Journal Report 20
3/2/20-3/6/20
Rahul Mittal
Computer Systems Research Lab
Period 2, White

Daily Log

Thursday March 5

Go back through Andrew Haven, look at choose_equal, the proof of choose_equal would be useful, google Andrew Haven, go to his advisor's website, look up excluded middle in Coq, find list of recommended axioms, decide to just axiomatize excluded middle, copy-pasting the sections on functions and inverses into CoqIDE, got all of that to work as well as the excluded middle axiom and the definitions of pinv (inverse for a T -¿ Prop) and disjoint_bij, got exist_eq axiom working, finally proved disjoint_bij_is_function, proved disjoint_bij_is_injective, proved disjoint_bij_is_surjective, proved disjoint_bij_is_total, and prove disjoint_bij_is_bijective, need to write a script to parse this thesis because I'm doing too much manual work, finished script, time to test, add special case for brackets to remove some spacing, made change to add underscores in names of definitions/lemmas/etc., script seems done, test on a few sections of Haven's code, make change to accommodate Variables.

Sunday March 8

Start thinking about what needs to be included in combinatorics foundation, for Colorado 1 all that's needed in addition to Haven's work is complement_is_bijection, for Colorado 2, what is ultimately needed is that there is a bijection between subsets of size k of a set of size n that include s, and subsets of size $k-1$ of a set of size $n-1$ (and a similar statement for subsets not including s), last part of proof (every subset either contains s or does not, so amount = amount include s + amount not include s) is conveniently covered by disjoint_bij_is_bijective, which I already proved in Coq, Colorado 3 requires representation of subset of size k paired with subset of size m of the subset, could be represented in Coq as subset of size k, subset of size m, and proof that latter is subset of former (as in latter proposition implies former proposition), then formulation in second part of proof can be described as pair of subsets, one of size m, of size $k-m$, that are disjoint, need to show bijection between the two formulations, Haven already has multiplication principle so that covers all of it, Colorado 4, right side (proving amount of subsets = 2^n) uses idea of creating sequences of choices, can be represented as lists in Coq, can use boolean as type of list element, need to show that amount of lists w/ elements of finite set is $|set|^length$, later on will need idea of creating list from "for each element, make this..." type statements, then just need bijection between boolean sequences and lists, left side need to show bijection between power set and union of set of subsets of each size, for Colorado 5, right side relies on amount of sets = 2^n and there is exactly one empty set, left side needs all nonempty sets of integers $1, \ldots, n$ to have maximum element k, then show bijection between sets of maximum element k and boolean lists of length $k-1$, as well something like disjoint_bij for more than two sets.

Timeline

Date	Goal	Met
February 24	Start work on combinatorics foundation in Coq	Read most of Coq tutorial
March 2	Finish Coq tutorial, go back to Haven's thesis on using Coq for combinatorics, express Colorado 1 in Coq	Yes, yes, started
March 9	Complete Coq combinatorics foundation (state/prove necessary foundational results not already shown by Haven)	Decided what results needed for Colorado proofs, disjoint_bij_is_bijective, finished script to quickly convert Haven's work to be usable by me
March 16	Complete combinatorics foundation (including tested proofs), start code (in Kotlin) to generate Coq output for basic components of a proof	
March 23	Finish code (in Kotlin) to generate Coq output for basic components of a proof	

Reflection

I made a kind of unfortunate discovery this week. Andrew Haven's thesis, while it contains all the theorems he uses in order to solve his combinatorial problem, does not contain the proofs for any of those foundational results, presumably because they would be too long. I assumed that these proofs would be available in some kind of appendix, but unfortunately, Haven's thesis has no appendix. I then proceeded to do some Googling, trying to see if I could find a GitHub, and looked at his advisor's website, but couldn't find anything. Since I really don't want to waste more time looking for his proofs, I decided to, at least for now, axiomatize all of his theorems and lemmas, based on the assumption that they do exist somewhere in the world. It's very much not ideal, but hopefully it'll be possible later on to get the proofs and add them to my output.

After that, I decided to make at least part of my unfinished proof that the complement is a bijection actually work in CoqIDE (i. e. verify the proof). Everything actually dealing with the complement itself relied on basically all of Haven's foundation, so I decided to start with verifying disjoint_bij_is_bijective, since that only required his section on Relations. I first had to copy-paste that section from his thesis into CoqIDE, and apparently copy-pasting that kind of formatting text from a PDF into a plaintext file works really poorly, so I had to do a lot of fixing. I also had to replace a lot of symbols that Haven used (like $\forall$) that my keyboard doesn't have and I would need to somehow add definitions for anyway. This wasn't too bad for just that section, but later on when I was trying to do the rest of his thesis so I could complete my proof, I realized that I was wasting time, so I wrote a quick script in Kotlin to format his code for me.

Once I had that done, I copy-pasted in my proof of disjoint_bij_is_function and quickly realized that it was going to take some work to make it actually compile. Most of the problem was random type issues, not supplying the type as an argument in certain places, etc., but there were also a few places where certain tactics didn't work the way that I expected them to, so fixing all of that took some time, but I was finally able to prove disjoint_bij_is_bijective.

For comparison, here's the first half of my previous proof of disjoint_bij_is_bijective:

```
Theorem disjoint_bij_is_function :
        forall (T : Type) (s1 : T -> Prop), function (disjoint_bij T s1).
    intros T.
    intros s1.
    intros x y y'.
    intros proof_f_x_y proof_f_x_y'.
    case y.
        intros ea.
        destruct ea as [a proof_a_s1].
        simpl in proof_f_x_y.
        case y'.
            intros ea'.
            destruct ea' as [a' proof_a_s1'].
            simpl in proof_f_x_y'.
            rewrite (thm_eq_trans (thm_eq_sym proof_f_x_y') proof_f_x_y)).
            exact (exist_eq T s1 a proof_a_s1 proof_a_s2).

            intros ea'.
            destruct ea' as [a' proof_not_a_s1'].
            simpl in proof_f_x_y'.
            rewrite (thm_eq_trans (thm_eq_sym proof_f_x_y') proof_f_x_y)).
            case (proof_not_a_s1' proof_a_s1).
```

And here's the first half of the one that actually works:

```
Theorem disjoint_bij_is_function :
        forall (T : Type) (s1 : T -> Prop),
        function T (sig s1 + sig (pinv s1)) (disjoint_bij s1).
Proof.
    intros T.
    intros s1.
    intros x y y'.
    intros proof_f_x_y proof_f_x_y'.
    destruct y as [ea | ea].
        destruct ea as [a proof_a_s1].
        simpl in proof_f_x_y.
        destruct y' as [ea' | ea'].
            destruct ea' as [a' proof_a_s1'].
            simpl in proof_f_x_y'.
            rewrite (exist_eq T s1 a' a (thm_eq_trans T a' x a (thm_eq_sym T x a' p
            exact (eq_refl (inl (exist s1 a proof_a_s1))).

            destruct ea' as [a' proof_not_a_s1'].
            simpl in proof_f_x_y'.
            pose (proof_not_a_s1'' := proof_not_a_s1').
            rewrite (thm_eq_trans T a' x a (thm_eq_sym T x a' proof_f_x_y') proof_f
            case (proof_not_a_s1'' proof_a_s1).
```

Of course, in order to prove it, I needed a fix to the issue with the excluded middle not necessarily being true in Coq. I initially thought that axiomatizing the excluded middle would cause issues, but I Googled it and found the excluded middle in a list of axioms that you might need to include in Coq, so I just axiomatized it because apparently it's fine.

The last thing was looking through all the Colorado proofs and deciding what the foundation needed to include so that they could be proved. You can see specifically what I thought in the log. What I'm concerned about is that I'm tailoring what's being included in the foundation too specifically to the Colorado proofs. Taking Colorado 2 for example, I said that "what is ultimately needed is that there is a bijection between subsets of size k of a set of size n that include s, and subsets of size $k1$ of a set of size $n1$", but this doesn't really seem like a foundational result to me. If I don't use this as a foundational result, then I have to find some foundational results to use which my code can then use to prove this result itself. I'm sure that such results do exist, but I have to think more about exactly which results they should be to be most generally applicable while still clearly able to be used to prove this.