We attacked the analyzer part of the project in general by finding ways to store needed info as properties of entities. Name analysis was done by comparing new input against info stored in the tree. If the entities existed then we know that we are dealing with a use occurrence opposed to a defining occurrence. First we added rules in our AST that would print out the stored info to check for correctness. This insured that our name analysis was working properly on a general level. Then we worked on making sure that only the parameters needed are passed in as input for certain functions. This entailed using the computational roles that allowed us to check for uniqueness. This also allows us to print error messages if the proper input for certain functions are not met. Our biggest problem was figuring out how to store a nodes children as entities when there are an unspecified number of children. We looked at writing our own computational role in C or using OS specific ways to dynamically store this info. In the end we saw it was easier to just create PTG nodes out of all the children and store this final result as an entity in the node.