Your performance on the first assignment appears quite satisfactory, mod the first point below.

1. Write the function which determines equivalence up to bound variables for your datatype of the first assignment. Package everything in Windows so I can run it (thanks, I know you are already working on this).

2. Re 1202 in the Andrews book: Give a syntactical proof that \( \mathcal{P} \) is consistent with respect to negation: prove by induction on the structure of formulas that you cannot prove both \( A \) and \( \neg A \). Bear in mind that you might be asked to extend this proof to full first order logic later. This proof will involve talking about syntax and proof and will not involve semantics at all. I don’t know how hard it will be; we may end up working on this in parallel.

3. Read the Andrews book, up to section 21, carefully. I’m quite interested in the Substitutivity of Implication rule; you might want to think about implementation of recognition of positive and negative instances of subformulas as a target.

4. Extend your formal language to allow introduction of defined propositional connectives. Develop a prover for your datatype which will handle the basic rules of propositional logic (\( \mathcal{P} \)) and ideally also the basic rules of \( \mathcal{F} \). You can fiddle with the style (you can work with proof lines or with sequents) but be aware I may ask you to use it to do proofs presented in Andrews’s book so you should be able to present them (although they might not look exactly the same). In your prover you should certainly have rules for negation and disjunction; you should...
at least be able to unpack the defined connectives to negation and disjunction; it would be wonderful if you could formally derive rules for the defined propositional connectives and then use the derived rules. (of course introducing the universal quantifier as a primitive or a defined notion (better) is also a good idea).

No explicit due date: this takes as long as it takes. I imagine the proof of 1202 will take rather less time than the programming assignment, and will of course be replaced with another paper and pencil exercise. On any problem I set you you can come and ask me for help: remember that my experience teaching this content is very limited...