Substitution and Unification
Summary

- Substitution and Unification [Chang-Lee Ch. 5.3]
- Unification Algorithm [Chang-Lee Ch. 5.4]
Finding complementary literals

To apply resolution we need to find complementary literals: $L_1 = P, L_2 = \neg P$.

This is not a problem for ground or propositional clauses.

When variables are involved things get more complicated.

It is not obvious to decide whether two literals are complementary.
Example (complementary literals with variables)

\[ C_1 = P(x) \lor Q(x), \quad C_2 = \neg P(f(y)) \lor R(y) \]

There is no complementary literal, but:

\[ C'_1 = P(x = f(a)) \lor Q(x = f(a)), \quad C'_2 = \neg P(f(y = a)) \lor R(y = a) \]

Then \( C'_1 \) and \( C'_2 \) are ground instances of \( C_1 \) and \( C_2 \), and \( P(f(a)) \) and \( \neg P(f(a)) \) are complementary.
Example (complementary literals with variables)

Then we can apply resolution and obtain:

\[
\frac{P(f(a)) \lor Q(f(a)) \land \neg P(f(a)) \lor R(a)}{Q(f(a)) \lor R(a)}
\]

Where \( C'_3 = Q(f(a)) \lor R(a) \) is a resolvent for \( C'_1 \) and \( C'_2 \)
Example (complementary literals with variables)

More in general, we can substitute $x = f(y)$ in $C_1$ and obtain

$$C_1^* = P(f(y)) \lor Q(f(y))$$

$$P(f(y)) \lor Q(f(y)) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad

$C_1^*$ is an instance of $C_1$ and $C_3'$ is a (ground) instance of $C_3 = Q(f(y)) \lor R(y)$

- By substituting appropriate terms we can generate new clauses for $C_1$ and $C_2$
- By applying resolution to such clauses we obtain other clauses which will all be instance of $C_3$.
- $C_3$ is the most general clause and is called a resolvent of $C_1$ and $C_2$. 
Substitutions

To obtain a resolvent from clauses containing variables we need to substitute variables with terms, and apply resolution.

Definition (Substitution)

A substitution is a finite set \( \{ t_1/v_1, \ldots, t_n/v_n \} \) where every \( v_i \) is a variable and every \( t_i \) is a term, different from \( v_i \), and no two elements in the set have the same variable after the / symbol.
Example, contd.

Example (Substitution)

- \( \{f(z)/x, y/z\} \) is a substitution
- \( \{a/x, g(y)/y, f(g(b))/z\} \) is a substitution
- \( \{y/x, g(b)/y\} \) is a substitution
- \( \{a/x, g(y)/x, f(g(b))/z\} \) is not a substitution
- \( \{g(y)/x, z/f(g(b))\} \) is not a substitution
Ground and Empty substitutions

**Definition (Ground substitution)**

A substitution \( \{ t_1/\nu_1, \ldots, t_n/\nu_n \} \) is ground when \( \{ t_1, \ldots, t_n \} \) are all ground terms.

**Example (Ground Substitution)**

- \( \{ f(a)/x, b/z \} \) is a ground substitution
- \( \{ a/x, g(b)/y, f(g(b))/z \} \) is a ground substitution

**Definition (Empty substitution)**

A substitution that contains no element \( \{ \} \) is the empty substitution, we denote the empty substitution with \( \epsilon \).
Instances of clauses

Definition (Instance)

Let $\theta = \{t_1/v_1, \ldots, t_n/v_n\}$ be a substitution and let $E$ be an expression. Then $E\theta$ is an expression obtained by replacing **simultaneously** all occurrences of every $v_i$, $1 \leq i \leq n$, in $E$ with the corresponding term $t_i$. $E\theta$ is an instance of $E$. 

### Example (Instances)

- Let $\theta = \{a/x, f(b)/y, c/z\}$ and $E = P(x, y, z)$ then $E\theta = P(a, f(b), c)$ is an instance of $E$.
- Let $\lambda = \{f(f(a))/x\}$ and $C = P(x) \lor Q(g(x))$ then $G\lambda = P(f(f(a)) \lor Q(g(f(f(a))))$ is an instance of $C$.
- Let $\gamma = \{y/x, f(b)/y\}$ and $R = P(x) \lor Q(y)$ then $R\gamma = P(y) \lor Q(f(b))$ is an instance of $R$.

### Notes

- Definition of **ground** instance of a clause is compatible with definition of **instance** given here.
- substitution is **simultaneous**. If not simultaneous we could have different outcomes:
  \[ R\gamma = P(x \leftarrow y \leftarrow f(b)) \lor Q(y \leftarrow f(b)) \]
Composition

Let $\theta = \{t_1/x_1, \cdots, t_n/x_n\}$ and $\lambda = \{u_1/y_1, \cdots, u_m/y_m\}$ be two substitutions. Then the composition of $\theta$ and $\lambda$ is denoted by $\theta \circ \lambda$, and is obtained by building the set $\{t_1 \lambda/x_1, \cdots, t_n \lambda/x_n, u_n/y_1, \cdots, u_m/y_m\}$ and deleting the following elements:

- any element $t_j \lambda/x_j$ such that $t_j \lambda = x_j$
- any element $u_i/y_i$ such that $y_i$ is in $\{x_1, \cdots, x_n\}$
Example (composition)

Given:
- $\theta = \{t_1/x_1, t_2/x_2\} = \{f(y)/x, z/y\}$
- $\lambda = \{u_1/y_1, u_2/y_2, u_3/y_3\} = \{a/x, b/y, y/z\}$

We build the following set:

$\{t_1\lambda/x_1, t_2\lambda/x_2, u_1/y_1, u_2/y_2, u_3/y_3\} = \{f(b)/x, y/y, a/x, b/y, y/z\}$

We remove the proper elements:
- $t_j\lambda/x_j$ such that $t_j\lambda = x_j$ therefore we remove $y/y$
- $u_i/y_i$ such that $y_i$ is in $\{x_1, \cdots, x_n\}$ therefore we remove $a/x$ and $b/y$

Finally,

$\theta \circ \lambda = \{f(b)/x, y/z\}$
## Properties of composition

### Associativeness

Let \( \theta, \lambda, \text{ and } \mu \) be substitutions we have that \((\theta \circ \lambda) \circ \mu = \theta \circ (\lambda \circ \mu)\).

### Example

Let \( \theta = \{f(y)/x\}, \lambda = \{z/y\} \text{ and } \mu = \{a/z\}. \) We have

\[
\phi = \theta \circ \lambda = \{f(y)\lambda/x, z/y\} = \{f(z)/x, z/y\}\]
\[
\phi \circ \mu = \{f(z)\mu/x, z\mu/y, a/z\} = \{f(a)/x, a/y, a/z\}
\]
\[
\phi' = \lambda \circ \mu = \{z\mu/y, a/z\} = \{a/y, a/z\}\]
\[
\theta \circ \phi' = \{f(y)\phi'/x, a/y, a/z\} = \{f(a)/x, a/y, a/z\}
\]

### Identity of empty substitution

Let \( \theta \) be a substitution then \( \epsilon \circ \theta = \theta \circ \epsilon = \theta \).
Unification and substitutions

Unifying expressions using substitutions

- When using resolution we need to match or unify expressions to find complementary pairs of literals.
- This can be done by applying proper substitutions to relevant expression to make them identical.

Example

Let $C_1 = P(x) \lor Q(x)$ and $C_2 = \neg P(f(y)) \lor Q(y)$, and let $L_1 = P(x)$, $\neg L_2 = P(f(y))$
- By applying $\theta = \{ f(y)/x \}$.
- We have that $L_1 \theta = \neg L_2 \theta = P(f(y))$. 
**Unifier**

**Definition (Unifier)**

A substitution $\theta$ is called a unifier for a set $\{E_1, \ldots, E_k\}$ iff $E_1\theta = E_2\theta = \cdots = E_k\theta$. The set $\{E_1, \ldots, E_k\}$ is unifiable iff there exists a unifier for it.

**Example**

The set $\{P(x), P(f(y))\}$ is unifiable and $\theta = \{f(y)/x\}$ is a unifier for it, because $P(x)\theta = P(f(y))\theta = P(f(y))$.
Most General Unifier

**Definition (MGF)**

A unifier $\theta$ is a most general unifier for a set $\{E_1, \cdots, E_k\}$ iff for each unifier $\lambda$ there exists a substitution $\mu$ such that $\lambda = \theta \circ \mu$.

**Example**

Consider the set $\{P(x), P(f(y))\}$ and $\lambda = \{f(f(z))/x, f(z)/y\}$.

- $\lambda$ is a unifier because $P(x)\lambda = P(f(y))\lambda = P(f(f(z)))$
- $\theta = \{f(y)/x\}$ is a unifier and is a most general unifier
- For example, we can find $\mu = \{f(z)/y\}$ such that $\theta \circ \mu = \{f(y)\mu/x, f(z)/y\} = \{f(f(z))/x, f(z)/y\} = \lambda$
Example: Most General Unifier

Consider the set \( \{P(a, y), P(x, f(b))\} \)

- The set is unifiable
- \( \theta = \{a/x, f(b)/y\} \) is a (most general) unifier for the set.
An algorithm for Unification

Unification Algorithm

- Need a procedure to find a MGU given a set of expressions
- Requirements:
  - stop after a finite number of steps
  - return an MGU if the set is unifiable
  - state that the set is not unifiable otherwise
- There are many possibilities
- We go for a recursive procedure.
Basic ideas

- Given a set of expressions \( \{E_1, \cdots, E_k\} \)
- Find a disagreement set
- Build a substitution that can eliminate the disagreement
Example (Disagreement elimination)

Consider the set \( \{ P(a), P(x) \} \). This expressions are not identical.

- They disagree because of the arguments \( a \) and \( x \)
- The disagreement set here is \( \{ a, x \} \)
- Since \( x \) is a variable, we can eliminate this disagreement by using the substitution \( \theta = \{ a/x \} \)
- \( P(a)\theta = P(x)\theta = P(a) \)
Disagreement set

Definition (Disagreement Set)

The disagreement set of a nonempty set of expressions \( W \) is obtained by finding the first position (starting from the left) at which not all the expressions in the \( W \) have the same symbol. We then extract, from each expression, the sub-expression that begins with the symbol occupying that position. The set of these sub-expressions is the Disagreement Set.

Example (Disagreement Set)

Consider the set \( \{P(a), P(x)\} \), the Disagreement Set is \( \{a, x\} \). because the first position at which the string of symbols \( P(a) \) and \( P(x) \) differ is the position number 3. The sub expression starting from position 3 is \( a \) and \( x \) respectively.
Example (Disagreement Set)

Find the Disagreement Set for
\[ W = \{ P(x, f(y, z)), P(x, a), P(x, g(h(k(x)))) \} \]
Example (Disagreement Set)

Find the Disagreement Set for
\( W = \{ P(x, f(y, z)), P(x, a), P(x, g(h(k(x)))) \} \)

Sol.

\[
D = \{ f(y, z), a, g(h(k(x))) \}
\]
Unification Algorithm: Basic Steps

Basic Steps

1. Set $k = 0$, $W_0 = W$ and $\sigma_0 = \epsilon$

2. If $W_k$ is a singleton, STOP, $\sigma_k$ is a MGU. Otherwise, find the disagreement set $D_k$ for $W_k$.

3. If there is a pair $\langle v_k, t_k \rangle$ such that $v_k, t_k \in D_k$, $v_k$ is a variable that does not occur in $t_k$ go to step 4, otherwise STOP, $W$ is not unifiable.

4. Let $\sigma_{k+1} = \sigma_k \circ \{ t_k / v_k \}$ and $W_{k+1} = W_k \{ t_k / v_k \}$.

5. Set $k = k + 1$ go to step 2.

Note

In step 4 $W_{k+1} = W_k \{ t_k / v_k \} = W \sigma_{k+1}$ because composition of substitutions is associative.
Example (Unification Algorithm)

Find a most general unifier for the set
\[ W = \{ P(a, y), P(x, f(b)) \} \]
Example (Unification Algorithm)

Find a most general unifier for the set $W = \{ P(a, y), P(x, f(b)) \}$

Sol.

$\theta = \{ a/x, f(b)/y \}$
Example II

Example (Unification Algorithm)

Find a most general unifier for the set
\[ W = \{ P(a, x, f(g(y))), P(z, f(z), f(u)) \} \]
Example II

**Example (Unification Algorithm)**

Find a most general unifier for the set
\[ W = \{ P(a, x, f(g(y))), P(z, f(z), f(u)) \} \]

**Sol.**

\[ \theta = \{ a/z, f(a)/x, g(y)/u \} \]
Example III

Example (Unification Algorithm)

Determine whether or not the set $W = \{ Q(f(a), g(x)), Q(y, y) \}$ is unifiable.
Example III

Example (Unification Algorithm)

Determine whether or not the set $W = \{ Q(f(a), g(x)), Q(y, y) \}$ is unifiable.

Sol.

$W$ is not unifiable
Unification Algorithm: Termination

Termination

The unification algorithm will always terminate after a finite number of steps.

- Otherwise we will have an infinite sequence
  \( \mathcal{W}\sigma_0, \mathcal{W}\sigma_1, \mathcal{W}\sigma_2, \cdots \)

- Each step eliminates one variable from \( \mathcal{W} \) (specifically \( \mathcal{W}\sigma_k \) contains \( \nu_k \) but \( \mathcal{W}\sigma_{k+1} \) does not)

- And \( \mathcal{W} \) has a finite number of variable

- Thus the algorithm will always terminate: returning a MGU or stating \( \mathcal{W} \) is not unifiable
Theorem (Correctness of Unification Algorithm)

If \( W \) is a finite, non empty, unifiable set of expressions, the unification algorithm will always terminate with \( \sigma_k \) a MGU for \( W \).

basic idea.

We can prove by induction on \( k \) that for any \( \theta \) which is a unifier we have \( \theta = \sigma_k \circ \lambda_k \).
Exercise

Determine whether each of the following set of expressions is unifiable. If yes give a MGU

1. \( W = \{ Q(a, x, f(x)), Q(a, y, y) \} \)
2. \( W = \{ Q(x, y, z), Q(u, h(v, v), u) \} \)