# Project Summary

*Short summary of the project setting.*

# Propositions

*List of the propositions used in the model, and their (English) interpretation.*

# Constraints

*List of constraint types used in the model and their (English) interpretation. You only need to provide one example for each constraint type: e.g., if you have constraints saying “cars have one colour assigned” in a car configuration setting, then you only need to show the constraints for a single car. Essentially, we want to see the pattern for all of the types of constraints, and not every constraint enumerated.*

# Model Exploration

*List all the ways that you have explored your model – not only the final version, but intermediate versions as well. See (C3) in the project description for ideas.*

# Jape Proof Ideas

*List the ideas you have to build sequents & proofs that relate to your project.*

# Requested Feedback

*Provide 2-3 questions you’d like the TA’s and other students to comment on.*

# First-Order Extension

*Describe how you might extend your model to a predicate logic setting, including how both the propositions and constraints would be updated.* ***There is no need to implement this extension!***

## Predicates

We will start with the propositions used for the base model, and extend to other predicates of interest.

* Edge(x,y): There is an edge from node x to node y
* Degree(x,n): Node x has degree n
* Connection(x,y,n): Node y is the nth connection starting at node x
* Connected: Graph is connected
* Distance(x,y,n): A path exists of length n from node x to y
* *Equality* : Implicitly lets us compare two objects for equality

To handle sorts, we would need the following predicates:

* Node(x): x is a node
* Number(n): n is a whole number

## Functions

* sum(n, n’): Function that returns an object that is the sum of n and n’
* sub(n, n’): Function that returns an object that is the subtraction of n and n’ (i.e., n – n’)

## Constraints

Some of the constraints we might have:

* If an edge exists between nodes, then their distance is 1
* If a node x can reach y with distance n, and y can reach z with distance n’, x can reach z with distance n+n’  
   )
* If distance\_x\_y\_n holds, then distance\_x\_z\_(n-1) must hold for some z connected to y
* If we can reach every node with some distance, then the graph is connected
* If a graph is connected, then between every pair of nodes there is a finite distance

## Potential Sequents/Theorems

*Things you might be able to prove, if you had infinite time/resources/jape window size*

…

# Useful Notation

*Feel free to copy/paste the symbols here and remove this section before submitting.*