Journal Report 8

10/28/19-11/8/19
Rahul Mittal
Computer Systems Research Lab
Period 2, White

Daily Log

Monday October 28

Divide inner loop into 3 run statements for Statement, Quantity, MathObject (for organization), add if statement in Quantity section for assigning dp value if it's a formula, add similar in Statement section for if it's 'g is true' for formula g, add thing in MathObject section for if it's an arbitrary math object word (right now if it's in the set of arbitrary math object words).

Tuesday October 29

Start writing actual DFS, add if statement checking if node's position in tokens or in syntax is at the end, if so if both are at end then set dp if minimal, if neither at end push node that skips next token, then add when statement, if next element of syntax is Token, check against next token and push node at next position if same, else if next element is SyntaxParameter, loop through right endpoint of parameter and push nodes corresponding to using dp values, start presentations.

Thursday October 31

Work on OMO problems for half an hour, presentations.

Wednesday November 6

Compile jar, scp jar to syslab, ssh to kingdedede, doesn't work, ssh to luigi, doesn't work, ssh to chopin, works, run the jar, input "we can choose b urns which will contain a ball in c ways", ExceptionInInitializerError from illegalargexception: bad, fix the thing by making it uppercase, auto-lowercase input text, recompiling… uploaded, tested, still doesn't work, find fix, will test next class.

Thursday November 7

Upload jar, ssh to hartmanis since it's right in front of me, doesn't work, add debug statements, recompile jar, upload jar, ssh to hartmanis, now it works... now try inputting next clause "the balls can be matched up with the urns in d ways", works, ok generally seems to work, add statement template "

Timeline

Date	Goal	Met
October 21	Get detailing function to work with two templates	Yes, one statement template and one math object template.
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	
November 25	Integrate actual formulae (as opposed to placeholders) into output as Quantity objects, list repetitions of same MathObject	

Reflection

Before anything else, I treated the goal for the previous next Monday as now being the goal for this whole two week period (and pushed back the following goal by a week as well) since I didn't know before about the presentations (and the two week work period ended up being about equivalent to a normal week).

Even then, this ended up being a kind of unproductive week, especially on the first Thursday when I did literally nothing. The good thing was that my main task, which was to finish the parser, was laid out fairly cleanly for me, so I finished that at about the right time.

I think the main problem I have with this parser is that I'm not sure how to test it; obviously, I'm going to make sure it works with the Colorado proofs, but that isn't necessarily going to give it any special cases. Ultimately, if no issues with the parser pop up, I'll probably have to assume that it works so I can move on with the rest of my project.

The first input I tested the parser against was "we can choose b urns which will contain a ball in c ways", which gives the following output:

```
Equals(
    q1=Amount(
        obj=Choice(
            noun=SubjectToCondition(
                obj=ArbitraryMathObject(noun=urns), condition=Contains(elem=Arbitra
            )
        )
    ),
    q2=Formula(placeholder=c)
)
```

The second input was "the balls can be matched up with the urns in d ways", which gives:

```
Equals(
    q1=Amount(
        obj=Matching(
```

```
            n1=ArbitraryMathObject(noun=balls),
            n2=ArbitraryMathObject(noun=urns)
        )
    ),
    q2=Formula(placeholder=d)
)
```