COMP 170 Guide to proof writing

COMP 170 Staff

Introduction

Proof writing is hard work and it will be hard work until you have written and read many proofs. For those new to proof writing and computation theory, knowing what a good proof looks like is a non-trivial task. To help you with this, we present a set of heuristics, requirements and tips to be a successful proof writer in this Computation Theory course.

Restate the problem

Restate the problem to your reader. A Proof that is out of context is also one that is difficult to understand. So let your reader know what problem you are trying to solve!

Proof idea and type

State what the big idea is for your proof. How would you explain the overall idea of your proof to a stranger? Writing a proof idea also allows you to explore the assumptions you can play with. In your proof idea, you could:

- State the proof type. Is this proof a reduction? A proof by contradiction? Or something else.
- Write down the definitions that you think you will need.
- Write down the constructions you can assume from the hypothesis.

The proof idea is a place where we can reward you for partial credit.
Use clear, focused, prose

Yes! Write in prose! A good proof is a clear convincing a story. The best proofs are written with full sentences and use clear, focused paragraphs to organize the necessary claims and ideas.

Very good proofs avoid symbols. Some symbols you should avoid in sentences are $\exists, \forall, \Rightarrow$ and $\therefore$. Use words and phrases to achieve what you mean by the symbols. For example, you could instead use, “there exists”, “for all”, “this implies that”, “hence”, “since”.

Symbols to indicate set membership and inclusion such as, $\in, \subseteq, \subset$ are ok.

In sample solutions, you will see a distinct lack of symbols. A sentence that is made up of mostly symbols is the computation theorist’s run-on sentence. They are hard to read and hard to understand.

Define Machines and other constructions

Explicitly define machines and constructions. By explicitly, we mean to define turing machines using psuedocode. Don’t give room for the reader to question what you have defined. Own it!

Separate your machine definition and psuedocode using new lines and tabs and do not violate the offside rule (Those of you who have taken 105 will be familiar with this). If you refer to a specific behavior later in your proof, this behavior should be represented. These requirements are similar to code formatting rules in COMP 11/15/40; the same expectatations apply to psuedo code.

Keep case analyses and constructions separate. When constructing a function, a TM, or some other object, only define its behavior. This keeps constructions concise, gives your proof structure, and is a heuristic you should follow to write a clear proof.

When defining a turing machine, avoid saying what what happens if a machine loops. $HALT$ is undecidable so if you are trying to define a behavior that depends on a loop, you are probably doing something wrong.
About case analyses

Constructions of functions, reductions and machines are not meaningful without case analyses. Stating definition of the machine/reduction is a construction and not a proof. Use case analysis to tell us why your construction is a reduction, decider or recognizer. Lead your reader to the conclusion they must arrive at! Leave no stone unturned, and leave no doubt in your reader’s mind.

Suppose that you are trying to show that a language $A$ many-one-reduces to another language $B$. Once you have constructed a function you believe to be a many one reduction, you can use the following templated sentences to convince that $f$ is indeed a many-one reduction:

1. “If $x$ is in $A$, then ⟨a thorough step-by-step explanation here⟩, Therefore $f(x)$ must be in $B$.”
2. “If $x$ is not in $A$, then ⟨a thorough step-by-step explanation here⟩. Hence $f(x)$ is not in $B$.”

Tieing everything together

Finish your proof with a bang! State the final conclusion and remind your reader what you have worked so hard to prove. Leave no doubt in their mind.

(For flare, you can finish with “$\Box$” on the bottom right of the proof!)

Things to avoid - Basket of deplorables

- Do not use ”I”, use ”We” in your proofs.
- Do not use words like ”trivially” and ”obviously”. Do not insult your reader’s intelligence; don’t badger them.