows}(\text{John}, \text{mother}(\text{Mary}))
\end{array}
\right.
\]

Is $cnf(S)$ satisfiable or unsatisfiable?
The key point here is to apply the right substitutions
Let $\Gamma$ 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 $\Gamma$, i.e., the set of ground terms that can be built with the signature of $\Gamma$.

Let $\Gamma_H$ be the set of clauses $\phi(t_1, \ldots, t_n)$ obtained by grounding the clauses in $\Gamma$ with all the possible $n$-tuple of terms of the Herbrand universe.

$\Gamma_H$ can be infinite. But Herbrand theorem guarantees that if $\Gamma$ is unsat, then there is a finite subset of $\Gamma_H$ that is unsat.

Theoretically, if $\Gamma$ is unsat, then by applying Propositional Resolution to $\Gamma_H$ we eventually derive the empty clause.
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 \( \sigma \) applied to \( t \).

\[
\begin{align*}
    c\sigma &= c & \text{(non) substitution of constants} \\
    x[x_1/t_1, \ldots x_n/t_n] &= t_i \text{ if } x = x_i \text{ for some } i \\
    x[x_1/t_1, \ldots x_n/t_n] &= x \text{ if } x \neq x_i \text{ for all } i \\
    f(t, u)\sigma &= f(t\sigma, u\sigma) & \text{substitution in terms} \\
    P(t, u)\sigma &= P(t\sigma, u\sigma) & \ldots \text{in literals} \\
    \{L_1, \ldots, L_m\}\sigma &= \{L_1\sigma, \ldots, L_m\sigma\} & \ldots \text{in clauses}
\end{align*}
\]
Composing Substitutions

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

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

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

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

Identity substitution

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

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

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

\[ f(g(x), f(y, x))\big[x/f(x, y)\big][x/g(a), y/x] = \]
\[ f(g(f(x, y)), f(y, f(x, y)))\big[x/g(a), y/x\big] = \]
\[ f(g(f(g(a), x)), f(x, f(g(a), x))) \]

\[ f(g(x), f(y, x))\big[x/g(a), y/x\big][x/f(x, y)] = \]
\[ f(g(g(a)), f(x, g(a)))\big[x/f(x, y)\big] = \]
\[ f(g(g(a)), f(f(x, y), g(a))) \]
Computing the composition of substitutions

The composition of two substitutions \( \tau = [t_1/x_1, \ldots, t_k/x_k] \) and \( \sigma \)

1. Extend the replaced variables of \( \tau \) with the variables that are replaced in \( \sigma \) but not in \( \tau \) with the identity substitution \( x/x \)

2. Apply the substitution \( \sigma \) 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

\[
[x/f(x, y), y/x][x/y, y/a, z/g(y)] =
[x/f(x, y), y/x, z/z][x/y, y/a, z/g(y)] =
[x/f(y, a), y/y, z/g(y)] =
[x/f(y, a), z/g(y)]
\]
Unifiers and Most General Unifiers

\(\sigma\) 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 \(x\).

\(\sigma\) is **more general than** \(\theta\) if \(\theta = \sigma \circ \phi\) for some substitution \(\phi\).

\(\sigma\) 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 \(\sigma\) unifies \(t\) and \(u\) then so does \(\sigma \circ \theta\) for any \(\theta\).

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

\[f(a, x)\sigma = f(a, g(z)) = f(y, g(z))\sigma\]
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:

<table>
<thead>
<tr>
<th>unifying substitution</th>
<th>common instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>([x/f(u), y/g(f(u))])</td>
<td>(g(g(f(u))))</td>
</tr>
<tr>
<td>([x/z, y/g(z)])</td>
<td>(g(g(z)))</td>
</tr>
<tr>
<td>([y/g(x)])</td>
<td>(g(g(x)))</td>
</tr>
</tbody>
</table>

The one marked in red are MGU.

**Exercise:** 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.

<table>
<thead>
<tr>
<th>Terms</th>
<th>MGU</th>
<th>Result of the substitution</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p(a, b, c))</td>
<td>([x/a, y/b, z/c])</td>
<td>(p(a, b, c))</td>
</tr>
<tr>
<td>(p(x, y, z))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(p(x, x))</td>
<td>(None)</td>
<td></td>
</tr>
<tr>
<td>(p(a, b))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(p(f(g(x, a), x)))</td>
<td>([x/b, z/f(g(b, a))])</td>
<td>(p(f(g(b, a), b)))</td>
</tr>
<tr>
<td>(p(z, b))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(p(f(x, y), z))</td>
<td>([z/f(a, y), x/a])</td>
<td>(p(f(a, y), f(a, y)))</td>
</tr>
<tr>
<td>(p(z, f(a, y)))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

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

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

↑ is $(a, k(x, y))$.
Robinson’s Unification Algorithm

**Input:** a set of terms $\Delta$

**Output:** $\sigma = \text{MGU}(\Delta)$ or Undefined!

\[
\sigma := [] \\
\textbf{while} |\Delta \sigma| > 1 \textbf{ do} \\
\quad \text{pick a disagreement pair } p \text{ in } \Delta \sigma' \\
\quad \textbf{if} \text{ no variable in } p \text{ then} \\
\quad \quad \text{return} \text{ ‘not unifiable’}; \\
\quad \textbf{else} \\
\quad \quad \text{let } p = (x, t) \text{ with } x \text{ being a variable;} \\
\quad \quad \textbf{if} \ x \text{ occurs in } t \text{ then} \\
\quad \quad \quad \text{return} \text{ ‘not unifiable’}; \\
\quad \quad \textbf{else} \ \sigma := \sigma \circ [x/t]; \\
\quad \text{return } \sigma
\]
Exercise 2:
Let $\sigma = [x/a, y/f(b), z/c] \text{ and } \theta = [v/f(f(a)), z/x, x/g(y)]$

- 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. $\phi = p(h(v)) \lor \neg q(z, x)$
3. $\phi = q(x, z, v) \lor \neg q(g(y), x, f(f(a)))$

- are $\sigma$ and $\theta$ and their compositions idempotent?

Definition

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

An example of idempotent function are $\text{round}(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$, that returns the closer integer $\text{round}(x)$ to a real number $x$. 
Exercise 3:
For every $C_1$, $C_2$ and $\sigma$, decide whether (i) $\sigma$ is a unifier of $C_1$ and $C_2$; and (ii) $\sigma$ is the MGU of $C_1$ and $C_2$

<table>
<thead>
<tr>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(a, f(y), z)$</td>
<td>$Q(x, f(f(v)), b)$</td>
<td>$[x/a, y/f(b), z/b]$</td>
</tr>
<tr>
<td>$Q(x, h(a, z), f(x))$</td>
<td>$Q(g(g(v)), y, f(w))$</td>
<td>$[x/g(g(v)), y/h(a, z), w/x]$</td>
</tr>
<tr>
<td>$Q(x, h(a, z), f(x))$</td>
<td>$Q(g(g(v)), y, f(w))$</td>
<td>$[x/g(g(v)), y/h(a, z), w/g(g(v))]$</td>
</tr>
<tr>
<td>$R(f(x), g(y))$</td>
<td>$R(z, g(v))$</td>
<td>$[x/a, z/f(a), y/v]$</td>
</tr>
</tbody>
</table>
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 unifiable. So the algorithm returns that the set of clauses are not unifiable.
Unification of \( \{P(x, x, z), P(f(a, a), y, y)\} \)

1. \( \{P(x, x, z), P(f(a, a), y, y)\} \)
2. \( \sigma = [x/f(a, a)] \)
3. \( \{P(x, x, z), P(f(a, a), y, y)\}\sigma = \)
   \( \{P(f(a, a), f(a, a), z), P(f(a, a), y, y)\} \)
4. \( \{P(f(a, a), f(a, a), z), P(f(a, a), y, y)\} \)
5. \( \sigma = [x/f(a, a), y/f(a, a)] \)
6. \( \{P(f(a, a), f(a, a), z), P(f(a, a), y, y)\}\sigma = \)
   \( \{P(f(a, a), f(a, a), z), P(f(a, a), f(a, a), f(a, a))\} \)
7. \( \{P(f(a, a), f(a, a), z), P(f(a, a), f(a, a), f(a, a))\}\sigma = \)
   \( \{P(f(a, a), f(a, a), f(a, a)), P(f(a, a), f(a, a), f(a, a))\} \)
8. the two terms are equal, so the initial terms are unifiable with the mgu equal to \( \sigma = [x/f(a, a), y/f(a, a), z/f(a, a)] \)
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 $\exists y \forall x R(x, y), \exists x \forall y \neg R(x, y)$

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

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 \(\forall x \exists y R(x, y), \ \forall y \exists x \neg R(x, y)\)

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

Unify \(MGU(R(x, f(x)), R(g(y), y)) = Undefined\)

Contrad.: We do not have the contradiction, so the formula is not valid.
The resolution rule for Propositional logic is

\[
\begin{align*}
\{l_1, \ldots, l_n, p\} & \quad \{\neg p, l_{n+1}, \ldots, l_m\} \\
\{l_1, \ldots l_m\} & \\
\end{align*}
\]
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 \( \sigma \), in such a way the resolution rule can be applied, to to \( C_1 \sigma \) and \( C_2 \sigma \), i.e., such that
\[
P(t_1, \ldots, t_n)\sigma = P(u_1, \ldots, u_n)\sigma.
\]

\[
\begin{array}{c}
\{l_1, \ldots, l_n, P(t_1, \ldots, t_n)\} \{\neg P(u_1, \ldots, u_n), l_{n+1}, \ldots, l_m\} \\
\{l_1, \ldots, l_m\} \sigma
\end{array}
\]

where \( \sigma \) is the \( MGU(P(t_1, \ldots, t_n), P(u_1, \ldots, u_n)) \).
The factoring rule

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

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 □
A Non-Trivial Proof

\[ \exists x [P \rightarrow Q(x)] \land \exists x [Q(x) \rightarrow P] \rightarrow \exists x [P \equiv Q(x)] \]

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

Apply resolution

\{\neg P, Q(a)\} \quad \{\neg P, \neg Q(x)\} \quad \{P, Q(x)\} \quad \{P, \neg Q(b)\}

\{\neg P\} \quad \{P\} 

\{\}\
Example

Assumptions:

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

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
  \[ \{x_1 \neq y_1, \ldots, x_n \neq y_n, f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)\} \text{ for each } f \text{ with arity equal to } n \]

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

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.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\neg p(x) \lor q(x, b)$</td>
<td>$\neg p(x) \lor q(x, x)$</td>
<td>$p(a) \lor q(a, b)$</td>
</tr>
<tr>
<td>$\neg p(x, y, u) \lor \neg p(y, z, v) \lor \neg p(x, v, w) \lor p(u, z, w)$</td>
<td>$\neg q(a, f(a))$</td>
<td>$p(g(x, y), x, y)$</td>
</tr>
<tr>
<td>$\neg p(v, z, v) \lor p(w, z, w)$</td>
<td>$p(g(x, y), x, y)$</td>
<td>$p(w, h(x, x), w)$</td>
</tr>
</tbody>
</table>

Solution

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\neg p(x) \lor q(x, b)$</td>
<td>$\neg q(a, f(a))$</td>
<td>$[x/a]$</td>
</tr>
<tr>
<td>$\neg p(x) \lor q(x, x)$</td>
<td>$p(g(x', y'), x', y')$</td>
<td>$NO$</td>
</tr>
<tr>
<td>$\neg p(x, y, u) \lor \neg p(y, z, v) \lor \neg p(x, v, w) \lor p(u, z, w)$</td>
<td>$p(g(x', y'), x', y')$</td>
<td></td>
</tr>
<tr>
<td>$\neg p(v, z, v) \lor p(w, z, w)$</td>
<td>$p(g(x', y'), x', y')$</td>
<td></td>
</tr>
</tbody>
</table>

Luciano Serafini (Fondazione Bruno Kessler, Knowledge Representation and Learning)
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

$$\begin{align*}
\phi_1 & \quad \neg m(x, 0) \\
\phi_2 & \quad \neg i(x, y, z) \lor m(x, z) \\
\phi_3 & \quad \neg m(x, z) \lor \neg i(v, z, y) \lor m(x, y) \\
\phi_4 & \quad i(x, y, f(x, y))
\end{align*}$$
Notice that variables in clauses can be renamed in any way to facilitate unification. So for instance in $\phi_3$ we rename variables in order to unify with $\phi_4$. 

Luciano Serafini (Fondazione Bruno Kessler, Trento, Italy)
Exercise

Show that the following set of formulas are not satisfiable:

1. $\forall x (P(x) \land \lnot Q(x) \rightarrow \exists y (R(x, y) \land S(y)))$
2. $\exists x (P(x) \land T(x))$
3. $\forall x (\exists y R(y, x) \rightarrow T(x))$
4. $\forall x (T(x) \rightarrow \lnot (Q(x) \lor S(x)))$
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.)