Journal Report 10

11/18/19-11/25/19
Bryan Lu
Computer Systems Research Lab
Period 2, White

Daily Log

Monday, November 18

Fixed github issues and successfully pushed all work and code to TJ CSL github page.

Tuesday, November 19

Finished assigning relations to half of my test cases. I added boolean relations to `finalrelations.txt` that check whether or not points or circles were special points/circles with respect to a given triangle, and relations that check for concyclicity of points.

Thursday, November 21

Finished assigning relations to 90% of my test cases, with some uncertainty as to how to deal with implicit relations and implicit objects present in a sentence that don't have explicit variables assigned to them.

Timeline

Date	Goal	Met
11/4	Create the log-linear classifier/learning algorithm and the training data, and begin testing.	I've started, but I've realized this is a rather ambitious task because I still need to finish creating my training data.
11/11	Complete a set of about 40 problems to serve as my training data set, with the correct relations.	I have the problems, but putting in the correct relations for these problems is a lot of work.
11/18	Build the model with `scikit`, tweaking previous steps as needed, and finish the necessary test cases.	I didn't have any time this week to start building the actual model, but my test cases are now more or less where I want them to be.
11/25	Build, test, and train the logistic model, given the test data and the features computed.	N/A
12/2	Refine logistic learning model with extra methods to try to increase accuracy, and begin extracting the most likely literals.	N/A
Winter Goal	Be able to output a set of possible literals (statements) based on detected relations in the problem.	N/A

Reflection

I don't have very much to show this week for my work because most of it was menial work that had to get done at some point. Attached is a screenshot of what my text file of relations contains as of right now – each problem is currently formatted so that it fits nicely on the page, with the associated relations that describe the problem below each problem.

```
Given a scalene triangle ABC. Its incircle touches BC, AC, AB at D, E, F respectvely. Let L, M, N be the reflections
of D over EF, of E over FD, of F over DE, respectively. Line AL intersects BC at P, line BM intersects CA at Q, line CN
intersects AB at R. Prove that P, Q, R are collinear.
IsTriangle(ABC)
IsIncircle(incircle, ABC)
IsTangent(BC, incircle)
IntersectsAt(BC, incircle, D)
IsTangent(CA, incircle)
IntersectsAt(CA, incircle, E)
IsTangent(AB, incircle)
IntersectsAt(AB, incircle, F)
ReflectionOf(D, EF, L)
ReflectionOf(E, FD, M)
ReflectionOf(F, DE, N)
IntersectsAt(AL, BC, P)
IntersectsAt(BM, CA, Q)
IntersectsAt(CN, AB, R)
Collinear(P, Q, R)

Given that points D, E lie on the side lines AB, BC of triangle ABC, respectively, point P is in interior of
triangle ABC such that PE = PC and triangle DEP is similar to triangle PCA. Prove that BP is tangent to the
circumcircle of triangle PAD.
IsTriangle(ABC)
Collinear(A, D, B)
Collinear(B, E, C)
Equals(PE, PC)
IsSimilar(DEP, PCA)
CircumcircleOf(circumcircle, PAD)
IsTangent(circumcircle, BP)

Let ABC be a triangle and its bisector of BAC cuts its circumcircle at D. Let I be the incenter of triangle ABC, M be the
midpoint of BC, P is the reflection of I over M (Assuming P is in the circumcircle). Extend DP until it cuts
the circumcircle again at N. Prove that among segments AN, BN, CN, there is a segment that is the sum of the other two.
IsTriangle(ABC)
ABisectorOf(bisector, BAC)
CircumcircleOf(circumcircle, BAC)
IntersectsAt(bisector, circumcircle, D)
IncenterOf(I, ABC)
MidpointOf(M, BC)
ReflectionOf(I, M, P)
OnCircle(P, circumcircle)
```

Currently, my `finalrelations` file only contains boolean relations – that is, all of my relations currently jsut take in some number of variables and output a boolean value corresponding to whether or not that statement is true. As I have been constructing these test cases, however, I've found that having relations that output variables as well would be useful, although it would make the testing much more difficult. For example, in the second displayed problem, the circumcircle of triangle PAD is relevant to the statement that is needed to be proved, although it's not given an explicit name other than "circumcircle." This doesn't present too much of an issue here, but with problems that involve more than one unnamed object, there is a pretty clear opportunity for confusion to occur while checking if a relation generated by the model is correct. Hence, I think it's reasonable to add one-argument relations such as `Circumcircle(var triangle)`, which would return a variable denoting the actual circle of that triangle, in order to avoid this.

Any relations that I've been adding to my `finalrelations` file over the past week or two also need to be somehow added into my `lexicon` file, or otherwise the code won't be able to detect these important relations that need to be detected for a successful "understanding" of the problem. This shouldn't take long, but it is something that needs to be addressed.