Journal Report 9
11/11/19-11/15/19
Rahul Mittal
Computer Systems Research Lab
Period 2, White

Daily Log

Monday November 11

Add HangingAmount subclass of Quantity, making major change to structure of MathThings, now, instead of fields of MathThings being MathThings, fields of MathThings are Parsed¡MathThing¿s, in order to retain information about where they came from, actually not doing that, instead having ParsedStatement subclass of Statement etc. added parsed function for easy conversion, when clause exhaustiveness is weird, getting strange type errors with this.

Tuesday November 12

The strange errors were because I made the argument of parsed have to be a subclass of MathObject rather than MathThing (maybe I should have chosen better names), also now realize that having one DFS for each of Statement, Quantity, and MathObject makes it really annoying to modify stuff and will probably lead to weird bugs, so joining everything into one DFS, ok finished that, finished testing all of the proof part of my sample proof.

Thursday November 14

Very conflicted about what templates to add to be able to represent the theorem part, create Bounded and Infinity subclasses of Quantity, add templates for ’at most %qq%’ and ’at least %qq%’, add ’each %xn1% contains %xn2%; template.

Timeline

Date	Goal	Met
October 28	Get detailing function to work on sample proof + first two Colorado proofs	No, haven't finished sample proof.
November 11	Get simplified parser to work on sample proof + first two Colorado proofs	No, but parser itself seems to work so far.
November 18	Make sure simplified parser works on all Colorado proofs	No, focused on restructuring code
November 25	Integrate actual formulae (as opposed to placeholders) into output as Quantity objects, make sure parser works on first 2 Colorado proofs	
November 25	Make sure parser works on rest of Colorado proofs, list repetitions of same MathObject	
Winter Goal	Take in as input a combinatorial proof, with no modifications other than expressing formulae with my formula syntax, and output tthe claim and all steps of the proof as instances of Statement, and an additional data structure representing relations between Statements in thet proof.	

Reflection

I kind of didn't end up fully working toward what my goal for this week was, but that's okay because what I did work on will be important for later.

The motivation for the first thing I did was that later on, when I'm trying to look for phrases that are actually references to the same thing, I'm going to want to be able to compare the actual words, not just the MathThing structure that the parser outputs. Because of this, I need to have, within those MathThings, a reference to the part of the input that that MathThing was parsed from. I originally intended to make a new class, Parsed¡S¿, that would contain a MathThing as well as that contextual information, and then have other MathThings have their fields be Parsed¡S¿s, but then realized that not all MathThings come directly from the parser so it wouldn't make sense for those ones to be wrapped in a Parsed¡S¿. My final solution was just to make a subclass of each of Statement, Quantity, and MathObject that would be a wrapper for things of its own type, which seems a bit weird in the way that works to me but it works.

The second thing I did was just to collapse the three DFSs I had into one, for mostly obvious reasons (readability, changeability, lessbugsfrommakingdifferentchangestoeachoneability, etc.). I originally wrote three separate DFSs because I thought of it as a separate search for each type of MathThing and couldn't figure out how to make the DFS code generic to where I could just somehow call it for each type. Now I realized that I could just literally put everything in the same DFS and it works pretty much fine.

I modified the way the goals are going to work for the next two weeks. I still want to make sure that all of the Colorado proofs work with the parser, but I don't want to put off integrating the formula stuff because it should be fairly straightforward. What I did, then, is to have the integration of the formula stuff first, then split the Colorado proofs across the next two weeks, and also start with the next step in that second week from now.

I was intending to just show the output of my parser on the last part of my sample proof before all of the modifications I made this week, but since I had to rerun the parser right now to actually get that output, it now has the contextual stuff as well, which is good.

Input:

The total amount is g.

Output:

```
ParsedStatement(
    statement=Equals(
        q1=ParsedQuantity(
            quantity=HangingAmount(unit=kotlin.Unit),
            context=ParseContext(from=2, to=3)
        ),
        q2=Formula(placeholder=g)),
    context=ParseContext(from=2, to=5)
)
```

(Note: the HangingAmount conatins a Unit as a member field - HangingAmount isn't supposed to have any fields, but I'm having all of these classes be data classes since (they are, and) Kotlin will autogenerate toString, equals, etc. for data classes. For some stupid reason, a data class has to have a nonzero amount of fields, so I added a useless field to make it work.)

In terms of how I'm progressing towards my winter goal, based on my schedule I should have all of December to work on finding the relations between statements in the proof, which should be enough time to get to a good place in terms of the relations that my program can identify and output.