Lab: The Propositional Calculus
CSC/MAT 208: Discrete Structures
Department of Computer Science
Department of Mathematics and Statistics
Grinnell College

Pairs

Here are the programmer pairs for today’s lab. Drivers are on the left.

- Marcel Champagne and Jong Hoon Bae
- David Chang and Bazil Mupisiri
- Adam Wesely and Elizabeth Zak
- Kathryn Yetter and Danielle Williams
- Michael Owusu and Tyler Williams
- Lindsey Byrne and Devin Dooley
- Ella Nicolson and Linda Oyolu
- Kevin Connors and Matt Jasaitis
- Dennis Chan and Samee Zahid
- Madeleine Goldman and Thu Nguyen
- Corey Simmonds and Adam Hudson
- Amanda Hinchman-Dominguez and Jacob Ekstrand
- Muhammad Hamza and Anita DeWitt

Julia Fay will take the place of anyone who is absent.

Exercises

Study the Form of the Conclusion

When proving the correctness of an argument in propositional calculus (that is, when showing that some Boolean expression can be inferred from others), you’ll often find that the main connective in the conclusion gives a strong hint about the overall structure of the proof. For instance, to obtain a conclusion that is a conjunction, one often uses conjunction introduction in the last step (as in the proof that conjunction is commutative, in today’s reading). If the conclusion is a negation, the last step is often justified by
reductio ad absurdum (as in the proof of the double negation theorem), and so on.

Exercise 0: Using this idea, prove that ‘¬(p ≡ ¬p)’ is a theorem. What will the justification of the final step be? What does that imply about the structure of the proof?

Proving Disjunctions

From the suggestion preceding exercise 0, you might think that the rules of disjunction introduction would be good ways to proving conclusions that are disjunctions. This works only if it is possible to infer one of the disjuncts from the available premises. Often, however, the disjunction can be inferred even if neither of the disjuncts can: the premises are sufficient to show that at least one of the disjuncts must hold, but are not sufficient to show which one it is.

Consequently, you should be prepared to try other rules as justifications for the last step in the proof of a disjunction. If one of the premises is also disjunctive in form, you might try disjunctive syllogism. Another stock pattern that is often encountered uses reductio ad absurdum as the penultimate step (getting a doubly-negated disjunction), then negation elimination at the end. The proof of the principle of the excluded middle in the reading has this form.

Exercise 1: Prove that, from the premiss ‘(p ∨ (q ∧ r))’, one can infer the conclusion ‘((p ∨ q) ∧ (p ∨ r))’.

Derived Inference Rules

Logically, forming the conjunction of any Boolean expression with ⊤ has no effect. Obviously, one can infer the Boolean expression itself from the conjunction, by conjunction elimination. It turns out that one can also infer the conjunction from the Boolean expression itself.

Exercise 2: Prove that p ⊢ (p ∧ ⊤).

Exercise 3: Replace the Boolean variable ‘p’ with the schematic letter ‘φ’ throughout the proof in exercise 2, converting it into a proof scheme that can be instantiated with any Boolean expression.

Exercise 4: Extend this proof scheme to show that every Boolean expression of the form ⊢(φ ≡ (φ ∧ ⊤)) ⊢ is a theorem.
The result you obtained in exercise 3, \( \vdash (\phi \land \top) \), is a derived inference rule. Since you’ve proved that the inference that it describes is correct, you can use any instance of this result as needed in subsequent proofs, making them shorter and clearer.

A theorem scheme, such as the one you proved in exercise 4, is like a “derived axiom” — a derived inference rule that doesn’t have any premisses. You can add theorems to your proofs as well, as if they were axioms.

**Lemmas**

If you persevered all the way to the end of the reading for today, and attempted the last exercise in the set at the end, you discovered that each of DeMorgan’s laws requires a series of seven or eight hypothetical arguments, preparing the groundwork for a final one-step proof.

To simplify the presentation of a long proof like this one, it is often broken up into stages called lemmas. A lemma is a subproof that plays some key role in a larger proof. What is proven in a lemma is usually somewhat interesting or useful in its own right, although it may be a rather specialized result; what makes it a lemma, though, is its relation to the long proof into which it fits.

Breaking a long proof into lemmas is quite similar to modularizing an overlong procedure or method definition by defining helper procedures — it’s not formally required, but it imparts a structure to the proof that makes it easier for both the author and the reader to grasp what is happening.

**Exercise 5:** Prove that ‘\((p \land q) \vdash (p \equiv q)\)’ is a correct inference.

**Exercise 6:** Prove that ‘\((\neg p \land \neg q) \vdash (p \equiv q)\)’ is a correct inference.

**Exercise 7:** Using the results of exercises 5 and 6 as lemmas, prove that ‘\(((p \land q) \lor (\neg p \land \neg q)) \vdash (p \equiv q)\)’ is a correct inference.

**Exercise 8:** Prove that ‘\((p \equiv q), p \vdash (p \land q)\)’ is a correct inference.

**Exercise 9:** Prove that ‘\((p \equiv q), \neg p \vdash (\neg p \land \neg q)\)’ is a correct inference.

**Exercise 10:** Using the results of exercises 8 and 9 as lemmas, prove that ‘\((p \equiv q) \vdash ((p \land q) \lor (\neg p \land \neg q))\)’ is a correct inference. (Hint: Also use the principle of the excluded middle — remember, you can add any theorem that has already been proven as a step in a proof.)

**Exercise 11:** Using the results of exercises 7 and 10 as lemmas, prove that ‘\(((p \equiv q) \equiv ((p \land q) \lor (\neg p \land \neg q)))\)’ is a theorem.
Exercise 12: The theorem that you proved in exercise 11 is an instance of the analysis of equivalence, $\Gamma \vdash ((\phi \equiv \psi) \equiv ((\phi \land \psi) \lor (\neg \phi \land \neg \psi)))$. Can you generalize your proof into a proof scheme that covers all instances of the analysis of equivalence? If so, do you have to generalize all of the lemmas as well, using the same schematic letters?