Journal Report 11

11/25/19-12/6/19
Rahul Mittal
Computer Systems Research Lab
Period 2, White

Daily Log

Tuesday November 26

Do the Thanksgiving GitHub assignment, make journals and research_resources folders, start putting journals in, journals in, write README, create Varying subclass of SyntaxElement, change ParsePosition to having constructor and remaining syntax, modified dfs code to support Varying.

Monday December 2

Added Not subclass of SyntaxElement, implemented Not in dfs, now time to implement parsing syntax that has Varying and Not, first writing tokenize function, now writing stack based syntax parsing function called parseSyntax (wow!), finished parseSyntax, testing, done testing.

Tuesday December 3

Added intent stuff: added dpIntent in dfs, added INTENT_TEMPLATES, put templates from INTENT_TEMPLATES into stack, put results into dpIntent, changed answer to statementRes and intentRes, changed return type to Pair¡Statement, Intent¿, changed to update both statementRes and intentRes, changed return type to Pair¡ParseResult¡Statement¿?, ParseResult¡Intent¿?¿, updated return, reformatted some code to make changes easier, added Intent class (not subclass of MathThing) sometime ago, now looking at Colorado 2, added Show subclass of Intent, added ”we will show that” intent template (intemplate?), added Determine subclass of Intent, added ”we will count” intemplate, added ”how many xn include xn” quantity template, getting rid of attribute stuff, changed SubjectToCondition to having a XN $\rightarrow$ ST field, changed Contains to being subclass of Statement, made similar change to Each, removed Satisfies, fixing statement templates now, done with changes to all templates (not just statements), added ”how many xn do not include xn” quantity template, added newline, now doing file reorganization, moved BetterParser into better_parser package, created SyntaxTemplate file, created BetterSyntax file, created TemplateParameters file, changing Statement, Quantity, MathObject from sealed classes to open classes, also changing Intent to open class, created ParseContext file, removing MathThing restriction in parsed function, made exception message in parsed function more appropriate, changed name of StuffyStuff to MathThing, Removed 1 import.

Sunday December 8

Starting on proof 3, adding ”xn of qq xn” math object template, also adding some words, actually just one word, also actually instead adding ”ways to choose xn from xn” template, also added ChoiceFrom subclass of MathObject, added Qualified subclass of MathObject, added ”xn where st” math object template, added Element subclass of MathObject, added ”qq of xn” template, added ObjectEquals subclass of Statement, ”added xn forms xn” statement template, added ”there are qq ways to do this” statement template, added MakeChoice subclass of Intent, added

”we choose from” intemplate, change to ChooseFor and ”we choose from xn to form”, added ”choose qq of xn to form xn” template, modified ”elements” in remainder template to ”(elements—portion)”, added ”body” to list of words, added Part subclass of MathObject, added ”part of” template, added Size subclass of Quantity, added convenience of using * to denote possibility of s (ex. ”size*” = ”(size—sizes)”), added ”size* of xn” quantity template, added Subset subclass of MathObject, changed ”qq is qq” to ”qq (is—are) qq”, skipping proof 4 because it’s stupid, added Largest subclass of MathObject, added ”largest xn” math object template.

Timeline

Date	Goal	Met
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	Yes, but need more functionality to better represent Colorado proof 2
December 9	Make sure parser works on rest of Colorado proofs, parse statements of intent, add optional functionality to syntax, add thing in syntax for ”does not”	Mostly - proof 4 unfinished, proof 5 unfinished but straightforward
December 6	Finish Colorado proofs, prepare code to accept whole proof as input for demonstration	
Winter Goal	Take in as input a combinatorial proof, with no modifications other than expressing formulae with my formula syntax, and output the claim and all steps of the proof as instances of Statement.	

Reflection

Starting with the things other than the Colorado proofs, implementing all of it was fairly straightforward and actually easier than I expected.

For the optional functionality and ”does not” syntax element, the expected easy part was (after adding the classes) implementing their actual functionality in the actual dfs, and it was easy.

The part that I expected to be more difficult was implementing the function to convert a syntax string like ”the [pie [hi]] %xn3% lop” to a List¡SyntaxElement¿; I had just been splitting the string and converting each token to a SyntaxElement individually, but with parentheses and brackets being important in the syntax now, I needed to use a different approach. Fortunately, the stack-based approach I ended up using turned out to be not very difficult to write, if a bit clunky in the way I wrote it.

Going into the Colorado proofs, Colorado proof 3 took a while to finish, but I did manage to get it to parse into something satisfactory. I didn’t do much on proof 5 because proof 4 came before it, but it looked fairly straightforward to implement.

Proof 4, however, I found to be problematic because of this sentence:

When creating a subset of S, for each element of S, there are two options: to include it or not to include it.

I really have no idea how to properly represent this using my MathThing structures without it being an overly specific construction that wouldn’t be applicable to proofs that weren’t written for extremely similar situations in an extremely similar way. This is clearly a problem regardless of what I do, because this makes it clear that there are other proofs which have similarly specific wording that I don't see how to parse and represent.

I could limit the scope of my goal and say that those proofs just won't be able to be parsed, but that's clearly not ideal, and I don't think the reflection is the place to make important decisions that I'm not confident about (I guess there is no place to make important decisions that I'm not confident about).

As for my winter goal, as you can see on my timeline, I took out the part about represent relations between Statements, because I don't see myself reasonably getting to a good place with that in a single week, and I both want to finish the Colorado proofs and put the current parser into a state better for the demonstrations we're doing in the last week before break.