and $C_2 = D \vee \neg L$, giving $R = C \vee D$. Now, L is either true or false. Suppose L is true. Then clearly C_1 is true, but since $\neg L$ is false and C_2 is true (by assumption), it must be that D, and hence R must be true. It is easy to see that we similarly arrive to the same conclusion about R even if L was false. $\square$

So, the theorems obtained by applying resolution to a set of axioms are all logical consequences of the axioms. In general, we will denote a clause C derived from a set of clauses Σ using resolution by $\Sigma \vdash_R C$. This means that there is a finite sequences of clauses $R_1, \ldots, R_k = C$ such that each C_i (where C_i is a clause being resolved upon in the i^{th} resolution step) is either in Σ or is a resolvent of a pair of clauses already derived (that is, from $\{R_1, \ldots, R_{i-1}\}$) . Now, although it is the case that if $\Sigma \vdash_R \alpha$ then $\Sigma \models \alpha$, the reverse does not hold. For example, a moment's thought should convince you that:

$$\{\textsf{Fred is an ape, Fred is human }\} \models \textsf{Fred is an ape} \leftarrow \textsf{Fred is human}$$

However, using resolution, there is no way of deriving Fred is an ape $\leftarrow$ Fred is human from Fred is an ape and Fred is human. As an inference rule, resolution is thus incomplete. However, it does have an extremely useful property known as *refutation completeness*. This is that if a formula Σ is inconsistent, then the empty clause $\square$ will be eventually derivable by resolution.

Thus, since Fred is an ape $\leftarrow$ Fred is human is a logical consequence of Fred is an ape and Fred is human, then the formula:

$$\Sigma : \{\textsf{Fred is an ape, Fred is human}, \neg(\textsf{Fred is an ape} \leftarrow \textsf{Fred is human})\}$$

must be inconsistent. This can be verified using resolution. First, the clausal form of (Fred is an ape $\leftarrow$ Fred is human) is (Fred is an ape $\vee \neg$ Fred is human). Using De Morgan's Law on this clausal form, we can see that $\neg$(Fred is an ape $\leftarrow$ Fred is human) is equivalent to $\neg$ Fred is an ape $\wedge$ Fred is human. We can now rewrite Σ :

$$\Sigma' : \{\textsf{Fred is an ape, Fred is human}, \neg\textsf{Fred is an ape}\}$$

Resolution of the pair Fred is an ape, $\neg$Fred is an ape would immediately result in the empty clause $\square$. The general steps in a refutation proof procedure using resolution are therefore:

- Let S be a set of clauses and α be a propositional formula. Let $C = S \cup \{\neg\alpha\}$.
- Repeatedly do the following:
    1. Select a pair of clauses C_1 and C_2 from C that can be resolved on some proposition P.
    2. Resolve C_1 and C_2 to give R.
    3. If $R = \square$ then stop. Otherwise, if R contains both a proposition Q and its negation $\neg Q$ then discard R. Otherwise add R to C.

In general, we know that any formula F can be converted to a conjunction of clauses. We can distinguish between the following sets. $Res^0(F)$, which is simply the set of clauses in F. $Res^n(F)$, for $n > 0$, which is the clauses containing all clauses in $Res^{n-1}(F)$ and all clauses obtained by resolving a pair of clauses from $Res^{n-1}(F)$. Since there are only a finite number of propositional symbols in F and a finite number of clauses in its CNF, we can see that there will only be a finite number of clauses that can be obtained using resolution. That is, there is some m such that $Res^m(F) = Res^{m-1}(F)$. Let us call this final set consisting of all the original clauses and all possible resolvents $Res^*(F)$. Then, the property of refutation-completeness for resolution can be stated more formally as follows:

Theorem 7 *For some formula F, $\Box \in Res^*(F)$ if and only if F is unsatisfiable.*

Though the resolution rule by itself is not (affirmation) complete for clauses in general, this property states that it is complete with respect to unsatisfiable sets of clauses. The complete proof of this will be provided on page 33. To get you started however, we show that if $\Box \in Res^*(F)$ then F is unsatisfiable. We can assume that $\Box \notin Res^0(F)$, since $\Box$ is not a disjunction of literals. Therefore there must be some k for which $\Box \notin Res^k(F)$ and $\Box \in Res^{k+1}(F)$. This can only mean that both L and $\neg L$ are in $Res^k(F)$. That is L and $\neg L$ are obtained from F by resolution. By the property of soundness of resolution, this means that $F \models (L \wedge \neg L)$. That is, F is unsatisfiable.

There are also other proof processes that are refutation-complete. Examples of such processes are the Davis-Putnam Procedure[13], Tableaux Procedure, *etc.* In the worst case, the resolution search procedure can take exponential time. This, however, very probably holds for all other proof procedures. For CNF formulae in propositional logic, a type of resolution process called the Davis-Putnam Procedure (backtracking over all truth values) is probably (in practice) the fastest refutation-complete process.

The Subsumption Theorem

A property related to logical implication is that of subsumption. A propositional clause C *subsumes* a propositional clause D if $C \subseteq D$. What does this mean? It just means that every literal in C appears in D. Here are a pair of clauses C and D such that C subsumes D:

$$C : \textsf{Fred is an ape}$$

$$D : \textsf{Fred is an ape} \leftarrow \textsf{Fred is human}$$

In general, it should be easy to see that if C and D are clauses such that $C \subseteq D$, then $C \models D$. In fact, for propositional logic, it is also the case that if $C \models D$ then C subsumes D (we see why this is so shortly).

The notion of subsumption acts as the basis for an important result linking resolution and logical implication, called the *subsumption theorem*:

[13] It can be proved that the Davis-Putnam procedure is sound as well as complete.