Journal Report 4
9/23/19-9/27/19
Rahul Mittal
Computer Systems Research Lab
Period 2, White

Daily Log

Monday September 23

Added case for ’does not’ being recognized as a potential math verb, added case for either in the case of ’subject either verb1 or verb2’ in order to account for last sentence of Colorado proof 2, removed dependency graph printing, modified constituency tree printing, proof 2 works now, transcribed proof 3

Tuesday September 24

Changed from replacing formulae of the form ’x = y’ with ’x equals y’ to with ’h is true’ to simplify replacement of more complicated things like ’x = y = z’ or ’x ¿= y’ etc., which also involved adding a separate isPlaceholder function, added case where having ’by’ indicates a deduction, proof 3 is done. Colorado proof 4 is rather problematic, majorly simplified the body of the identifyActions function, including moving the case of ’can’ into the isMathVerbActingOnMathNoun, all of the ’be’ cases are now grouped together, ’to be participle’ now requires math subject, saved a backup of the previous code, ’we make this choice n times’ is very annoying.

Thursday September 26

Tested variations of ’blah times’, seems to only be a problem when blah is a placeholder letter, whereas ’3 times’ for example yields a nicer constituency tree, ignoring it for now because I’m not willing to make changes to the code that would fix the case with placeholders when numbers work fine (maybe I can use numbers as placeholders instead of letters?), added case for ’verb preposition (ex. from) quantity’, proof 4 checked off, time for proof 5. ’b’ is being weird with the parsing, using ’g’ instead, added case so that cardinal numbers are recognized as math nouns, added case accounting for when one subject does multiple verbs (like ’subject v1 and v2’, or ’subject v1, v2, and v3’), added case for ’must’, made it so that ’therefore’ in any ancestor of the VP node is recognized to make it a deduction, not sure about that change, might fix it later, made a backup of the code.

Timeline

Date	Goal	Met
September 16	Run Stanford parser on more proofs, write a function trying to separate sentences into clauses with different actions	No, abandoning that idea completely
September 23	Finish prototype of function identifying actions and test it on more sample proofs to see what I didn't account for	Yes, almost done with making it work for Colorado proof 2
September 30	Revise function identifying actions and continue process to the point that 95% of random proofs on the Internet produce the correct output	No, only finished Colorado proofs
October 7	Modify function identifying actions to include information about statements made, meaning what verb used, what nouns acting/being acted on	
October 14	Created Haskell DSL for converting formulae into JSON format	

Reflection

This week made it clear that at the pace that I'm going at, there's no point continuing with just identifying actions. By the end of this week, I managed to make it so that the Colorado proofs were correctly classified (actually not true, I finished the last bit of proof 5 today on Monday), which is fine, but that's really slow for how little information I'm getting from the code that I'm producing, especially when, as I talked about in last week's reflection, it wouldn't be much more work to modify the code to produce much more detailed output, which was already my plan for this coming week, so that is how it's going to stay.

Based on how my work on the function has gone so far, what I think I can accomplish with this approach is to produce output that maps out the structure of statements being made (which can be definitionclaims, premises, deductions as already classified by my code) as much as possible entirely ignoring references - so, for example, if we consider the sentence ”Therefore, there are g such subsets.”, we can get a structure saying something like Amount(Subset(such)) = g, where the 'such' is a hole or loose end that we need to fix by relating different sentences or different parts of sentences to each other.

As an example of what I did last week, here is Colorado proof 3 (printed out separately because I didn't want to rewrite it in LaTeX):

And here is its rewritten form with the identified actions below each sentence:
For all h, g is true.

```
PREMISE: (VP (VBZ is) (ADJP (JJ true)))
```

QUESTION: We will show that both sides of the equation count the number of ways to choose a committee of k students from a student body of n students, where, in addition, a subcomittee of m of the k students form the executive committee.

```
No actions
```

LEFT: We will describe the counting process.

```
No actions
```

First, we choose k students from the student body of n students, to form the committee.

```
No actions
```

There are h ways to do this.

```
PREMISE: (VP (VBP are) (NP (NP (NN h) (NNS ways)) (SBAR (S (VP (TO to) (VP (VB do)
```

Then we choose m students from among those k to form the subcommittee.

```
No actions
```

There are j ways to do this.

```
PREMISE: (VP (VBP are) (NP (NP (NN j) (NNS ways)) (SBAR (S (VP (TO to) (VP (VB do)
```

By the multiplication principle, the left hand side counts the desired quantity.

```
DEDUCTION: (VP (VBZ counts) (NP (DT the) (JJ desired) (NN quantity)))
```

RIGHT: We will describe the counting process.

```
No actions
```

First, we choose m students from the student body of n students, to form the executive committee.

```
No actions
```

There are g ways to do this.

```
PREMISE: (VP (VBP are) (NP (NP (NN g) (NNS ways)) (SBAR (S (VP (TO to) (VP (VB do)
```

Then we choose l of the remaining portion of the student body (which consists of h students), to form the non-executive part of the committee.

```
No actions
```

There are b ways to do this.

```
PREMISE: (VP (VBP are) (NP (NP (NN b) (NNS ways)) (SBAR (S (VP (TO to) (VP (VB do)
```

By the multiplication principle, the right hand side counts the desired quantity.

```
DEDUCTION: (VP (VBZ counts) (NP (DT the) (JJ desired) (NN quantity)))
```