Math 2001 Homework 04

Exercises: LaP 5.3. 1–7

Due Friday 22 February by 2pm

5.3.1. Show how to derive the proof-by-contradiction rule from the law of the excluded middle, using the other rules of natural deduction. In other words, assume you have a proof of $\perp$ from $\neg A$. Using $A \vee \neg A$ as a hypothesis, but *without* using the rule RAA, show how you can go on to derive A.

5.3.2. Give a natural deduction proof of $\neg(A \wedge B)$ from $\neg A \vee \neg B$. (You do not need to use proof by contradiction.)

5.3.3. Construct a natural deduction proof of $\neg A \vee \neg B$ from $\neg(A \wedge B)$. You can do it as follows:

1. First, prove $\neg B$, and hence $\neg A \vee \neg B$, from $\neg(A \wedge B)$ and A.
2. Use this to construct a proof of $\neg A$, and hence $\neg A \vee \neg B$, from $\neg(A \wedge B)$ and $\neg(\neg A \vee \neg B)$.
3. Use this to construct a proof of a contradiction from $\neg(A \wedge B)$ and $\neg(\neg A \vee \neg B)$.
4. Using proof by contradiction, this gives you a proof of $\neg A \vee \neg B$ from $\neg(A \wedge B)$.

5.3.4. Give a natural deduction proof of $\neg A \vee B$ from $A \to B$. You may use the *Law of the Excluded Middle* (LEM).

5.3.5. Put $(A \vee B) \wedge (C \vee D) \wedge (E \vee F)$ in disjunctive normal form, that is, write it as a big “or” of multiple “and” expressions.

5.3.6. Prove $\neg(A \wedge B) \to \neg A \vee \neg B$ by replacing the sorry’s below by proofs.

```
open classical
variables {A B C : Prop}

-- Prove ¬ (A   B) → ¬ A   ¬ B by replacing the sorry's below
-- by proofs.

lemma step1 (h  : ¬ (A   B)) (h  : A) : ¬ A   ¬ B :=
have ¬ B, from sorry,
show ¬ A   ¬ B, from or.inr this

lemma step2 (h  : ¬ (A   B)) (h  : ¬ (¬ A   ¬ B)) : false :=
```

```
have ¬ A, from
  assume : A,
  have ¬ A   ¬ B, from step1 h  ‹A›,
  show false, from h  this,
show false, from sorry

theorem step3 (h : ¬ (A   B)) : ¬ A   ¬ B :=
by_contradiction
  (assume h' : ¬ (¬ A   ¬ B),
    show false, from step2 h h')
```

5.3.7. Also do these:

```
open classical
variables {A B C : Prop}

example (h : ¬ B → ¬ A) : A → B :=
sorry

example (h : A → B) : ¬ A   B :=
sorry
```