CS113 LAB 3 - Example 3.3

Example 3.3

Show that $\sim(p \to q) \equiv p \wedge (\sim q)$

p	q	$(p \to q)$	$\sim(p \to q)$	$\sim q$	$(p \wedge \sim q)$	$(\sim(p \to q) \leftrightarrow (p \wedge \sim q))$
F	F	T	F	T	F	T
F	T	T	F	F	F	T
T	F	F	T	T	T	T
T	T	T	F	F	F	T

Notice that ProofPower automatically applied the double-negation rule

$$
\begin{aligned}
\sim(p \to q) &\equiv \sim(\sim p \vee q) && \text{(Conditional)}\\
&\equiv \sim(\sim p) \wedge (\sim q) && \text{(De Morgan's)}\\
&\equiv p \wedge (\sim q) \blacksquare && \text{(Double Negation)}
\end{aligned}
$$

SML

```
val L0 = asm_rule ⌜ ¬(p ⇒ q) ⌝;
val T = ⌜ ¬(p ⇒ q) ⇔ ¬(¬p ∨ q) ⌝;
val conditional = prove_rule [ ] T;
val L1 = rewrite_rule [conditional] L0;
rewrite_rule [¬_∨_thm] L1;
```

```
:) val T = ⌜¬ (p ⇒ q) ⇔ ¬ (¬ p ∨ q)⌝: TERM
:) val conditional = ⊢ ¬ (p ⇒ q) ⇔ ¬ (¬ p ∨ q): THM
:) val L1 = ¬ (p ⇒ q) ⊢ ¬ (¬ p ∨ q): THM
:) val it = ¬ (p ⇒ q) ⊢ p ∧ ¬ q: THM
```

Example 3.5

Show that $p \to q \equiv \sim q \to \sim p$

p	q	$(p \to q)$	$\sim q$	$\sim p$	$(\sim q \to \sim p)$	$((p \to q) \leftrightarrow (\sim q \to \sim p))$
F	F	T	T	T	T	T
F	T	T	F	T	T	T
T	F	F	T	F	F	T
T	T	T	F	F	T	T

Notice that ProofPower automatically applied conditional rule and double-negation rule

$$
\begin{aligned}
p \to q &\equiv (\sim p) \vee q && \text{(Conditional)}\\
&\equiv \sim [p \wedge (\sim q)] && \text{(De Morgan's)}\\
&\equiv \sim [(\sim q) \wedge p] && \text{(Commutative)}\\
&\equiv [\sim (\sim q)] \vee (\sim p) && \text{(De Morgan's)}\\
&\equiv q \vee (\sim p) && \text{(Double Negation)}\\
&\equiv \sim q \to \sim p \blacksquare && \text{(Conditional)}
\end{aligned}
$$

SML

```
val L0 = asm_rule ⌜ p ⇒ q ⌝;
val L1 = rewrite_rule [⇒_thm] L0;
val T = ⌜ ¬p ∨ q ⇔ ¬(p ∧ ¬q) ⌝;
val demorgan = prove_rule [ ] T;
val L2 = rewrite_rule [demorgan] L1;
val T = ⌜ p ∧ ¬q ⇔ ¬q ∧ p ⌝;
val commutative = prove_rule [ ] T;
val L3 = rewrite_rule [commutative] L2;
val L4 = rewrite_rule [¬_∧_thm] L3;
val T = ⌜ q ∨ ¬p ⇔ ¬q ⇒ ¬p ⌝;
val conditional = prove_rule [ ] T;
rewrite_rule [conditional] L4;
```

```
:) val L1 = p ⇒ q ⊢ ¬ p ∨ q: THM
:) val T = ⌜¬ p ∨ q ⇔ ¬ (p ∧ ¬ q)⌝: TERM
:) val demorgan = ⊢ ¬ p ∨ q ⇔ ¬ (p ∧ ¬ q): THM
:) val L2 = p ⇒ q ⊢ ¬ (p ∧ ¬ q): THM
:) val T = ⌜p ∧ ¬ q ⇔ ¬ q ∧ p⌝: TERM
:) val commutative = ⊢ p ∧ ¬ q ⇔ ¬ q ∧ p: THM
:) val L3 = p ⇒ q ⊢ ¬ (¬ q ∧ p): THM
:) val L4 = p ⇒ q ⊢ q ∨ ¬ p: THM
:) val T = ⌜q ∨ ¬ p ⇔ ¬ q ⇒ ¬ p⌝: TERM
:) val conditional = ⊢ q ∨ ¬ p ⇔ ¬ q ⇒ ¬ p: THM
:) val it = p ⇒ q ⊢ ¬ q ⇒ ¬ p: THM
```

Problem 3.8

Show that $\sim(p \leftrightarrow q) \equiv (p \wedge \sim q) \vee (\sim p \wedge q)$

p	q	$(p \leftrightarrow q)$	$\sim(p \leftrightarrow q)$	$\sim q$	$(p \wedge \sim q)$	$\sim p$	$(\sim p \wedge q)$	$((p \wedge \sim q) \vee (\sim p \wedge q))$	$(\sim(p \leftrightarrow q) \leftrightarrow ((p \wedge \sim q) \vee (\sim p \wedge q)))$
F	F	T	F	T	F	T	F	F	T
F	T	F	T	F	F	T	T	T	T
T	F	F	T	T	T	F	F	T	T
T	T	T	F	F	F	F	F	F	T

Notice that ProofPower automatically applied biconditional rule

$$
\begin{aligned}
\sim(p \leftrightarrow q) &\equiv \sim[(p \rightarrow q) \wedge (q \rightarrow p)] && \text{(Biconditional)}\\
&\equiv [\sim(p \rightarrow q)] \vee [\sim(q \rightarrow p)] && \text{(De Morgan's)}\\
&\equiv (p \wedge \sim q) \vee (\sim p \wedge q) \blacksquare && \text{(Conditional)}
\end{aligned}
$$

SML

```
val L0 = asm_rule ⌜ ¬(p ⇔ q) ⌝;
val L1 = rewrite_rule [⇔_thm] L0;
val T = ⌜ ¬((p ⇒ q) ∧ (q ⇒ p)) ⇔ (¬(p ⇒ q)) ∨ (¬(q ⇒ p)) ⌝;
val demorgan = prove_rule [ ] T;
val L2 = rewrite_rule [demorgan] L1;
rewrite_rule [¬_⇒_thm] L2;
```

```
:) val L1 = ¬ (p ⇔ q) ⊢ ¬ ((p ⇒ q) ∧ (q ⇒ p)): THM
:) val T = ⌜¬ ((p ⇒ q) ∧ (q ⇒ p)) ⇔ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p)⌝: TERM
:) val demorgan = ⊢ ¬ ((p ⇒ q) ∧ (q ⇒ p)) ⇔ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p): THM
:) val L2 = ¬ (p ⇔ q) ⊢ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p): THM
:# val it = ¬ (p ⇔ q) ⊢ p ∧ ¬ q ∨ q ∧ ¬ p: THM
```

CS113 LAB 3 - Problem 3.11

Problem 3.11

Show using a chain of logical equivalences that $(p \to r) \land (q \to r) \equiv (p \lor q) \to r$

p	q	r	$(p \lor q)$	$((p \lor q) \to r)$	$(p \to r)$	$(q \to r)$	$((p \to r) \land (q \to r))$	$(((p \lor q) \to r) \leftrightarrow ((p \to r) \land (q \to r)))$
F	F	F	F	T	T	T	T	T
F	F	T	F	T	T	T	T	T
F	T	F	T	F	T	F	F	T
F	T	T	T	T	T	T	T	T
T	F	F	T	F	F	T	F	T
T	F	T	T	T	T	T	T	T
T	T	F	T	F	F	F	F	T
T	T	T	T	T	T	T	T	T

Notice that ProofPower automatically applied biconditional rule and conditional rule

$$
\begin{aligned}
(p \land q) \to r &\equiv [\sim (p \lor q)] \lor r && \text{(Conditional)}\\
&\equiv [(\sim p) \land (\sim q)] \lor r && \text{(De Morgan's)}\\
&\equiv [\sim p \lor r] \land [(\sim q)] \lor r && \text{(Distributive)}\\
&\equiv (p \to r) \land (q \to r) \blacksquare && \text{(Conditional)}
\end{aligned}
$$

SML

```
val L0 = asm_rule ⌜ ¬(p ⇔ q) ⌝;
val L1 = rewrite_rule [⇔_thm] L0;
val T = ⌜ ¬((p ⇒ q) ∧ (q ⇒ p)) ⇔ (¬(p ⇒ q)) ∨ (¬(q ⇒ p)) ⌝;
val demorgan = prove_rule [ ] T;
val L2 = rewrite_rule [demorgan] L1;
val L3 = rewrite_rule [¬_⇒_thm] L2;
```

```
:) val L1 = ¬ (p ⇔ q) ⊢ ¬ ((p ⇒ q) ∧ (q ⇒ p)): THM
:) val T = ⌜¬ ((p ⇒ q) ∧ (q ⇒ p)) ⇔ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p)⌝: TERM
:) val demorgan = ⊢ ¬ ((p ⇒ q) ∧ (q ⇒ p)) ⇔ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p): THM
:) val L2 = ¬ (p ⇔ q) ⊢ ¬ (p ⇒ q) ∨ ¬ (q ⇒ p): THM
:# val L3 = ¬ (p ⇔ q) ⊢ p ∧ ¬ q ∨ q ∧ ¬ p: THM
```

Problem 3.12

Show using a chain of logical equivalences that $p \leftrightarrow q \equiv (p \land q) \lor (\sim p \land \sim q)$

p	q	$(p \leftrightarrow q)$	$(p \land q)$	$\sim p$	$\sim q$	$(\sim p \land \sim q)$	$((p \land q) \lor (\sim p \land \sim q))$	$((p \leftrightarrow q) \leftrightarrow ((p \land q) \lor (\sim p \land \sim q)))$
F	F	T	F	T	T	T	T	T
F	T	F	F	T	F	F	F	T
T	F	F	F	F	T	F	F	T
T	T	T	T	F	F	F	T	T

Notice that ProofPower automatically applied biconditional rule and conditional rule

$$
\begin{aligned}
p \leftrightarrow q &\equiv (p \to q) \land (q \to p) && \text{(Biconditional)}\\
&\equiv (\sim p \lor q) \land (\sim q \lor p) && \text{(Conditional)}\\
&\equiv [\sim p \land (\sim q \lor p)] \lor [q \land (\sim q \lor p)] && \text{(Distributive OR)}\\
&\equiv [(\sim p \land \sim q) \lor (\sim p \land p)] \lor [(q \land \sim q) \lor (q \land p)] && \text{(Distributive AND)}\\
&\equiv [(\sim p \land \sim q) \lor c] \lor [c \lor (q \land p)] && \text{(Contradiction)}\\
&\equiv (\sim p \land \sim q) \lor (p \land q) && \text{(Identity)}\\
&\equiv (p \land q) \lor (\sim p \land \sim q) \ \blacksquare && \text{(Commutative)}
\end{aligned}
$$

SML

```
val L0 = asm_rule ⌜ (p ⇔ q) ⌝;
val L1 = rewrite_rule [⇔_thm] L0;
val L2 = rewrite_rule [⇒_thm] L1;
val T = ⌜ (¬p ∨ q) ∧ (¬q ∨ p) ⇔ (¬p ∧ (¬q ∨ p)) ∨ (q ∧ (¬q ∨ p)) ⌝;
val distributiveOR = prove_rule [ ] T;
val L3 = rewrite_rule [distributiveOR] L2;
val T = ⌜ (¬p ∧ (¬q ∨ p)) ∨ (q ∧ (¬q ∨ p)) ⇔ ((¬p ∧ ¬q) ∨ (¬p ∧ p)) ∨ ((q ∧ ¬q) ∨ (q ∧ p)) ⌝;
val distributiveAND = prove_rule [ ] T;
val L4 = rewrite_rule [distributiveAND] L3;
val T = ⌜ ((¬p ∧ ¬q) ∨ (¬p ∧ p)) ∨ ((q ∧ ¬q) ∨ (q ∧ p)) ⇔ ((¬p ∧ ¬q) ∨ F) ∨ (F ∨ (q ∧ p)) ⌝;
val contradiction = prove_rule [ ] T;
val L5 = rewrite_rule [contradiction] L4;
val T = ⌜ (¬p ∧ ¬q) ∨ (q ∧ p) ⇔ (p ∧ q) ∨ (¬p ∧ ¬q) ⌝;
val commutative = prove_rule [ ] T;
val L6 = rewrite_rule [commutative] L5;
```

```
:) val L1 = p ⇔ q ⊢ (p ⇒ q) ∧ (q ⇒ p): THM
:) val L2 = p ⇔ q ⊢ (¬ p ∨ q) ∧ (¬ q ∨ p): THM
:) val T = ⌜(¬ p ∨ q) ∧ (¬ q ∨ p) ⇔ ¬ p ∧ (¬ q ∨ p) ∨ q ∧ (¬ q ∨ p)⌝: TERM
:) val distributiveOR = ⊢ (¬ p ∨ q) ∧ (¬ q ∨ p) ⇔ ¬ p ∧ (¬ q ∨ p) ∨ q ∧ (¬ q ∨ p):
   THM
:) val L3 = p ⇔ q ⊢ ¬ p ∧ (¬ q ∨ p) ∨ q ∧ (¬ q ∨ p): THM
:) val T =
   ⌜¬ p ∧ (¬ q ∨ p) ∨ q ∧ (¬ q ∨ p) ⇔ (¬ p ∧ ¬ q ∨ ¬ p ∧ p) ∨ q ∧ ¬ q ∨ q ∧ p⌝:
   TERM
:) val distributiveAND =
   ⊢ ¬ p ∧ (¬ q ∨ p) ∨ q ∧ (¬ q ∨ p) ⇔ (¬ p ∧ ¬ q ∨ ¬ p ∧ p) ∨ q ∧ ¬ q ∨ q ∧ p:
   THM
:) val L4 = p ⇔ q ⊢ (¬ p ∧ ¬ q ∨ ¬ p ∧ p) ∨ q ∧ ¬ q ∨ q ∧ p: THM
:) val T = ⌜(¬ p ∧ ¬ q ∨ ¬ p ∧ p) ∨ q ∧ ¬ q ∨ q ∧ p ⇔ (¬ p ∧ ¬ q ∨ F) ∨ F ∨ q ∧ p⌝:
   TERM
:) val contradiction =
   ⊢ (¬ p ∧ ¬ q ∨ ¬ p ∧ p) ∨ q ∧ ¬ q ∨ q ∧ p ⇔ (¬ p ∧ ¬ q ∨ F) ∨ F ∨ q ∧ p: THM
:) val L5 = p ⇔ q ⊢ ¬ p ∧ ¬ q ∨ q ∧ p: THM
:) val T = ⌜¬ p ∧ ¬ q ∨ q ∧ p ⇔ p ∧ q ∨ ¬ p ∧ ¬ q⌝: TERM
:) val commutative = ⊢ ¬ p ∧ ¬ q ∨ q ∧ p ⇔ p ∧ q ∨ ¬ p ∧ ¬ q: THM
:) val L6 = p ⇔ q ⊢ p ∧ q ∨ ¬ p ∧ ¬ q: THM
```