BITS Pilani Hyderabad Campus
CS F214 Logic in Computer Science,
I Semester 2021-2022
Lecture Notes
Lecture 6-10

Prove: $\neg q \to \neg p \vdash p \to q$

Solution:

$$
\begin{array}{ll}
1. \neg q \to \neg p & \text{premise} \\
\boxed{\begin{array}{ll} 2. p & \text{assumption} \\ 3. \neg\neg p & \neg\neg i \ 2 \\ 4. \neg\neg q & \text{MT } 3,1 \\ 5. q & \neg\neg e \ 4 \end{array}} & \\
6. p \to q & \to \text{i } 2\text{-}5
\end{array}
$$

Consider,

$$
\begin{array}{ll}
\boxed{\begin{array}{ll} 1. p & \text{assumption} \end{array}} & \\
2. p \to p & \to \text{i } 1\text{-}1
\end{array}
$$

Above is a poof of the statement $\vdash (p \to p)$

Def: A logical formula ϕ with a valid sequent $\vdash \phi$ is called a ***Theorem.***
Remarks: Any sequent $\phi \vdash \psi$ is equivalent to $\vdash \phi \to \psi$.

Proof of $\vdash \phi \to \psi$:

$$
\begin{array}{l}
\boxed{\begin{array}{ll} 1. \phi & \text{assumption} \\ 2. & \\ . & \\ . & \\ n. \psi & <> \end{array}} \\
\phi \to \psi \quad \to i \text{ 1-n (n+1)}
\end{array}
$$

Remark(b): $\phi_1, \phi_2, ..., \phi_n \vdash \psi$ is equivalent to $\vdash \phi_1 \to (\phi_2 \to (...(.. \to (\phi_n \to \psi))$

5 Rules for Disjunction

5.1 OR-Introduction

$$\frac{\phi}{\phi \lor \psi} \lor_{i1}$$

$$\frac{\phi}{\psi \lor \phi} \lor_{i2}$$

5.2 OR-Elimination

$$\frac{\phi \lor \psi \quad \boxed{\begin{array}{c} \phi \\ \vdots \\ \chi \end{array}} \quad \boxed{\begin{array}{c} \psi \\ \vdots \\ \chi \end{array}}}{\chi} \lor_e$$

Prove: $p \lor q \vdash q \lor p$

1.$p \lor q$	premise
2.q	assumption
3.$q \lor p$	$\lor_{i2}$ 2
4.q	assumption
5.$q \lor p$	$\lor_i$4
6.$q \lor p$	$\lor_e$ 1,2-3,4-5

Prove: $q \to r \vdash p \lor q \to p \lor r$

1.$q \to r$	premise
2.$p \lor q$	assumption
3.p	assumption
4.$p \lor r$	$\lor_i$ 3
5.q	assumption
6.r	$\to e$ 5,1
7.$p \lor r$	$\lor_{i2}$ 6
8.$p \lor r$	$\lor_e$ 2,3-4,5-7
9.$(p \lor q) \to (p \lor r)$	$\to$ i 2-8

5.3 “Copy” Rule

$\perp p \to (q \to p)$

1.p	assumption
2.q	assumption
3.p	**copy 1**
4.$q \to p$	$\to i$ 2-3
5.$p \to (q \to p)$	$\to i$ 1-4

6 Rules for Negation

6.1 Contradiction

- Contradictions are expressions of the form $\phi \land \neg\phi$ or $\neg\phi \land \phi$ where ϕ is any proposition.
- $\perp$ represents a contradiction.
- Any proposition can be derived from contradiction.

6.2 Bottom Elimination

$$\frac{\perp}{\psi}\perp e$$

6.3 Not Elimination

$$\frac{\phi \quad \neg\phi}{\perp}\neg e$$

Example: Prove $\neg p \ \lor q \vdash p \to q$ is valid.

$$
\begin{array}{ll}
1. \neg p \ \vee q & \text{premise} \\
\boxed{\begin{array}{ll}
2. \neg p & \text{assumption} \\
\boxed{\begin{array}{ll}
3. p & \text{assumption} \\
4. \bot & \neg e 3, 2 \\
5. q & \bot e 4 \\
6. p \to q & [\to i 3 - 5]
\end{array}}
\end{array}} & \\
\boxed{\begin{array}{ll}
7. q & \text{assumption} \\
\boxed{\begin{array}{ll}
8. p & \text{assumption} \\
9. q & \text{from } 7 \\
10. p \to q & [\to i 8 - 9]
\end{array}}
\end{array}} & \\
11. p \to q \vee & 1, 2 - 6, 7 - 10
\end{array}
$$

6.4 Negation Introduction and Proof by Contradiction

6.4.1 Negation Introduction

$$
\frac{\boxed{\begin{array}{c} \phi \\ \vdots \\ \bot \end{array}}}{\neg \phi} \neg i
$$

6.4.2 Proof By Contradiction (Reductio Ad Absurdum)

$$
\frac{\boxed{\begin{array}{c} \neg \phi \\ \vdots \\ \bot \end{array}}}{\phi} PBC
$$

Example: Prove the following sequent is valid.

$$p \to q, p \to \neg q \bot \neg p$$

Solution:

$$
\begin{array}{ll}
1. p \to q & \text{premise} \\
2. p \to \neg q & \text{premise}
\end{array}
$$

$$
\boxed{\begin{array}{ll}
3. p & \text{assumption(3)} \\
4. q & \to e \text{ 3,1} \\
5. \neg q & \to e \text{ 3,2} \\
6. \bot & \neg e \text{ 4,5}
\end{array}}
$$

$$
\begin{array}{ll}
7. \neg p & \neg i \text{ 3-6}
\end{array}
$$

Example: Prove $p \to (q \to r), p, \neg r \vdash \neg q$ is valid, without using MT.
Solution:

$$
\begin{array}{ll}
1. p \to (q \to r) & \text{premise} \\
2. p & \text{premise} \\
3. \neg r & \text{premise}
\end{array}
$$

$$
\boxed{\begin{array}{ll}
4. q & \text{assumption} \\
5. q \to r & \to e \text{ 2,1} \\
6. r & \to e \text{ 4,5} \\
7. \bot & \neg e \text{ 3,6}
\end{array}}
$$

$$
\begin{array}{ll}
8. \neg q & \neg_i \text{ 4-7}
\end{array}
$$

7 Derived Rules

MT can be derived from $\to e, \neg e$ and $\neg_i$.

$$
\begin{array}{ll}
1. \phi \to \psi & \text{premise} \\
2. \neg \psi & \text{premise}
\end{array}
$$

$$
\boxed{\begin{array}{ll}
3. \neg \phi & \text{assumption)} \\
4. \neg \psi & \to e \text{ 3,1} \\
5. \bot & \neg e \text{ 2,4}
\end{array}}
$$

$$
\begin{array}{ll}
6. \neg \phi & \neg_i 3 - 5
\end{array}
$$

8 Law of the Excluded Middle (LEM)

- It is also called as Tertium non datur. (There is no third possibility)
- Says that $\phi \vee \neg \phi$ is always true.

$$\vdash \phi \vee \neg\phi$$

1. $\neg(\phi \vee \neg\phi)$	assumption
2. ϕ	assumption
3. $\phi \vee \neg\phi$	$\vee_i$ 2
4. $\bot$	$\neg e$ 3,1
5. $\neg\phi$	$\neg_i$ 2-4
6. $\phi \vee \neg\phi$	$\vee_{i2}$ 5
7. $\bot$	$\neg e$ 6
8. $\phi \vee \neg\phi$	PBC 1-7

Example: Using LEM, show that $p \to q \vdash \neg p \vee q$ as valid.

1. $p \to q$	premise
2. $p \vee \neg p$	LEM
3. p	assumption
4. q	$\to e$ 3,1
5. $\neg p \vee q$	$\vee_{i2}$ 4
6. $\neg p$	assumption
7. $\neg p \vee q$	$\vee_i$ 6
8. $\neg p \vee q$	$\vee_e$ 2,3-5,6-7

9 Provable Equivalence

- ϕ and ψ are provable equivalent if and only if the sequents $\phi \vdash \psi$ and $\psi \vdash \phi$ are valid.
- It is denoted by $\phi \dashv\vdash \psi$.
- Ultimately we could define the $\phi \dashv\vdash \psi$ as $\vdash (\phi \to \psi) \wedge (\psi \to \phi)$.

$$\neg(p \to q) \vdash p \wedge \neg q$$

$$
\begin{array}{ll}
1.\neg(p \to q) & \text{premise} \\
\boxed{
\begin{array}{ll}
2.\neg(p \wedge \neg q) & \text{assumption} \\
\boxed{
\begin{array}{ll}
3.p & \text{assumption(3)} \\
\boxed{
\begin{array}{ll}
4.\neg q & \text{assumption} \\
5.p \wedge \neg q & \wedge_i \text{ 3,4} \\
6.\bot & \neg e \text{ 2,5}
\end{array}
} \\
7.q & \text{PBC 4-6}
\end{array}
} \\
8.p \to q & \to i \text{ 3-7} \\
9.\bot & \neg e \text{ 8,1}
\end{array}
} \\
10.p \wedge \neg q & \text{PBC 2-9}
\end{array}
$$