Investigating Simple Gossip Networks and Epistemic Logic: A Study of Information Propagation

Luca Bandelli (s3221253)
Yara Bikowski (S3989585)

General introduction

In today's interconnected world, the exchange of information can both help and hinder us in our daily life as the information could be false. Gossip, which is often viewed as a negative concept when used for nefarious means, can serve as a powerful mechanism to disseminate knowledge among individuals within a network. By understanding gossip in a network, we can think about how to use it for better purposes like the exchange of important information.

In this report, our main goal is to investigate how long it would take for every agent within a network to become aware that everyone else in the network knows a piece of new information. We achieve this by constructing ring networks with different numbers of agents. The ring network represents the potential communication channels between agents which enables the propagation of information throughout the network. The gossip exchange among these agents can help them become aware that other agents (that they have no connection to) also know the new information.

The report will first provide a comprehensive background on the necessary epistemic logic that is used to reason about (higher-order) knowledge of agents within an often multi-agent system. We will specifically focus on the 'E' operator and the [connected] rules that allow us to analyze the evolution of knowledge as gossip circulates in the network. We will also mention the simplifications in our model that we make to help us keep this project to a manageable scope.

In this report, we will present an example that walks through the logic used for a network with four agents. This will further help explain the knowledge updates based on epistemic logic and gossip in the network.

Simplifications

Now let's look at the gossip networks. Each network consists of a set of m different agents where only agent 1 knows information "p". The networks have a ring shape where each agent is connected to two other agents. At any given time, an agent can only exchange their knowledge and information with one of these agents that they are connected to. However, there can be multiple different agents that share their knowledge with other agents at the same time. The possible announcements that an agent can make are 'p', 'Ep' and 'K_ip' with ($i \leq m$). This means that they can announce that information p is true and which agents *know* that information p is true. When agents make these announcements to each other, the other agents in the network won't know that any information is exchanged between them. So, these announcements are done in secret without anyone's knowledge.

Both agents will exchange their information but the agent that initiates the gossip will do the first announcement. This doesn't make a big difference when both agents have some information. However, it is mainly done for the case where only one of the agents has knowledge since the other agent can expand on the knowledge after obtaining it. As can be seen in the example below. The agents will always exchange all information that they have.

We measure how long it takes for information to spread based on different time steps. Here one single timestep passes after at least one agent (more is also allowed) exchanges information with another agent. This can most easily be counted by looking at agent 1 since this agent will always exchange information with one of the agents that they are connected with.

Epistemic logic

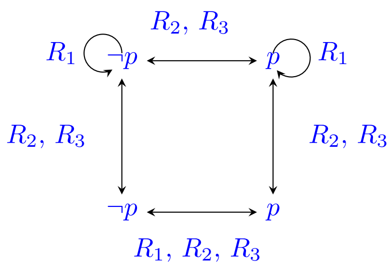
We will use the $KEC_{(m)}$ logic system combined with gossip through private announcements. The most important syntax from the logic system that will be used during the gossip in our model are:

- (R2): From φ follows $K_i\varphi$
- (A6): $E\varphi \leftrightarrow (K_1\varphi \wedge ... \wedge K_m\varphi)$

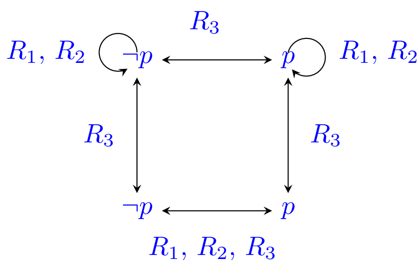
From axiom A6, we can get $(K_1\varphi \wedge ... \wedge K_m\varphi) \rightarrow E\varphi$ using equivalence-elimination as can be seen from the following proof:

1. $E\varphi \leftrightarrow (K_1\varphi \wedge ... \wedge K_m\varphi)$ (A6)
2. $(K_1\varphi \wedge ... \wedge K_m\varphi) \rightarrow E\varphi$ (EE:1)

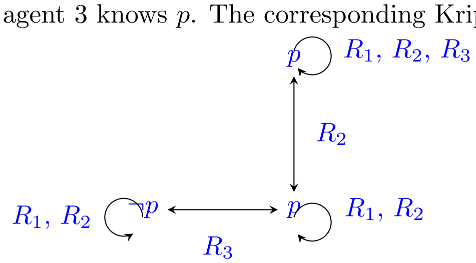
We will show how knowledge can change in a network with 3 agents based on the evolution of a simplified Kripke model. We have simplified the model to only look at second-order knowledge to make sure that the model doesn't become too big. At first, the knowledge in the network consists of K_1p, $\neg K_2K_1p$, $\neg K_3K_1p$. The corresponding Kripke model is:



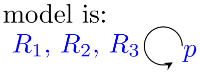
After communication between agent 1 and agent 2, the knowledge in the network consists of K_1p, K_2p, $\neg K_3K_1p$, $\neg K_3K_2p$. So, agent 3 is the only one that isn't aware of the knowledge of the other agents. The corresponding Kripke model is:



After communication between agent 1 and agent 3, the knowledge in the network consists of K_1p, K_2p, K_3p, $\neg K_2K_3p$. So, agent 2 is not yet aware that agent 3 knows p. The corresponding Kripke model is:



After communication between agent 1 and agent 2, the knowledge in the network consists of K_1p, K_2p, K_3p. So, every agent knows p (Ep) and every agent also knows that every other agent knows p (EEP). The corresponding Kripke model is:



Example with 4 agents

As we explained before, the final goal of the gossip is for all agents to have K_iEp in their knowledge base. This basically means that we want all agents to know that everyone knows information p.

At time 0, only agent 1 is aware of information 'p'. Using R2 from $KEC_{(m)}$, from p follows K_1p. So, the network at time 0 is as follows:

$$\begin{tikzcd} p,\ K_1p\ \ agent1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & agent2 \arrow[d, leftrightarrow] \\ agent4 \arrow[r, leftrightarrow] & agent3 \end{tikzcd}$$

At time 1, only agent 1 has any information they can exchange. They will first do this with the agent to their right which is agent 2 in this network. As we mentioned before, the agent will exchange all their information. This means that agent 1 makes the announcement "[p, K_1p]" to agent 2. Now agent 2 is also aware of the information p and will thus also have the knowledge K_2p based on R2 as can be seen in this example at time 0. At this point, agent 2 has the knowledge p, K_1p, K_2p. This is what they will announce back to agent 1 "[p, K_1p, K_2p]". Thus, agent 1 also obtains the knowledge p, K_1p, K_2p. The network at time 1 is as follows:

$$\begin{tikzcd} p,\ K_1p,\ K_2p\ \ agent1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & agent2\ \ p,\ K_1p,\ K_2p \arrow[d, leftrightarrow] \\ agent4 \arrow[r, leftrightarrow] & agent3 \end{tikzcd}$$

At time 2, agent 1 will communicate with the agent at the opposite side. This is agent 4. At the same time, agent 2 will try to communicate with the second agent that they are connected with as well. This is agent 3 and since this agent is not yet communicating with a different agent, the gossip wil also be done between these two agents. Agent 1 will announce all the information they have to agent 4, which is: [p, K_1p, K_2p]. Agent 4 then knows p and using R2 also knows K_4p. Thus, agent 4's knowledge is p, K_1p, K_2p, K_4p and that is then announced to agent 1. Agent 2 also announces their knowledge to agent 3 at the same time. They have the same knowledge as agent 1 and thus their announcement will also be [p, K_1p, K_2p]. Similar to agent 4, agent 3 will then also update their knowledge based on R2. This results in agent 3 having the knowledge p, K_1p, K_2p, and K_3p. Agent 3 will then announce this back to agent 2 which means that agent 2 gets the same knowledge as agent 3. The network at time 2 is as follows:

$$\begin{tikzcd} p,\ K_1p,\ K_2p,\ K_4p\ \ agent1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & agent2\ \ p,\ K_1p,\ K_2p,\ K_3p \arrow[d, leftrightarrow] \\ p,\ K_1p,\ K_2p,\ K_4p\ \ agent4 \arrow[r, leftrightarrow] & agent3\ \ p,\ K_1p,\ K_2p,\ K_3p \end{tikzcd}$$

At time 3, agent 1 will go communicate with agent 2 again and agent 3 will communicate with agent 4. Agent 1 will announce [p, K_1p, K_2p, K_4p] after which agent 2 has the knowledge p, K_1p, K_2p, K_3p, K_4p. As we showed before, from axiom A6 of $KEC_{(m)}$ we can get $(K_1p \wedge K_2p \wedge K_3p \wedge K_4p) \rightarrow Ep$. So, after using this rule, agent 2 has the knowledge Ep and after using R2, agent 2 has K_2Ep in their knowledge base which is the final goal for our gossip. Agent 2 will then announce the following parts of their knowledge (based on the simplifications we mentioned): [p, K_1p, K_2p, K_3p, K_4p, Ep]. This gives agent 1 the knowledge p, K_1p, K_2p, K_3p, K_4p, Ep. This agent already gets Ep from the announcement and thus doesn't have to use epistemic logic to get it themselves. However, they do use R2 to get K_1Ep. Agent 3 announces [p, K_1p, K_2p, K_3p] to agent 4 which allows agent 4 to get the knowledge K_3p and then just like agent 2, agent 4 will get the knowledge Ep and K_4Ep from axiom A6 and rule R2. Agent 4 announces [p, K_1p, K_2p, K_3p, K_4p, Ep] to agent 3 and then agent 3 will know Ep and from rule R2 K_3Ep. The network at time 3 is as follows:

$$\begin{tikzcd}
p,\ K_1p,\ K_2p,\ K_3p,\ K_4p,\ Ep,\ K_1Ep \quad agent1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & agent2 \quad p,\ K_1p,\ K_2p,\ K_3p,\ K_4p,\ Ep,\ K_2Ep \arrow[d, leftrightarrow] \\
p,\ K_1p,\ K_2p,\ K_3p,\ K_4p,\ Ep,\ K_4Ep \quad agent4 \arrow[r, leftrightarrow] & agent3 \quad p,\ K_1p,\ K_2p,\ K_3p,\ K_4p,\ Ep,\ K_3Ep
\end{tikzcd}$$

At this point, each agent has K_iEp with ($i = agent$) in their knowledge base. So, we stop the gossip here since the goal is met and no more information can be exchanged. It took 3 time steps for every agent to know that everyone knows the new information p when the network consists of 4 agents.

Implementation

We implemented a computer simulation of the ring-network protocol described above. The program makes use of the `networkx` package to generate and manage the graph structure. The graph structure contains one node per agent, and each agent contains a set of formulas, which we refer to as their *kowledge base*.

At each discrete time-step of the simulation, the next set of active edges is computed and used for the diffusion of information. The diffusion step which represents the act of gossip, consists of a symmetric update of the knowledge bases of the two agents involved in each edge.

After the update is performed the Necessitation and E-introduction rules are on the knowledge bases of agents that satisfy their precondition. The simulation stops when 2 subsequent steps result in no change.

Findings

In order to study the dynamics of the simulation we performed a simple experiment, where the independent variable is the number of agents making up the gossip ring network `n_agents`. For each setting of `n_agents` we run the simulation and collect as response variable the number of steps it takes to satisfy a number of global properties of the system. In particular, we observe the following properties:

all_p The number of steps required for all agents to have the atom p in their knowledge base.

all_K_1_p The number of steps required for all agents to have K_1p in their knowledge base.

all_E_p Steps for everyone to infer that Ep.

Moreover for each run we compute the following ratios:

- *all_E_p* to *all_p*
- *all_E_p* to *n_agents*
- *all_p* to *n_agents*

Figures 1 show the results of the experiment with the number of agents varying from 3 to 40. The results indicate that there is a linear relation between the number of agents and the time it takes to reach the point where everybody has Ep in the knowledge base. The staircase pattern the data exhibits is due to the difference in diffusion dynamics when the number of agents is even versus odd. When the number of agents is odd, because our gossip protocols selects pairs of agents for communication, and each agent can only be selected once, this results in one agent being left out. Which agent is left out changes at every iteration, starting from the last agent, and then shifting left towards the first one.

The red line in 1 shows that the ratio $\frac{all\_E\_p}{all\_p}$ is constant at 2, i.e. it always takes twice as many steps to reach property *all_E_p* than *all_p*. Figures 2 and its counterpart in a setting with very high number of agents 3, show that on the other hand the ratios $\frac{all\_E\_p}{n\_agents}$ and $\frac{all\_p}{n\_agents}$ have logarithmic dependence on the number of agents. The latter asymptotically converges to 0.5, indicating that, as expected, since information diffuses in both directions along the ring, it takes a number of steps approaching half of the agents to completely diffuse the p atom (same applies for K_0p). Conversely completely diffusing that for all i K_ip, and consequently firing the E-introduction rule, takes twice as many steps, converging to a ratio of 1. The results in figure 2 which contain also odd sized networks, highlights once more the inefficiency of the pair-wise protocol in these scenarios.

In future studies, it would be interesting to look at different topologies for the gossip network and see what influence this has on the diffusion speed of the network. For example dense randomly connected graphs or realistic, scale-free networks generated via preferential attachment.

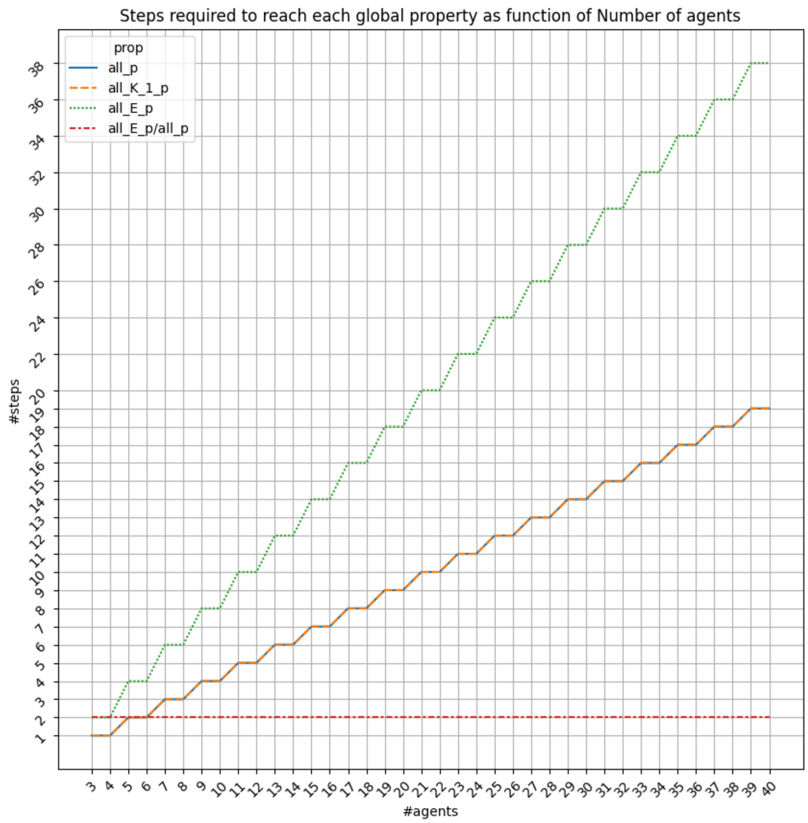


Figure 1: Visualization of the experiment results. The `E_p/p_ratio` line is added to show the constant relation between *all_E_p* and *all_p*

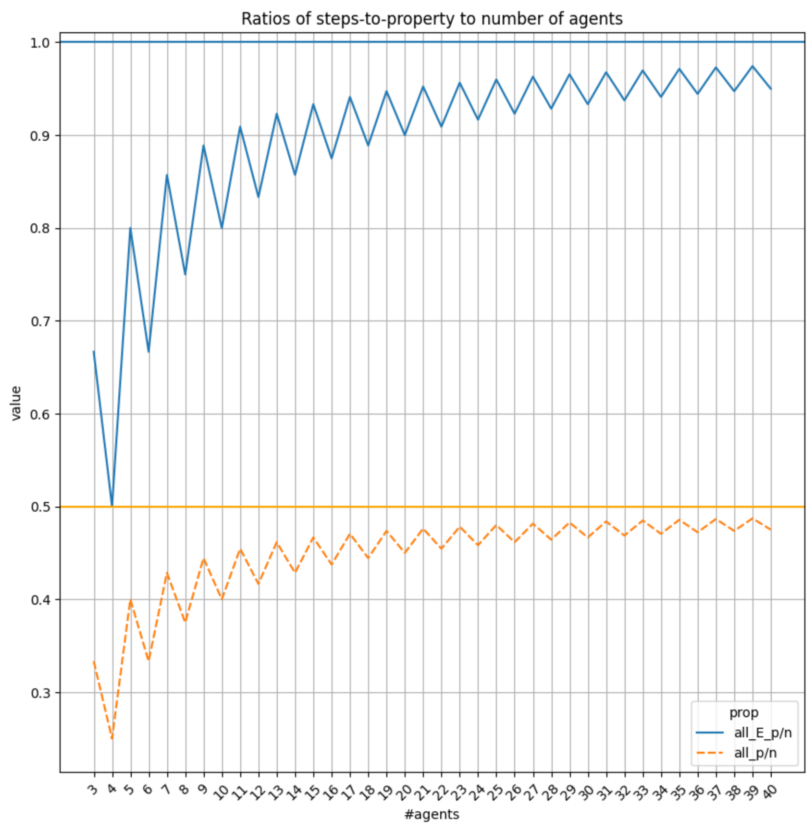


Figure 2: Visualization of network propagation efficiency expressed as the ratio of *all_E_p* to the number of agents.

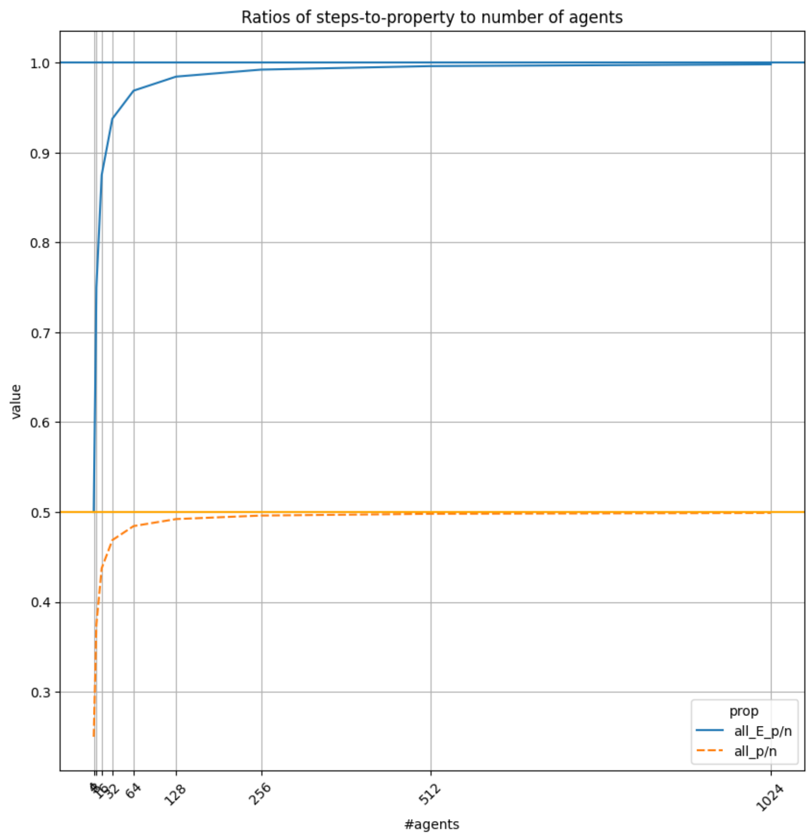


Figure 3: Visualization of network propagation efficiency expressed as the ratio of *all_E_p* to the number of agents. (Large number of agents experiment, all even sized networks)