2.5.4 Proposition (Renaming Variables)

Let $P$ be a propositional variable not occurring in $\psi[\phi]_p$.

1. If $\text{pol}(\psi, p) = 1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \land (P \rightarrow \phi)$ is satisfiable.

2. If $\text{pol}(\psi, p) = -1$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \land (\phi \rightarrow P)$ is satisfiable.

3. If $\text{pol}(\psi, p) = 0$, then $\psi[\phi]_p$ is satisfiable if and only if $\psi[P]_p \land (P \leftrightarrow \phi)$ is satisfiable.
Renaming

\textbf{SimpleRenaming} \quad \phi \Rightarrow_{\text{SimpRen}} \phi[P_1]_{p_1} [P_2]_{p_2} \cdots [P_n]_{p_n} \land \\
\text{def}(\phi, p_1, P_1) \land \cdots \land \text{def}(\phi[P_1]_{p_1} [P_2]_{p_2} \cdots [P_{n-1}]_{p_{n-1}}, p_n, P_n) \\
\text{provided } \{p_1, \ldots, p_n\} \subset \text{pos}(\phi) \text{ and for all } i, i + j \text{ either } p_i \parallel p_{i+j} \text{ or } p_i > p_{i+j} \text{ and the } P_i \text{ are different and new to } \phi

Simple choice: choose \{p_1, \ldots, p_n\} to be all non-literal and non-negation positions of \phi.
Renaming Definition

\[
\text{def}(\psi, p, P) := \begin{cases} 
(P \rightarrow \psi|_p) & \text{if } \text{pol}(\psi, p) = 1 \\
(\psi|_p \rightarrow P) & \text{if } \text{pol}(\psi, p) = -1 \\
(P \leftrightarrow \psi|_p) & \text{if } \text{pol}(\psi, p) = 0
\end{cases}
\]
Obvious Positions

A smaller set of positions from $\phi$, called *obvious positions*, is still preventing the explosion and given by the rules:

(i) $p$ is an obvious position if $\phi|_p$ is an equivalence and there is a position $q < p$ such that $\phi|_q$ is either an equivalence or disjunctive in $\phi$ or

(ii) $pq$ is an obvious position if $\phi|_{pq}$ is a conjunctive formula in $\phi$, $\phi|_p$ is a disjunctive formula in $\phi$ and for all positions $r$ with $p < r < pq$ the formula $\phi|_r$ is not a conjunctive formula.

A formula $\phi|_p$ is conjunctive in $\phi$ if $\phi|_p$ is a conjunction and $\text{pol}(\phi, p) \in \{0, 1\}$ or $\phi|_p$ is a disjunction or implication and $\text{pol}(\phi, p) \in \{0, -1\}$.

Analogously, a formula $\phi|_p$ is disjunctive in $\phi$ if $\phi|_p$ is a disjunction or implication and $\text{pol}(\phi, p) \in \{0, 1\}$ or $\phi|_p$ is a conjunction and $\text{pol}(\phi, p) \in \{0, -1\}$.
Polarity Dependent Equivalence Elimination

ElimEquiv1  \[ \chi[(\phi \leftrightarrow \psi)]_p \Rightarrow_{\text{ACNF}} \chi[(\phi \to \psi) \land (\psi \to \phi)]_p \]
provided \( \text{pol}(\chi, p) \in \{0, 1\} \)

ElimEquiv2  \[ \chi[(\phi \leftrightarrow \psi)]_p \Rightarrow_{\text{ACNF}} \chi[(\phi \land \psi) \lor (\neg \phi \land \neg \psi)]_p \]
provided \( \text{pol}(\chi, p) = -1 \)
Extra $\top, \bot$ Elimination Rules

\begin{align*}
\text{ElimTB7} & \quad \chi[\phi \rightarrow \bot]_p \Rightarrow \text{ACNF} \quad \chi[\neg \phi]_p \\
\text{ElimTB8} & \quad \chi[\bot \rightarrow \phi]_p \Rightarrow \text{ACNF} \quad \chi[\top]_p \\
\text{ElimTB9} & \quad \chi[\phi \rightarrow \top]_p \Rightarrow \text{ACNF} \quad \chi[\top]_p \\
\text{ElimTB10} & \quad \chi[\top \rightarrow \phi]_p \Rightarrow \text{ACNF} \quad \chi[\phi]_p \\
\text{ElimTB11} & \quad \chi[\phi \leftrightarrow \bot]_p \Rightarrow \text{ACNF} \quad \chi[\neg \phi]_p \\
\text{ElimTB12} & \quad \chi[\phi \leftrightarrow \top]_p \Rightarrow \text{ACNF} \quad \chi[\phi]_p
\end{align*}

where the two rules ElimTB11, ElimTB12 for equivalences are applied with respect to commutativity of $\leftrightarrow$. 
Advanced CNF Algorithm

1. **Algorithm:** $acnf(\phi)$

   **Input:** A formula $\phi$.

   **Output:** A formula $\psi$ in CNF satisfiability preserving to $\phi$.

2. while rule $(ElimTB1(\phi), \ldots, ElimTB12(\phi))$ do ;
3.   SimpleRenaming($\phi$) on obvious positions;
4. while rule $(ElimEquiv1(\phi), ElimEquiv2(\phi))$ do ;
5.   while rule $(ElimImp(\phi))$ do ;
6. while rule $(PushNeg1(\phi), \ldots, PushNeg3(\phi))$ do ;
7. while rule $(PushDisj(\phi))$ do ;
8. return $\phi$ ;
Propositional Resolution

The propositional resolution calculus operates on a set of clauses and tests unsatisfiability.

Recall that for clauses I switch between the notation as a disjunction, e.g., $P \lor Q \lor P \lor \neg R$, and the multiset notation, e.g., \{P, Q, P, \neg R\}. This makes no difference as we consider $\lor$ in the context of clauses always modulo AC. Note that $\bot$, the empty disjunction, corresponds to $\emptyset$, the empty multiset. Clauses are typically denoted by letters $C, D$, possibly with subscript.
Resolution Inference Rules

Resolution  \( (N \cup \{C_1 \lor P, C_2 \lor \neg P\}) \Rightarrow_{\text{RES}} (N \cup \{C_1 \lor P, C_2 \lor \neg P\} \cup \{C_1 \lor C_2\}) \)

Factoring  \( (N \cup \{C \lor L \lor L\}) \Rightarrow_{\text{RES}} (N \cup \{C \lor L \lor L\} \cup \{C \lor L\}) \)
2.6.1 Theorem (Soundness & Completeness)

The resolution calculus is sound and complete:

\( N \) is unsatisfiable iff \( N \Rightarrow^{*}_{\text{RES}} N' \) and \( \bot \in N' \) for some \( N' \)
Resolution Reduction Rules

**Subsumption**

\[ (N \cup \{C_1, C_2\}) \Rightarrow_{\text{RES}} (N \cup \{C_1\}) \]

provided \( C_1 \subset C_2 \)

**Tautology Deletion**

\[ (N \cup \{C \lor P \lor \neg P\}) \Rightarrow_{\text{RES}} (N) \]

**Condensation**

\[ (N \cup \{C_1 \lor L \lor L\}) \Rightarrow_{\text{RES}} (N \cup \{C_1 \lor L\}) \]

**Subsumption Resolution**

\[ (N \cup \{C_1 \lor L, C_2 \lor \text{comp}(L)\}) \Rightarrow_{\text{RES}} (N \cup \{C_1 \lor L, C_2\}) \]

where \( C_1 \subseteq C_2 \)
2.6.6 Theorem (Resolution Termination)

If reduction rules are preferred over inference rules and no inference rule is applied twice to the same clause(s), then $\Rightarrow^+_\text{RES}$ is well-founded.
The Overall Picture

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>System + Problem</td>
</tr>
<tr>
<td>System</td>
</tr>
<tr>
<td>Algorithm + Implementation</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Calculus + Strategy</td>
</tr>
<tr>
<td>Calculus</td>
</tr>
<tr>
<td>Logic + States + Rules</td>
</tr>
<tr>
<td>Logic</td>
</tr>
<tr>
<td>Syntax + Semantics</td>
</tr>
</tbody>
</table>