\( \neg ( ( \neg v ) \land r ) \) \Rightarrow \( ( p \Rightarrow r ) \)

\[
\begin{array}{c}
0 \\
1 \\
0 \\
1 \\
1 \\
0 \\
0 \\
0 \\
0 \\
1 \\
0 \\
1 \\
0 \\
0 \\
1
\end{array}
\]

\[ p = 1, \quad r = 0. \]

Note assigning values.
Trees for Predicate Logic

Method should give a decision procedure.

Yes/No for validity

Church-Turing proved no such method.

The following method "sometimes" works
Idea similar

Negate the formula

& try to construct a "model" of the negation (counterexample) or prove it is impossible

= new rules
  - Negate & simplify rules
    \[ \neg \exists x \equiv \forall x \neg \]
  - Instantiation rules = "make instances"
\((\exists x) (\forall x)\)  
\(Ra\ a\)  
\(\left\{\text{possibly new constant}\right\}\)

\((\forall x) (\exists x)\)  
for each such a also have

\(Ra\).

\((\exists x) (\exists y) (B_{xy})\)  
\(R_{ab} B_{cb}\)  
\(\delta c, \delta b\)
Idea: reduce predicate logic to a model in propositional logic.
1. \((\exists x) (F_x) \land (\exists x) G_x\) \Rightarrow \((\exists x) (F_x \land G_x)\)

2. \((\exists x) F_x \land (\exists y) G_y\) (Bound variable)
   
   \((\exists x) F_x \land (\exists y) G_y\) from 1.

3. \(\neg (\exists x) (F_x \land G_x)\)

4. \((\forall x) \neg (F_x \land G_x)\) from 3.

5. \(\exists x F_x\) from 2.

6. \(\exists y G_y\) from 2.

7. \(F_a\) (instantiation of \(x\) in 5.)

8. \(G_b\) (\(\ldots y \ldots b\))
Counterexample

universe of 2

a, b

F a, G b, G F b, G F a.

Invalid
1. \((\forall x)(\forall y)(\forall z)(Fxyz \rightarrow Fzyx)\)

2. \((\exists x)(\exists y)(Fxxxy)\)

Conclusion: \((\exists x)(\exists y)(\exists z)(Gxyz)\)

7. \((\exists x)(\exists y)Gxyz\)

4. \((\forall x)(\forall y)Gxyz\)

5. \((\exists x)(Fanya)\)

6. \(Faab\)

7. \(Faab \rightarrow Gbaaa\)

8. \(Gbaaa\) (Mohan money)

8. \(Gbaaa\)