DnD: Drinks n’ Dragons / CS1950y Final Project

Thomas Del Vecchio, Julie Wang, Katherine Sang

PREMISE

We initially set out to model the logic problem “Dragons”. To model this problem, we first modeled a similar, but simpler problem called “Logicians”. From modelling “Logicians”, we realized we could abstract our methodology to solve any similar logic problem, by intersecting each participant’s knowledge graphs with evidence graphs (more on that later). Additionally, we made our model more extensive by having it model any series of events, rather than hard coding the events given in the original premise.

Logicians: Three logicians walk into a bar. The bartender asks, “Do you all want drinks?” The first one says “I don’t know”. The second says “I don’t know”. The last says “yes”.

Dragons : There is an island of dragons, with either green or blue eyes. If a dragon knows it has green eyes, at midnight on the day of realization, they will leave the island. You tell the group of n dragons that at least one of them has green eyes (without stating who). On the nth day, they all transform into sparrows and leave the island.

MODEL

Both of the problems model each participant’s (Dragon/Logician) view of the world, known as their **knowledge graph**. Each world represents a possible situation, such as Logician 1, 2, and 3 wanting a drink. An edge between two worlds means that these two worlds are consistent (ex. From logician 1’s perspective, all worlds where Logician 1 wants a drink is in one equivalence class. The world where Logician 1 doesn’t want a drink is a separate equivalence class). Thus, we define equivalence classes as worlds that are consistent with each other.

Each participant also has an **evidence graph**, which is where each world is mapped to its outcome based on what event occurred. The evidence graph is public knowledge, thus all participants are able to access it. These equivalence classes tend to be coarser than the knowledge graphs.

Our general procedure is that we 1) establish everyone’s knowledge graphs 2) build the evidence graph 3) intersect each participant’s knowledge graph with the evidence graph to “update” their knowledge after an event. One thing to note is that in the Logicians problem, we update everyone’s knowledge graphs by intersecting with the speaker’s evidence graph, because the speaker speaking offers new information. In contrast, in the Dragons problem, we build a communal knowledge graph based on the participants' knowledge graphs. This is because no dragon “reveals” knowledge to the others like the Logician problem, because they discover their eye colors simultaneously.

To look at more specifics on how the evidence graph / knowledge graph / transitions are modeled per problem, we have left comments in the code.

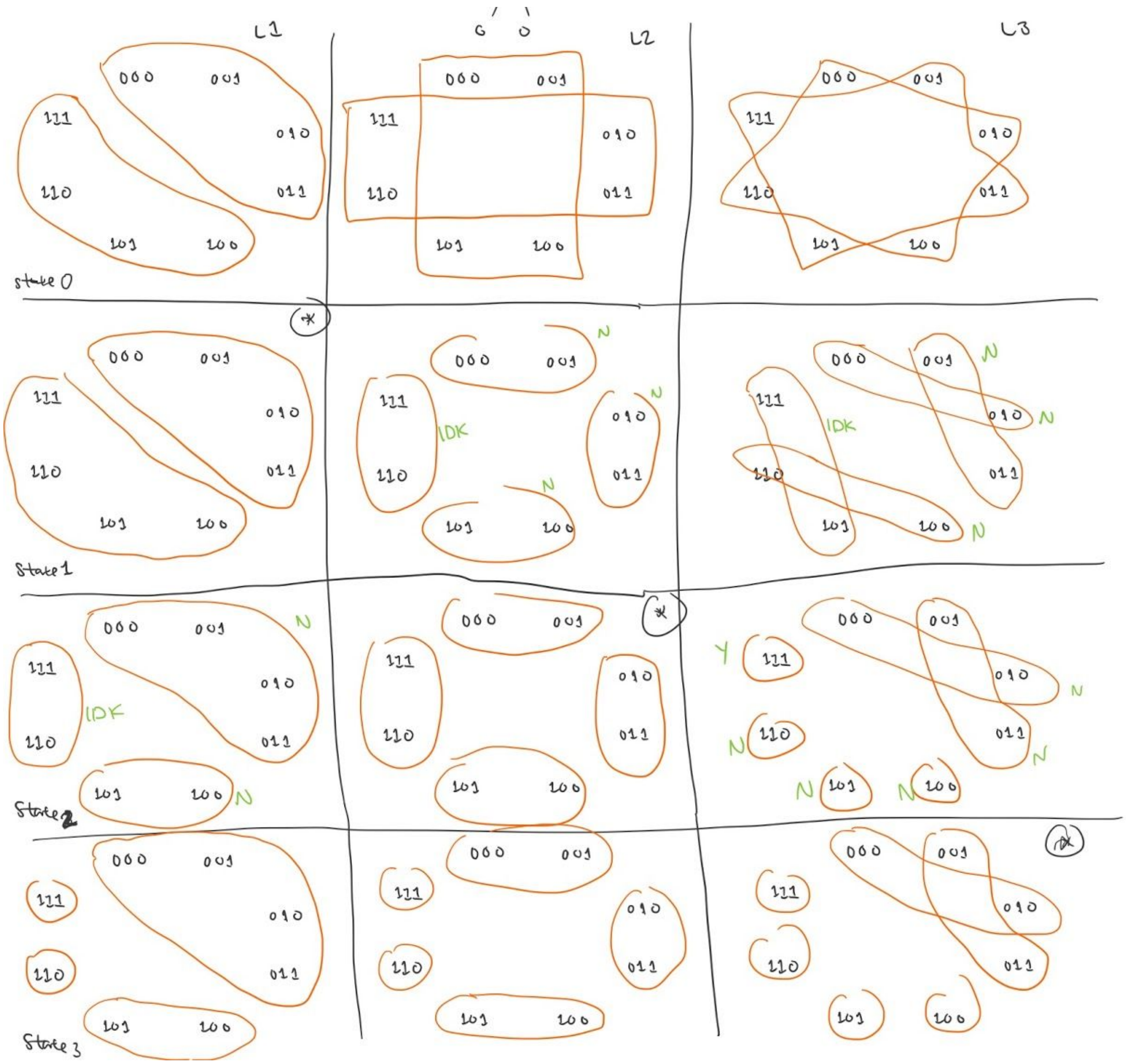

Fig 1: Hand simulation of an instance of the Logician’s problem, which we used to verify our outputs in Forge. Each column represents a logician, and each column represents a state, where each rectangle box contains the perspective of that specific logician in that state. The black * in the corner of a Logician’s state indicates which speaker has spoken. The 0/1 string represents a single world. Where Logiciian[i] wants a drink = 1, and Logician[i] doesn’t want a drink = 0, i being the index in the string.

The orange groupings mark an equivalence class in the knowledge graph (In forge, these are the worlds that would have edges between them). The green labels represent a labelling in the evidence graph. Note that transitions between each state is intersecting each logician’s previous knowledge and evidence graphs.

In the final state, not every world is in its own equivalence class. For example, when Logician 1 says no off the bat, everyone else will say no, so they will gain no more knowledge about the truth. Then, for Logician 1, worlds 000 001 010 and 011 don’t conflict with each other, thus are in the same equivalence class.

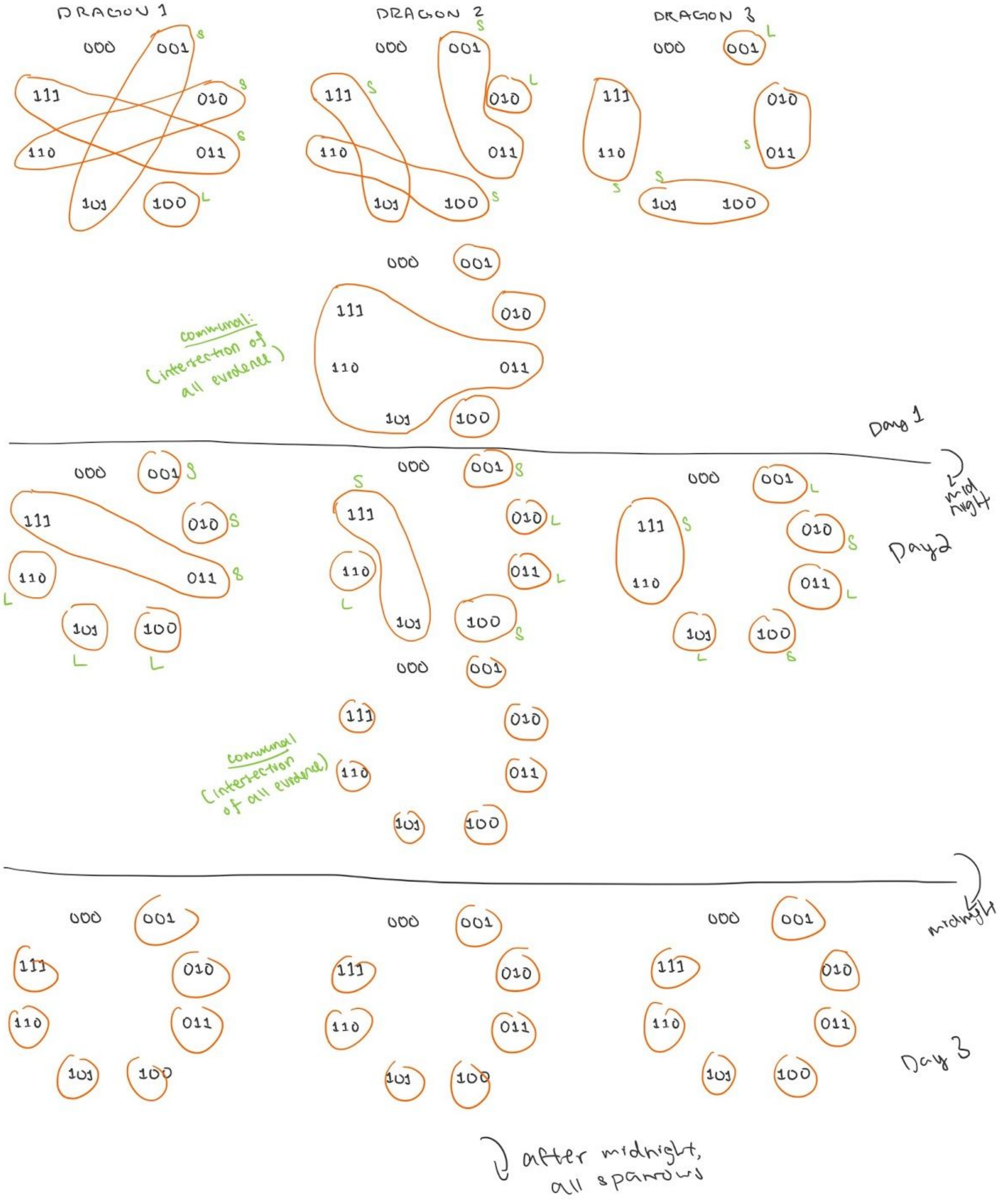
DRAGON 1
DRAGON 2
DRAGON 3
000
001
010
011
100
101
110
111
S
L
communal: (intersection of all evidence)
Day 1
midnight
Day2
communal (intersection of all evidence)
midnight
Day 3
after midnight, all sparrows

Fig 2: Hand simulation of an instance of the Dragon problem. The notation is the same as the logician diagram. Each world is a 0/1 string, and each orange circle is a collection of worlds that are consistent with each other. 1 means leave, and 0 means that the dragon will stay.

Notice that in each state, we take the intersection of all the dragons' evidence (green) to make the communal graph. The next state's evidence graph is the result of the intersection of the communal and previous knowledge graphs.

RECOMMENDED VISUALIZER SETTINGS IN FORGE:

1) State -> Dragon/Logician
2) Set eyeColors/preferences to display as attribute
3) Toggle through States to see an individual dragon/logician as the situation progresses and they learn more

VERIFICATION & TESTING

- Hand simulated (Fig 1 & 2) the outcomes for 3 logicians and 3 dragons in order to understand our approach and also verify the solutions in the visualizer
- Viewing up to 6 logicians / dragons concrete instances
  - Python script (`inst_script_dragon.py` and `inst_script_logician.py`) to hard code the possible worlds and number of logicians. When we originally tried running the problem without a concrete instance, it took quite a long time (4 dragons ~ 15 min). Thus, we created some concrete instances to see if forge would run faster. The run time for our concrete instances is significantly faster -- about 4-8 seconds for each. We have a **user interaction zone** on each problem's model, where the user can conveniently view the different concrete instances in Forge's visualizer.
- Testing facts that should be true across states (up to 4 participants)
  - `increasingKnowledge` - participants should know more (as reflected in the knowledge graph) as states progress.
  - `leavePermanent/answerPermanent` - once a participant has made an answer (for the dragon, leaving the island, for the logician, when they speak), they must be consistently in this world for later states.
  - `correctSolution` - the original answer to the premise. For Dragons, all dragons will leave on the final day because they discover their eyes are green. For Logicians, the series of speakers will go [I don't know, I don't know … I don't know, Yes] if they do all want drinks in the first place.

CONCLUDING THOUGHTS

We originally sought to model the situation of the original dragon premise. Then, we used the logicians puzzle as a stepping stone to solving the dragon problem. Additionally, our approach for modeling the problem (updating knowledge graphs by intersecting with an evidence graph) can be generally applied to other problems of the same nature. Given that we have a method that models all the possible conclusions made on all possible events in the given premise, we have reached our original goal.