Example using strategies 1 & 2:
$\{(A \wedge B) \rightarrow (C \leftrightarrow D), (A \wedge B) \leftrightarrow (F \vee G), G \wedge H\} \,/\therefore\, C \leftrightarrow D$

1

1	$(A \wedge B) \rightarrow (C \leftrightarrow D)$	P
2	$(A \wedge B) \leftrightarrow (F \vee G)$	P
3	$G \wedge H$	P
	⋮	
	$C \leftrightarrow D$	

2

Our first strategy hint tells us to try to extract our goal sentence from a more complex sentence in which it occurs as a whole. Here we can get the goal sentence from step 1 by →Elim if we have ‘$A \wedge B$’, so we aim for that.

1	$(A \wedge B) \rightarrow (C \leftrightarrow D)$	P
2	$(A \wedge B) \leftrightarrow (F \vee G)$	P
3	$G \wedge H$	P
	⋮	
	$A \wedge B$	
	$C \leftrightarrow D$	→ Elim: 1,?

3

The same hint tells us to look for a complex sentence in which ‘$A \wedge B$’ appears, from which we can pull out this sentence as a whole. We could get this from step 2 if we had the other side of the biconditional, ‘$F \vee G$’, as a whole, so now we aim for that.

1	$(A \wedge B) \rightarrow (C \leftrightarrow D)$	P
2	$(A \wedge B) \leftrightarrow (F \vee G)$	P
3	$G \wedge H$	P
	⋮	
	$F \vee G$	
	$A \wedge B$	↔Elim: 2,?
	$C \leftrightarrow D$	→Elim: 1,?

4

Our new goal, ‘$F \vee G$’, does not appear in its entirety as a part of any more complex sentence except in step 2. To get it from there, we would have to have ‘$A \wedge B$’ already, but the whole point of getting ‘$F \vee G$’ is that we don’t have ‘$A \wedge B$’ yet, and we need ‘$F \vee G$’ before ‘$A \wedge B$’ to help us get ‘$A \wedge B$’. So we won’t be able to get ‘$F \vee G$’ by pulling it out of line 2.

Our second strategy hint suggests building up our goal sentence. The main operator in ‘$F \vee G$’ is the ‘$\vee$’, so we ask whether one disjunct is obviously very easy to get from what we already have above. If so we, we will do that, then use $\vee$ Intro to get the disjunction we want. It’s easy to get ‘G’ from step 3, so we follow this strategy.

1	$(A \wedge B) \rightarrow (C \leftrightarrow D)$	P
2	$(A \wedge B) \leftrightarrow (F \vee G)$	P
3	$G \wedge H$	P
	G	∧Elim: 3
	$F \vee G$	∨ Intro
	$A \wedge B$	↔Elim: 2,?
	$C \leftrightarrow D$	→Elim: 1,?

5

All that remains to be done is numbering the rest of the lines, and using these line numbers to complete our justifications.

1	$(A \wedge B) \rightarrow (C \leftrightarrow D)$	P
2	$(A \wedge B) \leftrightarrow (F \vee G)$	P
3	$G \wedge H$	P
4	G	∧ Elim: 3
5	$F \vee G$	∨ Intro: 4
6	$A \wedge B$	↔Elim: 2,5
7	$C \leftrightarrow D$	→Elim: 1,6

1

```
1 | C ↔ D            P
2 | C ∧ ¬(D ∧ B)      P
  :
  | ¬B
```

STRATEGY:
{C ↔ D, C ∧ ¬(D ∧ B)} /∴ ¬B

2

```
1 | C ↔ D                P
2 | C ∧ ¬(D ∧ B)          P
  | | B
  : :
  : : ?
  : :
  : : ¬ ?
  : : ⊥                 ⊥ Intro
  | ¬B                  ¬ Intro
```

Notice that we don't have '¬B' in the premises, so we can't expect to get it directly using rules like ∧ Elim and ↔Elim. Thus we assume 'B' and try to show that it leads to a contradiction. We don't yet know what the contradiction will be, but we know we need some sentence and its negation.

3

We notice that we can get a negation, '¬(D ∧ B)', easily from 2 by ∧Elim., so we pick that as the negation to aim for. Then we try to get 'D ∧ B' as the other member of our contradictory pair of sentences. ⇒

```
1 | C ↔ D                   P
2 | C ∧ ¬(D ∧ B)             P
3 | | B
  : :
  | | D ∧ B
  | | ¬(D ∧ B)         ∧Elim: 2
  | | ⊥                 ⊥ Intro
  | ¬B                  ¬ Intro
```

4

When aiming for a conjunction, try to get each conjunct separately. We already have 'B' on line 3, so we just need 'D'.

```
1 | C ↔ D                 P
2 | C ∧ ¬(D ∧ B)           P
3 | | B
  : :
  | | D
  | | D ∧ B            ∧ Intro
  | | ¬(D ∧ B)        ∧ Elim: 2
  | | ⊥                ⊥ Intro
  | ¬B                 ¬ Intro
```

5

Notice that we can get 'C' from 2 by ∧ Elim. With 1, that will let us get 'D' by using the ↔ Elim rule.

```
1 | C ↔ D               P
2 | C ∧ ¬(D ∧ B)         P
3 | | B
  | | C             ∧ Elim: 2
  | | D               □ Elim
  | | D ∧ B          ∧ Intro
  | | ¬(D ∧ B)     ∧ Elim: 2
  | | ⊥              ⊥ Intro
  | ¬B               ¬ Intro
```

6

No other steps are needed. We fill in the line numbers and complete the justifications. ⇒

```
1 | C ↔ D                   P
2 | C ∧ ¬(D ∧ B)             P
3 | | B
4 | | C               ∧ Elim: 2
5 | | D              ↔Elim: 1,4
6 | | D ∧ B         ∧ Intro: 3,5
7 | | ¬ (D ∧ B)       ∧ Elim: 2
8 | | ⊥              ⊥ Intro: 6,7
9 | ¬B             ¬ Intro: 3-8
```

STRATEGY: {¬(Tet(a) ∧ Small(a)), RightOf(b,c) → Small(b), LeftOf(c,b), a=b} /∴ ¬Tet(a)

1

```
1 | ¬(Tet(a) ∧ Small(a))
2 | RightOf(b,c) → Small(b)
3 | LeftOf(c,b)
4 | a = b
  |----
5 | | Tet(a)
  | |----
: | :
  | | ⊥
  | ¬Tet(a)
```

⇐ 1 Since goal is a negation, assume its opposite, planning for ¬Intro.

2

⇓

```
1 | ¬(Tet(a) ∧ Small(a))
2 | RightOf(b,c) → Small(b)
3 | LeftOf(c,b)
4 | a = b
  |----
5 | | Tet(a)
  | |----
: | :
  | | Tet(a) ∧ Small(a)
  | | ⊥
  | ¬Tet(a)
```

2 ⇓ The only negation we have is on line 1, so we aim for 'Tet(a) ∧ Small(a)' to contradict line 1.

3

```
1 | ¬(Tet(a) ∧ Small(a))
2 | RightOf(b,c) → Small(b)
3 | LeftOf(c,b)
4 | a = b
  |----
5 | | Tet(a)
  | |----
: | :
  | | Small(a)
  | | Tet(a) ∧ Small(a)      ∧Intro
  | | ⊥                      ⊥Intro
  | ¬Tet(a)
```

⇐ 3 We already have the first conjunct, but still need the second, 'Small(a)'.

4

```
1 | ¬(Tet(a) ∧ Small(a))
2 | RightOf(b,c) → Small(b)
3 | LeftOf(c,b)
4 | a = b
  |----
5 | | Tet(a)
  | |----
: | :
  | | RightOf(b,c)
  | | Small(b)               →Elim
: | :
  | | Small(a)
  | | Tet(a) ∧ Small(a)      ∧Intro
  | | ⊥                      ⊥Intro
  | ¬Tet(a)                  ¬Intro
```

⇐ 4 Given line 4, if we could get 'Small(b)' out of line 2, we should eventually be able to get 'Small(a)'. Aim for the antecedent of line 2 to get 'Small(b)'

5 ⇓ The meaning relationship between 'RightOf' and 'LeftOf', is programmed into Fitch, so it lets us derive 'RightOf(b,c)' from 'LeftOf(c,b). (Without that, we need a premise, not stated here, specifying that each implies the other. See variant on next page.)

5 ⇒

```
1 | ¬(Tet(a) ∧ Small(a))         P
2 | RightOf(b,c) → Small(b)      P
3 | LeftOf(c,b)                  P
4 | a = b                        P
  |----
5 | | Tet(a)
  | |----
6 | | RightOf(b,c)         AnaCon:3
7 | | Small(b)               →Elim
: | :
  | | Small(a)
  | | Tet(a) ∧ Small(a)      ∧Intro
  | | ⊥                      ⊥Intro
  | ¬Tet(a)
```

6 ⇓ ⇓ ⇓ If we apply=Elim strictly, allowing term to the right of '=' to replace that on left but not vice versa, we need 'b = a' to be able to derive 'Small(a)' from 'Small(b)'. (Fitch is not as strict, letting us omit this reversal. See variant on next page.)

```
 1 | ¬(Tet(a) ∧ Small(a))           P
 2 | RightOf(b,c) → Small(b)        P
 3 | LeftOf(c,b)                    P
 4 | a = b                          P
   |----
 5 | | Tet(a)
   | |----
 6 | | RightOf(b,c)           AnaCon:3
 7 | | Small(b)             →Elim: 2,6
 8 | | a = a                    =Intro
 9 | | b = a                =Elim: 4,8
10 | | Small(a)             =Elim: 7,9
11 | | Tet(a) ∧ Small(a)   ∧Intro:5,10
12 | | ⊥                   ⊥Intro:1,11
13 | ¬Tet(a)               ¬Intro: 5-12
```

The derivation on the previous page uses AnaCon, and applies =Elim strictly:

1	¬(Tet(a) ∧ Small(a))	P
2	RightOf(b,c) → Small(b)	P
3	LeftOf(c,b)	P
4	a = b	P
5	Tet(a)	
6	RightOf(b,c)	AnaCon:3
7	Small(b)	→Elim: 2,6
8	a = a	=Intro
9	b = a	=Elim: 4,8
10	Small(a)	=Elim: 7,9
11	Tet(a) ∧ Small(a)	∧Intro: 5,10
12	⊥	⊥Intro:1,11
13	¬Tet(a)	¬Intro: 5-12

Remember that AnaCon is not really a derivation rule in F, but a shortcut permitted by the software, Fitch. In most systems of logic, the set of rules does not have any rules for predicates, except the very special '=', representing (numerical) identity. Like any standard set of rules of logic, F does not include AnaCon, since each related pair of predicates in the language would require its own rule. The system of rules would then be infinite, making it impractical. To compensate for lack of AnaCon in a standard set of rules, in any argument where we use AnaCon, we make explicit an originally unstated premise specifying the relevant relationship between the specific predicates used in the argument (as in step 5 below).

It is common for a system of rules of logic to state their versions of the =Elim to allow the replacement of either term in an identity sentence for the other. Fitch, not being as strict as F, lets us do this, too.

Taking these two differences into account, a more standard variation on the derivation above would look like this:

1	¬(Tet(a) ∧ Small(a))	P
2	RightOf(b,c) → Small(b)	P
3	LeftOf(c,b)	P
4	a = b	P
5	RightOf(b,c) ↔ LeftOf(c,b)	P
6	Tet(a)	
7	Tet(b)	=Elim: 4,6
8	RightOf(b,c)	↔Elim: 3,5
9	Small(b)	→Elim: 2,8
10	Tet(b) ∧ Small(b)	∧Intro: 7,9
11	¬(Tet(b) ∧ Small(b)	=Elim: 1,10
12	⊥	⊥Intro: 10,11
13	¬Tet(a)	¬Intro: 6-12

1

```
1 | J                      P
2 | ¬H → (J → K)           P
3 | (K ∧ J) ↔ ¬H           P
  ⋮
  | K ↔ ¬H
```

> STRATEGY:
> {J, ¬H → (J → K), (K ∧ J) ↔ ¬H} /∴ K ↔ ¬H

2

To get a biconditional (a sentence with '↔' as its main operator), use the ↔ Intro rule. This requires 2 subproofs, one starting from the sentence to the left of the '↔' and going to the right, and the other in the opposite direction. Our first step is to set up these subproofs.

```
1 | J                      P
2 | ¬H → (J → K)           P
3 | (K ∧ J) ↔ ¬H           P
4 |  | K
  ⋮  ⋮
  |  | ¬H
  |
  |  | ¬H
  ⋮  ⋮
  |  | K
  | K ↔ ¬H                 ↔Intro
```

3

```
1 | J                      P
2 | ¬H → (J → K)           P
3 | (K ∧ J) ↔ ¬H           P
4 |  | K
  ⋮  ⋮
  |  | K ∧ J         ∧ Intro: 1,3
  |  | ¬H              ↔ Elim
  |
  |  | ¬H
  ⋮  ⋮
  |  | K
  | K ↔ ¬H             ↔ Intro
```

We now try to get from 'K' to '¬H'. We could use line 3 to get '¬H' if we only had 'K ∧ J', so we aim for that. We just need the two conjuncts, which we have already on lines 1 and 4.

4

Having completed the first subproof, we try to fill in the second. To get 'K', 2 looks like the most promising premise, although it will not give us 'K' directly. With line 7, though, line 2 will let us get 'J → K'. If we can also get 'J', we will be able to get 'K'.

```
1 | J                      P
2 | ¬H → (J → K)           P
3 | (K ∧ J) ↔ ¬H           P
4 |  | K
5 |  | K ∧ J         ∧ Intro: 1,4
6 |  | ¬H            ↔Elim: 3,5
  |
7 |  | ¬H
8 |  | J → K         →Elim: 2,7
  ⋮  ⋮
  |  | K
  | K ↔ ¬H              ↔Intro
```

5

```
 1 | J                        P
 2 | ¬H → (J → K)             P
 3 | (K ∧ J) ↔ ¬H             P
 4 |  | K
 5 |  | K ∧ J           ∧ Intro: 1,4
 6 |  | ¬H              ↔ Elim: 3,5
   |
 7 |  | ¬H
 8 |  | J → K           → Elim: 2,7
 9 |  | K               → Elim: 1,8
10 | K ↔ ¬H        ↔ Intro: 4-6,7-9
```

Looking back at the premises, we see that we already have 'J'. So we just need to fill in the justifications to complete this proof.

1

```
1 | C ∨ L              P
2 | L → M              P
3 | (C ∧ J) → M        P
  ⋮
  | J → M
```

STRATEGY for:
{ C ∨ L, L → M, (C ∧ J) → M}/∴ J → M

2

When aiming for a conditional, set up for → Intro by assuming the antecedent and aiming for the consequent.

```
1 | C ∨ L              P
2 | L → M              P
3 | (C ∧ J) → M        P
4 | | J
  ⋮ ⋮
  | | M
  | J → M        → Intro
```

3

Line 1 says ‘C’ is true or ‘L’ is. We will now show that ‘M’ follows either way. We first assume ‘C’ and show that it leads to M. Then we give up ‘C’, assume ‘L’ instead, and show that ‘L’ leads to ‘M’. ⇒ ⇒ ⇒

3 ⇓

```
1 | C ∨ L              P
2 | L → M              P
3 | (C ∧ J) → M        P
4 | | J
5 | | | C
  ⋮ ⋮ ⋮
  | | | M

  | | | L
  ⋮ ⋮ ⋮
  | | | M
  | | M          ∨ Elim
  | J → M        →Intro
```

4

```
1 | C ∨ L              P
2 | L → M              P
3 | (C ∧ J) → M        P
4 | | J
5 | | | C
  ⋮ ⋮ ⋮
  | | | M

  | | | L
  | | | M          →Elim
  | | M            ∨ Elim
  | J → M          →Intro
```

⇐ 4

We notice that ‘L’ leads directly to ‘M’, but ‘C’ does not. This shows us how to justify ‘M’ in the subproof that begins with L.

5

Now we need to get from ‘C’ to ‘M’. We see that we could do that by →Elim with the help of line 3, if only we could get the conjunction ‘C ∧ J’. And to get that, all we need to do is join 4 and 5 by ∧ Intro.

Once we’ve done this, we just need to complete the justifications to finish the proof.

5 ⇒

```
1  | C ∨ L                    P
2  | L → M                    P
3  | (C ∧ J) → M              P
4  | | J
5  | | | C
6  | | | C ∧ J        ∧ Intro: 4,5
7  | | | M            → Elim: 3,6

8  | | | L
9  | | | M            → Elim: 2,8
10 | | M        ∨ Elim: 1,5-7,8-9
11 | J → M          → Intro: 4-10
```

1

```
1 | G                    P
2 | ¬F → (G ↔ H)         P
  :
  | F ∨ H
```

Example using strategy 2.c:
{G, ¬F →(G ↔ H)} /∴ F ∨ H

2

When aiming for a disjunction:

A. If one disjunct is easy to get, do that, then use ∨I.

B. If neither disjunct is easy to get, see if you have or can easily get another disjunction. If so, maybe each disjunct in that one will lead to one disjunct in the goal disjunction. So apply ∨Elim to the one you have, using ∨Intro in each subproof to get the disjunction you want.

C. If neither (A) nor (B) works, assume the opposite of your goal. Then use ¬I or ¬E to get the disjunction you want.

```
1 | G                    P
2 | ¬F → (G ↔ H)         P
3 |   | ¬(F ∨ H)
  :   :
  | F ∨ H                ¬E
```

3

Assume one disjunct as PA. From this, get the disjunction, contradicting the PA. Apply ¬Intro.

In this case, 'F' was chosen as PA because its negation can be used with 2 to apply the →Elim rule.

```
1 | G                          P
2 | ¬F → (G ↔ H)               P
3 |   | ¬(F ∨ H)
4 |   |   | F
5 |   |   | F ∨ H              ∨ Intro: 4
6 |   |   | ⊥                  ⊥ Intro: 3,5
7 |   | ¬F                     ¬Intro: 4-6
  :   :
  |   | F ∨ H                  ∨ Intro
  |   | ⊥                      ⊥ Intro: 3,?
  | F ∨ H                      ¬ Elim
```

```
1 | G                          P
2 | ¬F → (G ↔ H)               P
3 |   | ¬(F ∨ H)
4 |   |   | F
5 |   |   | F ∨ H     ∨Intro: 4
6 |   |   | ⊥         ⊥ Intro: 3,5
7 |   | ¬F            ¬Intro: 4-6
  :   :
  |   | H
  |   | F ∨ H                  ∨ Intro
  |   | ⊥
  | F ∨ H                      ¬ Elim
```

⇐ 4

Next try to get the OTHER disjunct in our target disjunction. Then we can use ∨I to get that disjunction, again contradicting line 2.

Stages 2 and 3 of the process above use a common **GENERAL STRATEGY**:

```
|   | ¬(P ∨ Q)
|   |   | P
|   |   | P ∨ Q              ∨ Intro
|   |   | ⊥                  ⊥ Intro
|   | ¬P                     ¬ Intro
:   :
|   | Q
|   | P ∨ Q                  ∨ Intro
|   | ⊥                      ⊥ Intro
| P ∨ Q                      ¬ Elim
```

5

We see that by applying →Elim to 2 and 6 we get 'G ↔ H'. Using this and line 1, we reach our goal of 'H' by ↔Elim.

```
1  | G                             P
2  | ¬F → (G ↔ H)                  P
3  |   | ¬(F ∨ H)
4  |   |   | F
5  |   |   | F ∨ H           ∨Intro: 4
6  |   |   | ⊥               ⊥ Intro: 3,5
7  |   | ¬F                  ¬Intro: 4-6
8  |   | G ↔ H               →Elim: 2,7
9  |   | H                   ↔Elim: 1,8
10 |   | F ∨ H               ∨ Intro: 9
11 |   | ⊥                   ⊥ Intro: 3,10
12 | F ∨ H                   ¬Elim: 3-11
```

EXAMPLES USING THE ∨ELIM RULE

1	$A \vee P$	P	Teresa has either appendicitis or a severe case of food poisoning.
2	$(A \to S) \wedge (S \to H)$	P	If she has appendicitis, she'll need surgery, which would keep her in the hospital tomorrow.
3	$H \to \neg W$	P	Of course, she can't be at work tomorrow if she's in the hospital.
4	$(P \to \neg E) \wedge (\neg E \to \neg W)$	P	On the other hand, if she has severe food poisoning, she won't be up to eating anything for at least another day, in which case she won't be able to come to work tomorrow.
5	A		Consider the possibility that Teresa has appendicitis. (Line 1 tells us she has appendicitis or food poisoning (or both) but doesn't say which. We are temporarily assuming she has appendicitis.)
6	$A \to S$	∧Elim: 2	If she has appendicitis, she'll need surgery.
7	S	→ Elim: 5,6	So on this assumption, she'll need surgery..
8	$S \to H$	∧ Elim: 2	If she needs surgery, she'll still be in the hospital tomorrow.
9	H	→ Elim: 7,8	So on this assumption, she'll be in the hospital tomorrow.
10	$\neg W$	→ Elim: 3,9	So on this assumption, she will not be at work tomorrow.
11	P		On the other hand, suppose she has food poisoning.
12	$P \to \neg E$	∧ Elim: 4	If she has food poisoning, she won't eat for at least another day.
13	$\neg E \to \neg W$	∧ Elim: 4	If she can't eat for another day, she can't work tomorrow.
14	$\neg E$	→Elim 11,12	So ON THIS ASSUMPTION, she won't eat for another day.
15	$\neg W$	→Elim: 13,14	So ON THIS ASSUMPTION, she won't work tomorrow.
16	$\neg W$	v Elim:1,5-10,11-15	So Teresa will not work tomorrow.

Our conclusion doesn't depend on the specific diagnosis because we would reach the same conclusion regardless of which of the two possible problems Teresa has. We were told she has (at least) one of two problems, and we saw that either one would keep her from being at work. So we can be sure she won't be at work without figuring out which of these two things is wrong with her.

1	$L(n) \vee P(n)$	P	Nancy will lose her job or get a promotion.
2	$L(n) \to S(n)$	P	If she loses her job, she'll limit her spending to necessities.
3	$S(n) \to \neg V(n)$	P	If she's only spending on necessities, she won't take a vacation.
4	$C(n) \leftrightarrow P(n)$	P	She'll buy a new car if, but only if, she gets a promotion.
5	$\neg(C(n) \wedge V(n))$	P	She can't afford both a car and a vacation (even with a promotion).
6	$L(n)$		First CONSIDER THE POSSIBILITY THAT SHE LOSES HER JOB.
7	$S(n)$	→Elim: 2	IN THAT CASE, she will limit her spend to necessities.
8	$\neg V(n)$	→Elim: 3,7	IN THAT CASE, she won't take a vacation.
9	$P(n)$		On the other hand, SUPPOSE SHE GETS THE PROMOTION.
10	$C(n)$	↔ Elim: 4,9	IN THAT CASE, she'll buy a new car.
11	$V(n)$		Suppose also that she goes on vacation.
12	$C(n) \wedge V(n)$	∧Intro:10,11	THEN she would be both buying a car and taking a vacation,
13	$\perp$	⊥ Intro: 12,5	contradicting the premise that she CAN'T do BOTH of those.
14	$\neg V(n)$	11-13,¬ Intro	So, on the assumption that she gets the promotion, she would not take a vacation.
15	$\neg V(n)$	v Elim:1,6-8,9-14	So Nancy will not be taking a vacation.

We don't need to find out whether Nancy is losing her job or getting a promotion to be sure that she won't take a vacation. Step 1 says *at least one* of those two things will happen. We showed that *either way*, she wouldn't take a vacation. So she definitely won't take a vacation.

Example using strategy 3:
{J → (K ∨ L), M ↔ (J ∧ K), L → (N ∧ P), J ∧ ¬S} /∴ N ∨ M

1

```
1 | J → (K ∨ L)    P
2 | M ↔ (J ∧ K)    P
3 | L → (N ∧ P)    P
4 | J ∧ ¬S         P
  |  ⋮
  | N ∨ M
```

2

Our goal sentence is not in any of our premises. Neither of its disjuncts is obviously the easy one to get. We do not have a disjunction, but we can get one easily. Our strategy hints tell us to get that disjunction, then set up the subderivations we would need to apply ∨ Elim to that disjunction.

```
1 | J → (K ∨ L)          P
2 | M ↔ (J ∧ K)          P
3 | L → (N ∧ P)          P
4 | J ∧ ¬S               P
5 | J              ∧Elim: 4
6 | K ∨ L        →Elim: 1,5
7 | | K
  | |  ⋮
  | |
  | | L
  | |  ⋮
  | |
  | N ∨ M             ∨Elim
```

3

```
1 | J → (K ∨ L)          P
2 | M ↔ (J ∧ K)          P
3 | L → (N ∧ P)          P
4 | J ∧ ¬S               P
5 | J              ∧Elim: 4
6 | K ∨ L        →Elim: 1,5
7 | | K
  | |  ⋮
  | | N ∨ M
  |
  | | L
  | |  ⋮
  | | N ∨ M
  | N ∨ M            ∨ Elim
```

COMMON STRATEGY: Often when we have one disjunction and are aiming for another, we get the disjunction we are aiming for inside each subderivation. Then we pull it out by ∨Elim.

4

In the first subderivation, we see if we can get one disjunct of the desired disjunction easily. Here it's easy to get 'M'. So we do that, and then apply∨Intro to get the disjunction.

```
 1 | J → (K ∨ L)          P
 2 | M ↔ (J ∧ K)          P
 3 | L → (N ∧ P)          P
 4 | J ∧ ¬S               P
 5 | J              ∧Elim: 4
 6 | K ∨ L        →Elim: 1,5
 7 | | K
 8 | | J ∧ K      ∧Intro: 5,7
 9 | | M          ↔Elim: 2,8
10 | | N ∨ M      ∨ Intro: 9
   |
   | | L
   | |  ⋮
   | | N ∨ M
   | N ∨ M             ∨Elim
```

5

```
 1 | J → (K ∨ L)                 P
 2 | M ↔ (J ∧ K)                 P
 3 | L →(N ∧ P)                  P
 4 | J ∧ ¬S                      P
 5 | J                     ∧Elim: 4
 6 | K ∨ L               →Elim: 1,5
 7 | | K
 8 | | J ∧ K            ∧Intro: 5,7
 9 | | M              ↔Elim: 2,8
10 | | N ∨ M            ∨ Intro: 9
   |
11 | | L
12 | | N ∧ P           →Elim: 3,11
13 | | N                ∧ Elim: 12
14 | | N ∨ M           ∨ Intro: 13
15 | N ∨ M  ∨Elim:6,7-10,11-14
```

The second subderivation uses the same strategy in the first. In this case it is easy to get 'N'. So we get 'N', then apply ∨Intro to get 'N ∨ M' inside the subderivation. Once we have 'N ∨ M' in both subderivations, we can justify 'N ∨ M' outside the subderivations by ∨Elim.

COMMON STRATEGY to get one disjunction from another:

```
| A ∨ B
|  ⋮
| | A
| |  ⋮
| | P
| | P ∨ Q   ∨Intro
|
| | B
| |  ⋮
| | D
| | P ∨ Q   ∨Intro
| P ∨ Q     ∨Elim
```

Use ∨Elim on 1st disjunction to get the second. Inside the 2 subderivations, get the 2 disjuncts in the goal disjunction, then reach goal by ∨Intro in each. Finally, apply ∨Elim.

(1)

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
	⋮	
	¬D ∨ E	

STRATEGY:
{A ∧ (B ∨ C), B → (¬D ↔ A), (A ∧ C) → E} /∴ ¬D ∨ E

(2)

When aiming for a disjunction, we first see if one disjunct is obviously very simple to get. If so, get it, then use ∨ Intro.

Since neither disjunct is clearly easy, we look for a different disjunction we have or can easily get. Here, it is easy to get 'B ∨ C' from line 1.

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
4	B ∨ C	
	⋮	
	¬D ∨ E	

(3)

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
4	B ∨ C	∧ Elim: 1
	| B	
	| ⋮	
	| ¬D ∨ E	
	| C	
	| ⋮	
	| ¬D ∨ E	
	¬D ∨ E	∨ Elim

(3) We set up subproofs to apply ∨Elim to 'B ∨ C'. We aim for our goal disjunction within each subproof, then use the ∨Elim rule.

(4)

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
4	B ∨ C	∧Elim: 1
	| B	
	| ⋮	
	| ¬D	
	| ¬D ∨ E	∨ Intro
	| C	
	| ⋮	
	| ¬D ∨ E	
	¬D ∨ E	∨ Elim

(4) In each subproof, we try to get one disjunct, then use ∨Intro to build the disjunction we want. Since line 2 connects 'B' and '¬D', but no sentence connects 'B' and 'E', we aim for '¬D'.

(5)

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
4	B ∨ C	∧ Elim: 1
5	A	∧ Elim: 1
6	| B	
7	| ¬D↔A	→Elim: 2,6
8	| ¬D	↔Elim: 6,7
9	| ¬D ∨ E	∨Intro: 8
10	| C	
	| ⋮	
	| ¬D ∨ E	∨ Elim
	¬D ∨ E	∨ Elim

We saw we'll use line 2. Now we realize we also need 'A'. We can get it by ∧ Elim.

We put 'A' *before* the first subproof because we see we'll use it in the second subproof, too. (If we put it *inside* the subproof, we'll need it in both of them.)

(6)

⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒ ⇒

To finish, we get '¬D ∨ E' by ∨Intro through the other disjunct, 'E', in the second subproof.

1	A ∧ (B ∨ C)	P
2	B → (¬D ↔ A)	P
3	(A ∧ C) → E	P
4	B ∨ C	∧ Elim: 1
5	A	∧ Elim: 1
6	| B	
7	| ¬D ↔A	→Elim: 2,6
8	| ¬D	↔Elim: 6,7
9	| ¬D ∨ E	∨Intro: 8
10	| C	
11	| A ∧ C	∧Intro: 5,10
12	| E	→Elim: 3,11
13	| ¬D ∨ E	∨ Intro: 12
14	¬D ∨ E	∨Elim:4,6-9,10-13