CHAPTER 5

Propositional Resolution

5.1 Introduction

Propositional Resolution is a powerful rule of inference for Propositional Logic. Using Propositional Resolution (without axiom schemata or other rules of inference), it is possible to build a theorem prover that is sound and complete for all of Propositional Logic. What's more, the search space using Propositional Resolution is much smaller than for standard Propositional Logic.

This chapter is devoted entirely to Propositional Resolution. We start with a look at clausal form, a variation of the language of Propositional Logic. We then examine the resolution rule itself. We close with some examples.

5.2 Clausal Form

Propositional Resolution works only on expressions in *clausal form*. Before the rule can be applied, the premises and conclusions must be converted to this form. Fortunately, as we shall see, there is a simple procedure for making this conversion.

A *literal* is either an atomic sentence or a negation of an atomic sentence. For example, if p is a logical constant, the following sentences are both literals.

$$p$$
$$\neg p$$

A *clausal sentence* is either a literal or a disjunction of literals. If p and q are logical constants, then the following are clausal sentences.

$$p$$
$$\neg p$$
$$\neg p \vee q$$

A *clause* is the set of literals in a clausal sentence. For example, the following sets are the clauses corresponding to the clausal sentences above.

As an example, consider the job of converting the sentence $(g \wedge (r \Rightarrow f))$ to clausal form. The conversion process is shown below.

$$
\begin{array}{ll}
 & g \wedge (r \Rightarrow f) \\
\mathrm{I} & g \wedge (\neg r \vee f) \\
\mathrm{N} & g \wedge (\neg r \vee f) \\
\mathrm{D} & g \wedge (\neg r \vee f) \\
\mathrm{O} & \{g\} \\
 & \{\neg r, f\}
\end{array}
$$

As a slightly more complicated case, consider the following conversion. We start with the same sentence except that, in this case, it is negated.

$$
\begin{array}{ll}
 & \neg(g \wedge (r \Rightarrow f)) \\
\mathrm{I} & \neg(g \wedge (\neg r \vee f)) \\
\mathrm{N} & \neg g \vee \neg(\neg r \vee f) \\
 & \neg g \vee (\neg\neg r \wedge \neg f) \\
 & \neg g \vee (r \wedge \neg f) \\
\mathrm{D} & (\neg g \vee r) \wedge (\neg g \vee \neg f) \\
\mathrm{O} & \{\neg g, r\} \\
 & \{\neg g, \neg f\}
\end{array}
$$

Note that, even though the sentences in these two examples are similar to start with (disagreeing on just one $\neg$ operator), the results are quite different.

5.3 Resolution Principle

The idea of Propositional Resolution is simple. Suppose we have the clause $\{p, q\}$. In other words, we know that p is true or q is true. Suppose we also have the clause $\{\neg q, r\}$. In other words, we know that q is false or r is true. One clause contains q, and the other contains $\neg q$. If q is false, then by the first clause p must be true. If q is true, then, by the second clause, r must be true.
Since $q$must be either true or false, then it must be the case that either p is true or r is true. So we should be able to derive the clause $\{p, r\}$.

This intuition is the basis for the rule of inference shown below. Given a clause containing a literal χ and another clause containing the literal $\neg\chi$, we can infer the clause consisting of all the literals of

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

Note that the empty set {} is also a clause. It is equivalent to an empty disjunction and, therefore, is unsatisfiable. As we shall see, it is a particularly important special case.

As mentioned earlier, there is a simple procedure for converting an arbitrary set of Propositional Logic sentences to an equivalent set of clauses.The conversion rules are summarized below and should be applied in order.

1. Implications (I):

$$\begin{array}{lll} \varphi \Rightarrow \psi & \rightarrow & \neg\varphi \vee \psi \\ \varphi \Leftarrow \psi & \rightarrow & \varphi \vee \neg\psi \\ \varphi \Leftrightarrow \psi & \rightarrow & (\neg\varphi \vee \psi) \wedge (\varphi \vee \neg\psi) \end{array}$$

2. Negations (N):

$$\begin{array}{lll} \neg\neg\varphi & \rightarrow & \varphi \\ \neg(\varphi \wedge \psi) & \rightarrow & \neg\varphi \vee \neg\psi \\ \neg(\varphi \vee \psi) & \rightarrow & \neg\varphi \wedge \neg\psi \end{array}$$

3. Distribution (D):

$$\begin{array}{lll} \varphi \vee (\psi \wedge \chi) & \rightarrow & (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\ (\varphi \wedge \psi) \vee \chi & \rightarrow & (\varphi \vee \chi) \wedge (\psi \vee \chi) \\ \varphi \vee (\varphi_1 \vee \ldots \vee \varphi_n) & \rightarrow & \varphi \vee \varphi_1 \vee \ldots \vee \varphi_n \\ (\varphi_1 \vee \ldots \vee \varphi_n) \vee \varphi & \rightarrow & \varphi_1 \vee \ldots \vee \varphi_n \vee \varphi \\ \varphi \wedge (\varphi_1 \wedge \ldots \wedge \varphi_n) & \rightarrow & \varphi \wedge \varphi_1 \wedge \ldots \wedge \varphi_n \\ (\varphi_1 \wedge \ldots \wedge \varphi_n) \wedge \varphi & \rightarrow & \varphi_1 \wedge \ldots \wedge \varphi_n \wedge \varphi \end{array}$$

4. Operators (O):

$$\begin{array}{lll} \varphi_1 \vee \ldots \vee \varphi & \rightarrow & \{\varphi_1, \ldots, \varphi_n\} \\ \varphi_1 \wedge \ldots \wedge \varphi_n & \rightarrow & \{\varphi_1\}, \ldots, \{\varphi_n\} \end{array}$$

both clauses without the complementary pair. This rule of inference is called *Propositional Resolution* or the *Resolution Principle*.

$$\frac{\begin{array}{l}\{\varphi_1, \dots, \chi, \dots, \varphi_m\}\\ \{\psi_1, \dots, \neg\chi, \dots, \psi_n\}\end{array}}{\{\varphi_1, \dots, \varphi_m, \psi_1, \dots, \psi_n\}}$$

The case we just discussed is an example. If we have the clause $\{p, q\}$ and we also have the clause $\{\neg q, r\}$, then we can derive the clause $\{p, r\}$ in a single step.

$$\frac{\begin{array}{l}\{p, q\}\\ \{\neg q, r\}\end{array}}{\{p, r\}}$$

Note that, since clauses are sets, there cannot be two occurrences of any literal in a clause. Therefore, in drawing a conclusion from two clauses that share a literal, we merge the two occurrences into one, as in the following example.

$$\frac{\begin{array}{l}\{\neg p, q\}\\ \{p, q\}\end{array}}{\{q\}}$$

If either of the clauses is a singleton set, we see that the number of literals in the result is less than the number of literals in the other clause. For example, from the clause $\{p, q, r\}$ and the singleton clause $\{\neg p\}$, we can derive the shorter clause $\{q, r\}$.

$$\frac{\begin{array}{l}\{p, q, r\}\\ \{\neg p\}\end{array}}{\{q, r\}}$$

Resolving two singleton clauses leads to the *empty clause*; i.e. the clause consisting of no literals at all, as shown below. The derivation of the empty clause means that the database contains a contradiction.

$$\frac{\{p\} \quad \{\neg p\}}{\{\}}$$

If two clauses resolve, they may have more than one resolvent because there can be more than one way in which to choose the resolvents. Consider the following deductions.

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

Note, however, when two clauses have multiple pairs of complementary literals, only *one pair* of literals may be resolved at a time. For example, the following is *not* a legal application of Propositional Resolution.

$$\frac{\{p, q\} \quad \{\neg p, \neg q\}}{\{\}} \quad \text{Wrong!}$$

If we were to allow this to go through, we would be saying that these two clauses are inconsistent. However, it is perfectly possible for $(p \vee q)$ to be true and $(\neg p \vee \neg q)$ to be true at the same time. For example, we just let p be true and q be false, and we have satisfied both clauses.

It is noteworthy that resolution subsumes many of our other rules of inference. Consider, for example, Implication Elimination, shown below on the left. If we have $(p \Rightarrow q)$ and we have p, then we can deduce q. The clausal form of the premises and conclusion are shown below on the right. The implication $(p \Rightarrow q)$ corresponds to the clause $\{\neg p, q\}$, and p corresponds to the singleton clause $\{p\}$. We have two clauses with a complementary literal, and so we cancel the complementary literals and derive the clause $\{q\}$, which is the clausal form of q.

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

q	$\{q\}$

As another example, recall the example of formal reasoning introduced in Chapter 1. We said that, whenever we have two rules in which the left hand side of one contains a proposition constant that occurs on the right hand side of the other, then we can cancel those constants and deduce a new rule by combining the remaining constants on the left hand sides of both rules and the remaining constants on the right hand sides of both rules. As it turns out, this is just Propositional Resolution.

Recall that we illustrated this rule with the deduction shown below on the left. Given $(m \Rightarrow p \vee q)$ and $(p \Rightarrow q)$, we deduce $(m \Rightarrow q)$. On the right, we have the clausal form of the sentences on the left. In place of the first sentence, we have the clause $\{\neg m, p, q\}$; and, in place of the second sentence, we have $\{\neg p, q\}$. Using resolution, we can deduce $\{\neg m, q\}$, which is the clausal form of the sentence we derived on the left.

$m \Rightarrow p \vee q$	$\{\neg m, p, q\}$
$p \Rightarrow q$	$\{\neg p, q\}$
$m \Rightarrow q$	$\{\neg m, q\}$

5.4 Resolution Reasoning

Reasoning with the Resolution Principle is analogous to reasoning with other rules of inference. We start with premises; we apply the Resolution Principle to those premises; we apply the rule to the results of those applications; and so forth until we get to our desired conclusion or we run out of things to do.

Formally, we define a *resolution derivation* of a conclusion from a set of premises to be a finite sequence of clauses terminating in the conclusion in which each clause is either a premise or the result of applying the Resolution Principle to earlier members of the sequence.

Note that our definition of resolution derivation is analogous to our definition of linear proof. However, in this case, we do not use the word *proof*, because we reserve that word for a slightly different concept, which is discussed below.

In many cases, it is possible to find resolution derivations of conclusions from premises. Suppose, for example, we are given the clauses $\{\neg p, r\}$ and $\{\neg q, r\}$ and $\{p, q\}$. Then we can derive the conclusion $\{r\}$ as shown below.

1. $\{\neg p, r\}$ Premise

2.	$\{\neg q, r\}$	Premise
3.	$\{p, q\}$	Premise
4.	$\{q, r\}$	1, 3
5.	$\{r\}$	2, 4

It is noteworthy that the resolution is not *generatively complete*, i.e. it is not possible to find resolution derivations for all clauses that are logically entailed by a set of premise clauses.

For example, given the clause $\{p\}$ and the clause $\{q\}$, there is no resolution derivation of $\{p, q\}$, despite the fact that it is logically entailed by the premises in this case.

As another example, consider that valid clauses (such as $\{p, \neg p\}$) are always true, and so they are logically entailed by any set of premises, including the empty set. However, Propositional Resolution requires some premises to have any effect. Given an empty set of premises, we would not be able to derive any conclusions, including these valid clauses.

On the other hand, we can be sure of one thing. If a set Δ of clauses is unsatisfiable, then there is guaranteed to be a resolution derivation of the empty clause from Δ. More generally, if a set Δ of Propositional Logic sentences is unsatisfiable, then there is guaranteed to be a resolution derivation of the empty clause from the clausal form of Δ.

As an example, consider the clauses $\{p, q\}$, $\{p, \neg q\}$, $\{\neg p, q\}$, and $\{\neg p, \neg q\}$. There is no truth assignment that satisfies all four of these clauses. Consequently, starting with these clauses, we should be able to derive the empty clause; and we can. A resolution derivation is shown below.

1.	$\{p, q\}$	Premise	$p \vee q$
2.	$\{p, \neg q\}$	Premise	$q \rightarrow p$
3.	$\{\neg p, q\}$	Premise	$p \rightarrow q$
4.	$\{\neg p, \neg q\}$	Premise	$\neg p \vee \neg q$
5.	$\{p\}$	1, 2	
6.	$\{\neg p\}$	3, 4	
7.	$\{\}$	5, 6	

The good news is that we can use the relationship between unsatisfiability and logical entailment to produce a method for determining logical entailment as well. Recall that the Unsatisfiability Theorem introduced in Chapter 3 tells that a set Δ of sentences logically entails a sentence φ if and only if the set of sentences $\Delta \cup \{\neg\varphi\}$ is inconsistent. So, to determine logical entailment, all we need to do is to negate our goal, add it to our premises, and use Resolution to determine whether the resulting set is unsatisfiable.

Let's capture this idea with some definitions. A *resolution proof* of a sentence φ from a set Δ of sentences is a resolution derivation of the empty clause from the clausal form of $\Delta \cup \{\neg\varphi\}$. A sentence φ is *provable* from a set of sentences Δ by Propositional Resolution (written $\Delta \vdash \varphi$) if and only if there is a resolution proof of φ from Δ.

As an example of a resolution proof, consider one of the problems we saw earlier. We have three premises - p, $(p \Rightarrow q)$, and $(p \Rightarrow q) \Rightarrow (q \Rightarrow r)$. Our job is to prove r. A resolution proof is shown below. The first two clauses in the proof correspond to the first two premises of the problem. The third and fourth clauses in the proof correspond to the third premise. The fifth clause comes from the negation of the goal. Resolving the first clause with the second, we get the clause q, shown on line 6. Resolving this with the fourth clause gives us r. And resolving this with the clause on line 5 gives us the empty clause.

resolution proofs

1. $\{p\}$	Premise
2. $\{\neg p, q\}$	Premise
3. $\{p, \neg q, r\}$	Premise
4. $\{\neg q, r\}$	Premise
5. $\{\neg r\}$	Premise
6. $\{q\}$	1, 2
7. $\{r\}$	4, 6
8. $\{\}$	5, 7

Here is another example, this time illustrating the way in which we can use Resolution to prove valid sentences. Let's say that we have no premises at all and we want to prove $(p \Rightarrow (q \Rightarrow p))$, an instance of the Implication Creation axiom schema.

The first step is to negate this sentence and convert to clausal form. A trace of the conversion process is shown below. Note that we end up with three clauses.

$$
\begin{array}{ll}
 & \neg(p \Rightarrow (q \Rightarrow p)) \\
\text{I} & \neg(\neg p \vee (\neg q \vee p)) \\
\text{N} & \neg\neg p \wedge \neg(\neg q \vee p) \\
 & p \wedge (\neg\neg q \wedge \neg p) \\
\text{D} & p \wedge q \wedge \neg p \\
\text{O} & \{p\}
\end{array}
$$

Δ. Propositional Resolution can be used in a proof procedure that always terminates without losing completeness.

Exercises

Exercise 5.1: Convert the following sentences to clausal form.

(*a*) $p \wedge q \Rightarrow r \vee s$
(*b*) $p \vee q \Rightarrow r \vee s$
(*c*) $\neg(p \vee q \vee r)$
(*d*) $\neg(p \wedge q \wedge r)$
(*e*) $p \wedge q \Leftrightarrow r$

Exercise 5.2: What are the results of applying Propositional Resolution to the following pairs of clauses.

(*a*) $\{p, q, \neg r\}$ and $\{r, s\}$
(*b*) $\{p, q, r\}$ and $\{r, \neg s, \neg t\}$
(*c*) $\{q, \neg q\}$ and $\{q, \neg q\}$
(*d*) $\{\neg p, q, r\}$ and $\{p, \neg q, \neg r\}$

Exercise 5.3: Use Propositional Resolution to show that the clauses $\{p, q\}$, $\{\neg p, r\}$, $\{\neg p, \neg r\}$, $\{p, \neg q\}$ are not simultaneously satisfiable.

Exercise 5.4: Given the premises $(p \Rightarrow q)$ and $(r \Rightarrow s)$, use Propositional Resolution to prove the conclusion $(p \vee r \Rightarrow q \vee s)$.

Relational Logic Rules of Inference

$$\frac{\phi}{\forall v.\phi}$$

Universal Introduction

UI

where v does not occur free in both ϕ and an active assumption

Can be used like this

likes(jane, y)

$\forall y.$likes(jane, y)

this makes the universal quantification explicit

Also consider

likes(jane, jill)

$\forall x.$likes(jane, jill)

$\forall z.$likes(jane, jill)

$\forall x.$likes(jane, y)

cannot use when the variable is in the sentence and appears free in an active subproof

Universal Introduction

$$\frac{\phi}{\forall_v . \phi}$$

Here it is noteworthy that ϕ might or might not be $\phi(y)$. that is ϕ may or may not involve y.

preys(wolves, y) means $\forall y . p(w,y)$ (makes universal quantifier explicit)

but this could also be used when y does not exist in the sentence.

preys(~~wo~~ fox, rabbits) or $p(f,r)$

$\forall x . p(f,r)$

$\forall y . p(f,r)$ etc

1C

1. $\forall y . \forall x . (p(y,x) \to C(y))$ premise

2. $\forall y . (p(y,r) \to C(y))$ UE(1)

3. $p(y,r) \to C(y)$ UE(2)

4. $p(y,r)$ Assume

5. $C(y)$ IE(3)

6. $\forall y . C(y)$ UI(5)

This is the illegal step you cannot universally quantify a variable that is in an active assumption

7. $\underbrace{p(y,r) \to \forall y . C(y)}$

This ~~statement~~ impliciation says if an animal y preys on rabbits then all animals are carnivors.

(1d)

1. $\forall y. \forall x. (p(y,x) \rightarrow c(y))$ premise

2. $\forall y. \, p(y, \text{rabbit}) \rightarrow c(y)$ UE(1)

3. $p(y,r) \rightarrow c(y)$ UE(2)

4. | $p(y,r)$ Assumption

5. | $c(y)$ IE(3,4)

6 | $\forall y. c(y)$ UI(5) NOT ALLOWED MUY MALO

7. $p(y,r) \rightarrow \forall y. c(y)$ II(4,6)

~~8. $\exists x. (p(y,x) \rightarrow \forall y \, c(y))$~~

8. $\exists x. (p(y,x) \rightarrow \forall y \, c(y))$ EI(7)

9. $\forall y. \exists x. (p(y,x) \rightarrow \forall y \, c(y))$

y preys on x then All animals ar carnivore

Universal Elimination UE

$$\frac{\forall v .\ \phi(v)}{\phi(\tau)}$$

example

from $\forall y .$ likes (jane, y)

we get likes (jane, jill)

Also $\forall x .$ like (jane, x)

becomes like (jane, x)

like (jane, y)

Avoid conflicts with other quantified variables within the sentence.

$\forall x . \exists y .$ likes (x, y) Everyone ~~hates~~ likes someone

$\exists y .$ likes (jane, y) Jane likes someone

But we could <u>not</u> conclude

likes (y, y) as in

Someone likes themself

τ is free for v in a sentence ϕ IFF no free occurence of v occurs within the scope of a quantifier of some variable in τ

Universal Elimination

$$\frac{\forall_v \cdot \Phi(v)}{\Phi(\tau)}$$

For example

$$\frac{\forall_x \cdot p(w,x)}{p(w,d)}$$

means For all x, wolves prey on x or wolves prey on all animals.

we can conclude wolves prey on deer. since wolves prey on all animals.

put another way. $\forall x.\, p(w,x) = p(w,d) \wedge p(w,r) \wedge p(w,w) \wedge p(w,f)$

$p(w,d) \wedge p(w,r) \wedge p(w,w) \wedge p(w,f)$

using And elimination we could say

$p(w,d)$

So Universal Elimination is like And Elimination on Steroids.

NOTE Avoid conflicts with other quantified variables

$\forall x.\exists y.\, likes(x,y)$ Everyone likes someone

implies $\exists y.\, likes(jane,y)$ or $\exists y.\, likes(z,y)$

but <u>not</u> $\exists y\, likes(y,y)$ someone likes herself.

Existential Introduction EI

$$\frac{\phi(\tau)}{\exists v.\ \phi(v)}$$

Example

hates(jill, jane)

$\exists x.$ hates(jill, x)

OR

If – hates(jill, jill)

then $\exists x.$ hates(x, x)

or $\exists x.$ hates(x, jill)

or $\exists x.$ hates(jill, x)

or $\exists y. \exists x.$ hates(x, y)

Avoid ~~a~~ introducing variables already in sentence.

If $\exists x.$hates(jane, x) jane hates somebody

don't to this

$\exists x. \exists x.$ hates(x, x) someone hates themself.

Existential Introduction

(3a)

$$\frac{\phi(\tau)}{\exists v.\ \phi(v)}$$

For Example

$$p(f,r) \Rightarrow \exists y.\ p(f,y)$$

if we know foxes prey on rabbits that implies there is at least on y such that foxes prey on y.

$$p(f,r) \Rightarrow p(f,d) \vee p(f,r) \vee p(f,w) \vee p(f,f)$$

$$\exists y.\ p(f,y)$$

Existentially quantified = or statement

EI is like or introduction on steroids

Existential Elimination EE

$$\exists v . \phi(v)$$
$$\forall v . (\phi(v) \Rightarrow \psi)$$
$$\overline{\psi}$$

Analogous to OR Elimination

where v does not occur free in ψ

consider

$$\exists x . hates(jane, x)$$
$$\forall x . (hates(jane, x) \Rightarrow \neg nice(jane))$$
$$\overline{\neg nice(jane)} \quad EE$$

Existential Elimination

$$\frac{\begin{array}{l}\exists v.\ \phi(v) \\ \forall v.(\phi(v) \Rightarrow \psi)\end{array}}{\psi}$$

where v does <u>not</u> occur free in ψ

Consider example

$$\frac{\begin{array}{l}\exists x.\ p(f,x) \\ \forall x.(p(f,x) \rightarrow c(f))\end{array}}{c(f)}$$

$$\frac{\begin{array}{l}p(f,d) \vee p(f,r) \vee p(f,w) \vee p(f,f) \\ p(f,d) \rightarrow c(f) \\ p(f,r) \rightarrow c(f) \\ p(f,w) \rightarrow c(f) \\ p(f,f) \rightarrow c(f)\end{array}}{c(f)}$$

Existential Elimination is like OR Elimination on Steroids

CANNOT DO THIS

$$\frac{\begin{array}{l}\exists x.\ p(x) \\ \forall x.(p(x) \rightarrow r(x))\end{array}}{r(x)}$$

Logical Fallacy Due to Incorrect use of EE

4b

1. $c(f)$ premise

we start with premise that foxes are carnivores

2. $\exists x. c(x)$ EI(1)
3. | $c(x)$ Assume
4. | $c(x)$ Reiteration(3)
5. $c(x) \to c(x)$ II(3,4)
6. $\forall x.(c(x) \to c(x))$ UI(5)
7. $c(x)$ EE(2,6)

This is the illegal step.

8. $\forall x. c(x)$ UI(7)
9. $c(d)$ UE(8)

here we conclude that deer are carnivores. this is a fallacious conclusion because

EE says this

$$\frac{\exists x.\, c(x) \qquad \forall x.(c(x) \to \psi)}{\psi}$$

In the proof above we said

$$\frac{\exists x.\, c(x) \qquad \forall x.(c(x) \to c(x))}{c(x)}$$

But the problem is that ψ cannot involve the quantified variable x.
In our proof $\psi = c(x)$ which does involve the quantified variable.

~~problem~~ 2 4c

1. $\forall x. (p(x) \Rightarrow q(x))$

2. $\forall y. (r(y) \Leftrightarrow q(y))$

3. $p(a)$

goal

$r(b)$

---

4. $p(x) \rightarrow q(x)$ UE (1)

5. $r(x) \leftrightarrow q(x)$ UE (2)

6. $q(x) \rightarrow r(x)$ BE (5)

7. | $p(x)$ assumption

8. | $q(x)$ IE (4,7)

9. | $r(x)$ IE (6,8)

10. $p(x) \rightarrow r(x)$ II (7,9)

11. $\forall x (p(x) \rightarrow r(x))$ UI (10)

12. $\exists x. p(x)$ EI (3)

13. $r(x)$ EE (11,12) NO, NOT ALLOWED

---

14. $\forall x. (r(x))$ UI (13)

15. $r(b)$ UE (14)

cannot have this

$$\frac{\exists x.\ p(x) \qquad \forall x. (p(x) \rightarrow \psi)}{\psi}$$

ψ cannot be a function of x

Domain Closure

(5)

For any language with finite object constants

$\sigma_1, \sigma_2, \ldots, \sigma_n$

Then

$$\frac{\begin{array}{l}\phi(\sigma_1)\\ \phi(\sigma_2)\\ \vdots\\ \phi(\sigma_n)\end{array}}{\forall x.\, \phi(x)}$$

for example if we have 4 object constants in our world a, b, c, d.

$$\frac{\begin{array}{l}\phi(a)\\ \phi(b)\\ \phi(c)\\ \phi(d)\end{array}}{\forall v.\, \phi(v)}$$

If we know $\phi(a)$, $\phi(b)$, $\phi(c)$, $\phi(d)$ then we conclude

$$\forall v.\, \phi(v)$$

Example

premises

everybody loves somebody

everyone loves a lover

goal.

Jack Loves Jill

1 $\forall y. \exists z.\ loves(y, z)$

Everybody loves a lover

2 $\forall x. \forall y. \forall z. (\ loves(y, z) \Rightarrow loves(x, y))$ If a person is a lover then everyone loves him

3. $\exists z.\ loves(jill, z)$ UE(1)

If a person loves another person, then everyone loves him

4 $\forall y. \forall z. (\ loves(y, z) \Rightarrow loves(jack, y))$ UE(2)

loves(y, z) implies loves(x, y)

5 $\forall z(loves(jill, z) \Rightarrow loves(jack, jill))$ UE (4)

6. $loves(jack, jill)$ EE (3,5)

goal $c(f)$ (6a)

1. $p(f,r)$
2. $\forall y . \forall x . (p(y,x) \to c(y))$

---

3. $\exists x . p(f,x)$ EI(1)
4. $\forall x . (p(f,x) \to c(f))$ UE(2)
5. $c(f)$ EE(3,4)

Another way

---

goal $c(f)$

1. $p(f,r)$
2. $\forall y . \forall x . (p(y,x) \to c(y))$

---

3. $\forall x . (p(f,x) \to c(f))$ UE(2)
4. $p(f,r) \to c(f)$ UE(3)
5. $c(f)$ IE(1,4)

7

1. $\forall y . \exists z . \text{loves}(y,z)$

2. $\forall x . \forall y . \forall z . (\text{loves}(y,z) \Rightarrow \text{loves}(x,y))$

goal
$\forall x . \forall y . \text{loves}(x,y)$

---

3. $\exists z . \text{loves}(y,z)$ UE(1)
4. $\forall y \forall z (\text{loves}(y,z) \Rightarrow \text{loves}(x,y))$ UE(2)
5. $\forall z . (\text{loves}(y,z) \Rightarrow \text{loves}(x,y))$ U(E)4
6. $\text{loves}(x,y)$ EE(3,5)
7. $\forall x . \text{loves}(x,y)$ UI(6)
8. $\forall x . \forall y . \text{loves}(x,y)$ UI(7)

φ_6

1. $\forall x . \forall y . (h(x) \land d(y) \to f(x,y))$ premise — Horses are faster than dogs.

2. $\exists y . (g(y) \land \forall z (r(z) \to f(y,z)))$ premise — there is a greyhord that is faster than all rabils.

3. $\forall y . (g(y) \to d(y))$ premise — All greyhords are dogs.

4. $\forall x . \forall y . \forall z . (f(x,y) \land f(y,z) \to f(x,z))$ transitive property.

5. $h(harry)$

6. $r(ralph)$

goal $f(harry, ralph)$

---

7. $g(y) \land \forall z . (r(z) \to f(y,z))$ Assumption

8. $g(y)$ AE(7)

9. $\forall z . (r(z) \to f(y,z))$ AE(7)

10. $r(ralph) \to f(y, ralph)$ UE(9)

11. $f(y, ralph)$ IE (6,10)

12. $g(y) \to d(y)$ UE (3)

13. $d(y)$ IE (8,12)

14. ~~$\forall y . (h(x) \land d(y) \to f(x,y))$ UE (1)~~

15. ~~$h(x) \land d(y) \to f(x,y)$ UE (14)~~

16.

14 $\forall y. (h(harry) \wedge d(y) \to f(harry, y))$ UE(1)

15. $h(harry) \wedge d(y) \to f(harry, y)$ UE(14)

16 $h(harry) \wedge d(y)$ AI (5, 13)

17 $f(harry, y)$ IE (15, 16)

18 $\forall y. \forall z. (f(harry, y) \wedge f(y, z) \to f(harry, z))$ UE (4)

19 $\forall z. (f(harry, y) \wedge f(y, z) \to f(harry, z))$ UE (18)

20 $f(harry, y) \wedge f(y, ralph) \to f(harry, ralph)$ UE(19)

21 $f(harry, y) \wedge f(y, ralph)$ AI (17, 11)

22. $f(harry, ralph)$ IE (20, 21)

23. $g(y) \wedge \forall (r(z) \Rightarrow f(y, z)) \Rightarrow f(harry, ralph)$ II(7, 22)

24. $\forall y. (g(y) \wedge \forall z. (r(z) \Rightarrow f(y, z)) \Rightarrow f(harry, ralph))$

25 $f(harry, ralph)$ EE (2, 24)

15.

h

1. $\forall x.\forall y.(p(x,y) \to q(x))$ premise

2. $\exists y.\, p(x,y)$ Assume

3. $\forall y.(p(x,y) \to q(x))$ UE(1)

4. $q(x)$ EE(2,3) no y

5. $\exists y.\, p(x,y) \to q(x)$ II(2,4)

6. $\forall x.(\exists y.\, p(x,y) \to q(x))$ UI(5)

So, what does this say?

$\forall x.\forall y.(p(x,y) \to q(x))$ is same as $\forall x.(\exists y.\, p(x,y) \to q(x$ as long as no y

---

So, from our example.

$\forall x.\forall y.(p(x,y) \to c(x))$

this says for all animals x, y if x preys on y then x is a carnivore

$\forall x.(\exists y.\, p(x,y) \to c(x))$

This says for all animals x If there exists an animal y such that x preys on y then x is a carnivor

$\forall y.(\exists x.\, p(x,y) \to c(x))$

This says. for all animals y. if there exists an animal x that preys on y then all animals are carnivores

this is not correct Because $\exists x.$ comes from the $\forall x.$ that quantifies $c(x)$

object constants are a, b, c

Goal $\forall x. p(x)$

1. $p(a) \land p(b)$
2. $\exists x. p(x) \rightarrow \forall y. q(y)$
3. $\forall x. (q(x) \rightarrow p(x))$

---

4. ~~p(a~~ ~~$\exists x. p(x)$~~ ~~EI~~
4. $p(a)$ AE(1)
5. $\exists x. p(x)$ EI(4)
6. $\forall y. q(y)$ IE(2,5)
7. $q(y)$ UE(6)
8. $q(y) \rightarrow p(y)$ UE(3)
9. $p(y)$ IE(7,8)
10. $\forall x. p(x)$ UI(9)

1. $\forall x . (\exists y . p(x,y) \to q(x))$

---

2. $p(x,y)$ assume
3. $\exists y . P(x,y)$ EI(2)
4. $\exists y . P(x,y) \to q(x)$ UE(1)
5. $q(x)$ IE (3,4)
6. $p(x,y) \to q(x)$ II (2,5)
7. $\forall y . (p(x,y) \to q(x))$ UI (6)
8. $\forall x . \forall y . (p(x,y) \to q(x))$ UI (7)

$\forall x. (\exists y. p(x,y) \rightarrow q(x)))$

1. ~~$\forall x. \forall y. (p(x,y) \rightarrow q(x)$~~

Goal

$\forall x (\exists y\, p(x,y) \rightarrow q(x$

2.	$p(x,y)$	assume
3.	$\exists y. P(x,y)$	EI
4.		

problem 8

1. $\forall x . (\exists y . p(x,y) \Rightarrow q(x))$

goal
$\forall x . \forall y . (p(x,y) \to q(x))$

2 | $p(x,y)$ assumptin

3. | $\exists y . p(x,y)$ EI (2)

4 | $\exists y . p(x,y) \to q(x)$ ~~UI~~ E

5 | $q(x)$ IE

6. $p(x,y) \Rightarrow q(x)$

7. $\forall y . (p(x,y) \to q(x))$

8. $\forall x . \forall y . (p(x,y) \to q(x))$

1. $\forall x. \forall y. (p(x,y) \Rightarrow q(x))$

problem 7

goal $\forall x. (\exists y. p(x,y) \Rightarrow q(x))$

problem 7

1. Premise

1. $p(a) \rightarrow q(a)$

goal $f(a,b)$

problem 1

$r(a) \Leftrightarrow q(a)$

3 $\forall x((p(x) \rightarrow r(x)) \Rightarrow f(a,b))$

Goal $\forall x. \forall y. \text{loves}(x,y)$ Problem 5

1. $\forall y. \exists z. \text{loves}(y,z)$
2. $\forall x. \forall y. \forall z. (\text{loves}(y,z) \Rightarrow \text{loves}(x,y))$