Logical Aspects of Multi-Agent Systems Go Fish! Analysis and application using epistemic logic

Charlie Albietz (S3058735)
Corine Nijhof (S2967383)
Rafael Tappe Maestro (S3258734)

June 25, 2021

Abstract

In this work the game *Go Fish!* is analyzed using the framework of modal epistemic logic and apply our findings to an online playable version of the game. The use of Kripke models helps to find efficient strategies for winning the game. A web application was developed that allows a human user to play the game against two AI agents. The AI agents can use different strategies that are based on the logical analysis. A small experiment in which four different strategies are tested against each other showed that using first- or second-order knowledge improves a player's chances of winning significantly.

1 Introduction

Go Fish! could not only be a call to the fishing rods but is also the name of a card game. It is known under different names and variations in Europe and the Anglosphere; it resembles the game Quartets. In this work we analyse the game *Go Fish!* using the framework of modal epistemic logic and apply our findings to an online playable version of the game.

This report is structured as follows. First, the rules of *Go Fish!* are described in Section 1.1. Second, in section 2 we analyse game scenarios by example to develop an intuition about inferences on the game state that can be made with epistemic logic. Third, we formalize our notion of epistemic inferences on the game logic with Kripke models in section 3. After that section 4 presents our methods on the development of a web application that allows a human user to play a game of *Go Fish!* against AI opponents. In particular the strategy played by the AI opponents is an application derived from Kripke models. Section 5 describes methods and results of a small experiment to test the game strategies in an adapted version of the application in which three AI agents play against each other, without a human user player. Lastly, conclusions and discussion of the project can be found in sections 6 and 7.

1.1 Game Rules

Go Fish! is a card game played with a French-suited standard 52-card deck. In a real world setting the game is played by two to five players. In this work we focus on three player games. The goal for each player is to maximize the number of books collected by the end of the game. A book is a set of four cards with identical rank (card number or face). Suits don't play a role in *Go Fish!*.

At the beginning of the game each player is dealt seven random cards from the deck (five cards for four or more players). Dealt cards are only visible to the receiving player. The deck lays face down.

After dealing, a random player is designated to take a turn. The turn taking player asks another player for all cards of a rank that he owns at least one card of. If the asked player has any number of cards of the requested rank the cards are handed over to the asking player. In the case that cards were handed over, the turn taking player again asks another player for all cards of some rank. This condition may apply an unlimited number of times; the asked player and the asked card can be freely chosen each time by the turn taker. Whenever cards are handed over they are revealed to all players. If the asked player has no cards of the requested rank then the asked player says *"Go Fish!"* and the asking player draws a card from the top of the deck; this ends the turn.

After completing a turn, the player to the left of the previous player takes a turn (clockwise play). Once a player completes a book, the respective cards are placed on the playing table face up. Whenever a player has no cards on hand the player immediately draws one card from the top of the deck. Afterwards the game proceeds as described.

The game ends when all books have been completed. The winning player is the player with most books.

2 Example scenarios

To give the reader an intuitive idea of how epistemic knowledge can be used in the game, we walk through some possible scenarios.

1. Suppose player A has one king and the other players do not know that. In A's turn, he asks player B for kings, and is told to Go Fish. After this action, every player in the game knows that B has no kings. Additionally, since A asked for kings, he must have at least one king, which is now also common knowledge. Whether B has kings or not is only known by player B himself.

2. Suppose players A and B both have one queen, player C has two queens, and nobody knows about the queens of the other players. In C's turn, he asks player A for queens and receives A's queen. After this action, similarly as in the previous scenario, it becomes common knowledge that A has no queens, and C has at least two queens (he had to already have one, and he received another one), but at most three (a complete book has to be laid down on the table). Only C knows that he has exactly three queens. Then it is still C's turn. At this point C knows that B knows that C has queens. However, C does not know whether B has the fourth queen. From C's perspective, if C does not ask B for queens right now,

and B does have a queen, then B will probably ask C for queens and C will lose all his queens. Therefore, the best move for C is now to also ask B for queens, to eliminate the chance of losing the book of queens to B.

3. Suppose player A has three jacks after receiving one jack from B and one jack from C. Then it is common knowledge that A has three jacks, as he has not laid down a complete book yet. After this turn of A, both players B and C cannot ask for A's jacks, as they do not have jacks anymore and have not yet drawn new cards from the deck (assuming no one draws a card after emptying their hand). In A's next turn however, both players B and C have drawn at least one new card from the deck, which A knows, and therefore A considers it possible that B or C has a jack. So we end up in a similar situation as in the previous scenario. If A does not ask for the jack now, and B or C has indeed drawn the jack, A will lose all his jacks. The best move for A is therefore to ask B or C for jacks. Which one of the two is a random guess.

After working through these scenarios, some general remarks can be made about the course of the game. Firstly, we have seen that as soon as a player asks for a card that he has not asked for before, everyone gains important information about the player's hand. This can be a reason for players to not start asking for a 'new' card unless there is no other option. Secondly, once a player has three cards of one book, there exists a new chance that the other players draw the fourth card from the deck each round. It might therefore happen, if the player does not like taking risks, that he keeps asking for the same card each round, until either he receives the card from another player or the deck, or another player that he did not choose to ask, drew the card and asks back all the other three cards. If the fourth card happens to be very low in the deck, the hand of the player keeps growing with every turn, while the other players gain very little knowledge about all the other cards in his hand.

3 Kripke model

Translating the game of *Go Fish!* to Kripke models is not a straightforward task. The game is hard to track, because instead of the state changing from one state in the set of possible states to another state in that set, the whole set of possible states changes with each action. After every action the state of the turn-taking player and one of the other players changes by adding or removing cards.

3.1 Formal model

We use the following formal notations for the Kripke models. Let B be the set of players, with $|B| = n$ and let h be the number of initial hand cards. Let $h(i,s)$ be the number of hand cards of player i in state s. Let P be the set of propositions. Specifically, we denote the sentence "player i has card c"

as proposition i_c, where $c \in C = \{2, \ldots, 10, J, Q, K, A\}$. Multiple cards can be indicated as a sequence in the subscript: "player i has a 3 and J" is the proposition i_{3J}. We define a Kripke model $M = \langle S, \pi, R \rangle$, where

1. S is the set of states $s = (s_1, \ldots, s_p)$, where s_i is the set of hand cards $\langle c_1, \ldots, c_{h(i,s)} \rangle$ if player i;
2. $\pi : P \to (S \to \{\mathbb{t}, \mathbb{f}\})$ assigns a truth value to all propositions in P; and
3. $R = \{R_1, \ldots, R_p\}$ is the set of relations $R_i \subseteq S \times S$ such that $R_i = \{(s,t) \in S \times S | s_i = t_i, h(j,s) = h(j,t) \text{ for all } j \in B\}$, i.e. each player knows its own cards and also how many cards the other players have.

The set of relations in this model resembles the set of relations of distributed systems, as an agent can only access states in which its own local state is equal. The difference is that in this model, the agents can also only access states in which the number of cards in the hands of the other players are equal. Note that the accessibility relations in the model are reflexive, transitive and Euclidean.

Before proceeding to working out the Kripke models in an example game, we note that unlike what might be expected, the Kripke model of a *Go Fish!* game does not necessarily decrease in size throughout the game. In fact, it might happen that the Kripke model increases in size after a move. For example, when a player asks for a 5 for two turns in a row, then no new knowledge becomes available in the second turn about this player's cards, but since a new card has to be drawn from the closed deck after each turn, a new factor of uncertainty is added to the model, which increases the number of possible states and therefore the size of the model.

3.2 Simplified game

To be able to visualize a Kripke model of a *Go Fish!* game, we use a simplified version of the game. We use only two books of cards (queens and aces), and let the players initially only have a single hand card. Furthermore, since the number of possible combinations of cards can be rather large, the states of which everybody knows they cannot be true (e.g. a state where a player does not have a Q, even though they just received a Q from another player) are not shown in the visualized models.

The game is started by dealing the hands. Player 1 and 2 both get a queen, player 3 gets an ace. Formally, we initially have 1_Q, 2_Q and 3_A in the real world. The players only know their own card, so we have $K_1 1_Q$ and $\neg K_1(2_Q \wedge 3_A)$, $K_2 2_Q$ and $\neg K_2(1_Q \wedge 3_A)$, and $K_3 3_A$ and $\neg K_3(1_Q \wedge 2_Q)$. The Kripke model of this initial situation can be seen in Figure 1a. Only states with one card for each player are shown. The set of states includes other states, in which players have more than one card, but these states are not connected to this graph through accessibility relations (as the number of hand cards is not the same for each player) and therefore not shown. Table 1 shows an example of how this game could unfold.

Turn	Player	Asks player	Card	Result	1	2	3
initial					Q	Q	A
1	1	2	Q	1 gets one Q 2 draws a new card	$\underline{\text{QQ}}$	A	A
2	1	3	Q	Go Fish!	$\underline{\text{QQ}}$A	A	$\underline{\text{A*}}$
3	2	3	A	2 gets one A 3 draws a new card	$\underline{\text{QQ}}$A	$\underline{\text{AA}}$	Q
4	2	1	A	2 gets one A	$\underline{\text{QQ}}$	$\underline{\text{AAA}}$	Q
5	2	3	A	Go Fish!	$\underline{\text{QQ}}$	$\underline{\text{Q*AAA}}$	$\underline{\text{Q*}}$
6	3	1	Q	3 gets two Q's 1 draws a new card	$\underline{\text{A*}}$	$\underline{\text{QAAA}}$	$\underline{\text{QQQ}}$
7	3	2	Q	3 gets one Q 3's turn ends	$\underline{\text{A}}$	$\underline{\text{AAA}}$	$\underline{\text{QQQQ}}$
8	1	2	A	1 gets three A's game ends	$\underline{\text{AAAA}}$		$\underline{\text{QQQQ}}$

Table 1: Draft representation. A simplified game with three players, two books of cards and one initial hand card each. The first column numbers the turns, the second column shows the player on turn, the third column shows the player that is asked a question, the fourth column shows the card that is asked for, and the fifth column shows the result of the turn, whether the player gets a card or has to Go Fish. When a player ends up without any cards because of this, he draws a new card. The last three columns show the cards that the three players have after each turn. Cards are underlined when, assuming all players are perfect logicians, it is common knowledge that the player has them. The four stars mark special cases of indirect knowledge, explained in the text.

After player 1 asks player 2 for a queen in turn 1, it becomes common knowledge that player 1 has a queen: $C1_Q$.[1] When player 2 answers and gives away one queen, it becomes common knowledge that player 1 has two queens, and player 2 has no queens: $C(1_{QQ} \wedge \neg 2_Q)$. Figure 1b shows that at this point player 1 has the most uncertainty, as he only knows his own cards, while the other players know both player 1's cards ánd their own card.

In the next turn, player 1 asking player 3 the same question does not reveal any additional information in the game. Player 3's answer however reveals that player 3 does not have any Q's which becomes common knowledge: $C\neg 3_Q$. Visually, Figure 1c shows that after player 1 draws a card, player 3 becomes the player with the most uncertainty, as he considers four states possible, while the other players only consider 2 states possible. In Table 1, the star in turn 2 marks the special case that this ace card of player 3 is not directly but indirectly known by player 1 and 2, because player 3 claims not to have a queen by telling player 1

[1] One could look at player 1's question as if it was a public announcement, i.e. $\langle$player 1 asks for a queen$\rangle C1_Q$.

to Go Fish. If p_i would be the sentence "player i has one card", then this indirect common knowledge can be (more generally) formulated as $(C\neg i_Q \wedge p_i) \rightarrow Ci_A$, or conversely $(C\neg i_A \wedge p_i) \rightarrow Ci_Q$ for each player $i \in B$. Note that these are only true in games with only the two books of queens and aces.

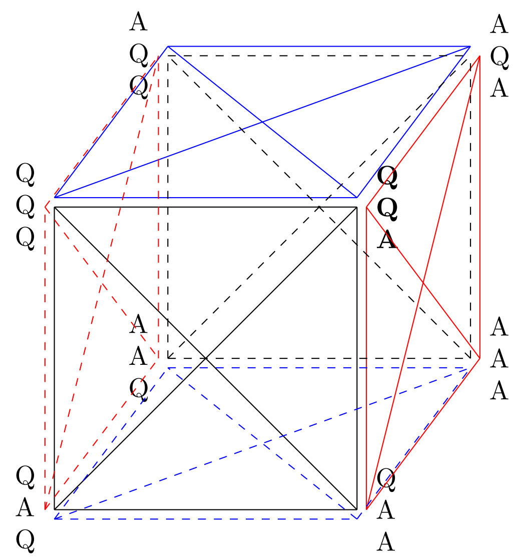
As player 1's turn ends, it is player 2's turn. He has an ace, and since he knows player 3 has an ace ($K_2 3_A$), he asks for that ace (after which $C2_A$), receives it (after which $C2_A A$) and gets another turn. Player 3 draws a new card, as he does not have hand cards anymore. Figure 1d shows that player 2 is the player with the most uncertainty, as he is uncertain about two hand cards in the game, while the other players are only uncertain about one hand card in the game. At this point player 2 still only has aces and therefore can only ask for aces. Both opponents have one card of which player 2 does not know whether they are aces or queens. He chooses to ask player 1 for aces, and receives another one. Now $C(\neg 1_A \wedge 2_A AA)$, as can be seen in Figure 1e, so player 2 asks player 3 again for aces, after which he has to *Go Fish!* and ends his turn. The second star in turn 5 indicates the same kind of indirect knowledge as in turn 2: $(C\neg 3_A \wedge p_3) \rightarrow C3_Q$. The first star in player 2's hand indicates that it is indirectly known this card is a queen, because if it would have been an ace, player 2 would have a complete book and shown this to everyone.

Player 3's turn starts. At this point, because of the small number of cards in the game, all hand cards are known by all players: $C(1_Q Q \wedge 2_Q AAA \wedge 3_Q QQ)$. Player 3 chooses to first ask player 1 for queens, after which player 1 has no hand cards anymore and draws the last card from the deck. The star in turn 6 marks the case that this card is indirectly known by the other players because they all know that there are already four queens in the game and therefore this card is and ace. After also asking player 2 for the last queen, player 3's turn ends, as there is no card left in the deck. Player 1 then asks player 2 for the aces and the game ends: player 2 lost and players 1 and 3 tied in first place.

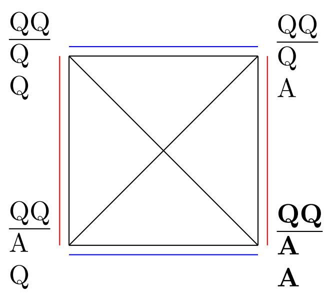
3.3 Perfect logicians?

A big assumption that is made when saying certain facts become common knowledge is that all players in the game are perfect logicians. Moreover, it is assumed that players have perfect memory. In real life, this might not be the case. Players could not realize that opponents know about cards that they have, they could forget common knowledge facts from earlier in the game, or they could simply be playing for fun instead of playing to win.

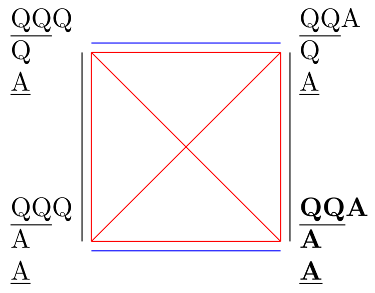
Formally one could distinguish between the order of knowledge that players have. Players that have zeroth-order knowledge do not keep track of anything in the game, they do not know anything about hand cards of any player and play randomly. Players with first-order knowledge keep track of the hand cards of their opponents, e.g. when an opponent asks for jacks, this player knows that the opponent has at least one jack. Players with second-order knowledge also keep track of what their opponents know about his own hand cards, e.g. when he asks for kings, this player knows that all other players know he has at least one king. These different levels of knowledge can be translated to different



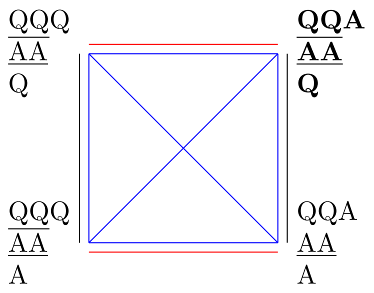
(a) Kripke model after initialization of the game.



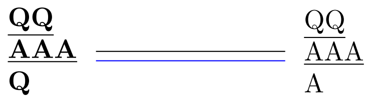
(b) Kripke model after the first turn in the game.



(c) Kripke model after the second turn in the game.



(d) Kripke model after the third turn in the game.



(e) Kripke model after the fourth turn in the game.

Figure 1: Visualizations of the Kripke models at initialization and after the first four steps in the game. The states show the hand cards of player 1 on top, player 2 in the middle and player 3 on the bottom. The true state is in bold. Black lines represent the symmetric accessibility relations of player **1**, blue lines those of player **2** and red lines those of player **3**. Underlined cards in the states are cards that are commonly known among the players (assuming all players are perfect logicians). Reflexive arrows and states that are not considered possible by any player are not shown (except for the state A/A/Q in the initial model, which is included for completeness).

strategies, which will be discussed more in section 4.1.

4 Application

To be able to apply the theoretical analysis of the *Go Fish!* game, we have developed an interactive web application. The application models the *Go Fish!* game. The final version on the website enables a human user to play the game against two AI opponents. Only for our experiment, we made three AI players play against each other, the results of which can be found in section 5. The application is built on top of HTML, CSS and JavaScript without the use of frameworks. The JavaScript source code aims to follow the ECMAScript 2020 standard. The source code only relies on the JavaScript standard library and has no external dependencies. The developed application was tested against Chrome 91, Firefox 89 and Node version 12.18.2. The application is deployed on GitHub pages and can be found at `https://wehzie.github.io/2021-lamas/`.

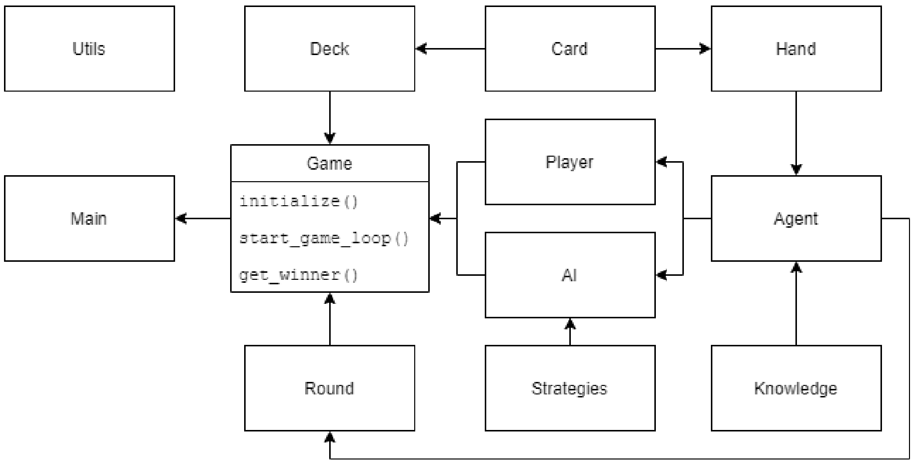


Figure 2: Dependency graph of the implementation.

Figure 2 shows a dependency graph for the game implementation in JavaScript. Each box corresponds to a class in the object-oriented programming paradigm; with the exception of the Main and Utils boxes. For the Game class, its methods are listed below. These three methods control the main course of a game from initialization, to turn taking to ending the game. The arrows in the dependency graph flow from dependencies towards depending classes. For example class Agent requires the notion of a hand of cards; this is implemented in class Hand. A hand in turn depends on the notion of cards found defined in class Card. No dependency arrows are shown for the Utils class because the Utils class itself has no dependencies and it contains only pure functions; Utils may be considered accessible from any other class.

4.1 Strategies

Because of the big number of possible states in the Kripke model of a *Go Fish!* game, we chose to not implement the game using Kripke models, but directly implement strategies that are based on the Kripke model. Using the epistemic logic analysis of the game, we implemented four strategies for the AI agents, representing four different difficulty levels. The user will be able to choose a strategy for each AI he is playing against.

4.1.1 Random

The name of the first strategy says it all: an agent with this strategy will play randomly. In its turn the agent will choose a random opponent to ask and a random (hand) card to ask for. No knowledge or strategy is involved.

4.1.2 Most cards

Involving a little more strategy than randomly choosing, but without considering epistemic logic, brings us to the most cards strategy. With this strategy an agent asks a random opponent for the card that it has the most of. Ties are broken randomly.

4.1.3 First-order knowledge

An agent with the third strategy involving first-order knowledge can keep track of what he knows about the hand cards of its opponents. As explained above, this means for example after an opponent asks for a king, the agent knows this opponent has to have at least one king. This knowledge enables the agent to check whether he knows about cards of opponents that he can ask for in its turn, which gives the agent more certainty of succeeding and therefore increases its chance to win. In the implementation, an agent with first-order knowledge keeps track of its knowledge about its opponents using two arrays, one for each opponent. When there is more than one option for asking for known cards, the agent considers the net gain of these options, e.g. when the agent has two kings and one queen, and it knows one opponent has at least one king, while another has three queens, asking for the queens to complete a book is the better option (contrary to what the most cards strategy would choose). When there is no certain option, the most cards strategy is used.

4.1.4 Second-order knowledge

The fourth strategy is the most involved strategy, the agent with this strategy uses second-order knowledge to keep track of what its opponents might know about its own hand cards. For example, by asking for kings, this agent realizes its opponents now know it has at least one king. This second-order knowledge enables the agent to conceal as much of its own cards as possible. The agent first checks if its opponents know something about its own cards. If that is the case, it will conceal the rest of its cards by asking for cards that its opponents already know it has. Ties are broken randomly. If no opponent knows anything about the agent's hand, it will choose an action using the first-order knowledge strategy. In the implementation the agent uses an array to keep track of its hand cards that its opponents know about. Obviously, in some cases the agent can complete a book in one turn using first-order knowledge, in which case it would be rather foolish for the agent to still go with concealing its cards instead of taking that opportunity. Therefore we implemented that an agent with this strategy will still first check whether it can complete a book in one turn, and only after that use the second-order knowledge to pick an action.

5 Experiment

To test whether the strategies performed like we expected, we created a small experiment in which the AI's with different strategies were tested against each other. The experiment answers the question which of the four strategies performs best in a competition against other strategies. Our hypothesis is that the second-order knowledge strategy will give the best performance, closely followed by the first-order knowledge strategy, the most cards strategy will give a much worse performance, and the random strategy will result in the worst performance. In other words, we expect the four strategies to have increasing difficulty levels.

5.1 Methods

For the experiment we adapted the implementation in such a way that it does not have one user and two AI agents, but three AI agents competing with each other. Since there are only three players in the implementation and four strategies, we compared the strategies with each other by running all four possible combinations. Each combination was run 10,000 times, such that each strategy was tested in 30,000 runs in total, in order to get complete and reliable results.

5.2 Results

Strategy	Run 1	Run 2	Run 3	Run 4	Total	Ranking
Random	1059	883	663	-	2605	4
Most cards	1425	1526	-	1090	4041	3
First-order knowledge	7516	-	4372	4384	16272	2
Second-order knowledge	-	7591	4965	4526	17082	1

Table 2: Results of the experiment: number of wins out of 10,000 game runs of each strategy.

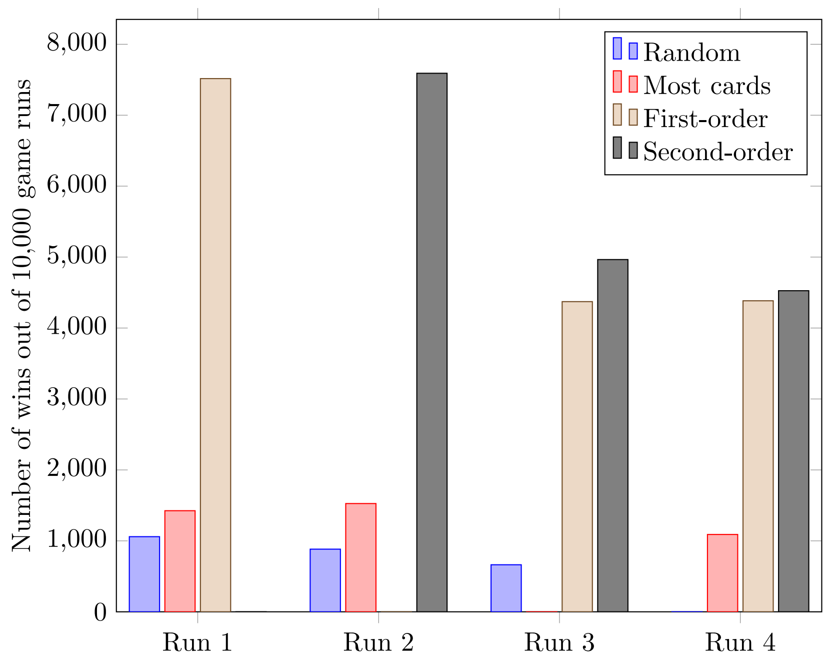


Figure 3: Number of wins of each strategy in each run.

5.3 Interpretation of the results

As can be seen in Table 2 and Figure 3, the results confirm our hypothesis that with increasing knowledge the performance of the strategies also increases. There is a significant difference between the performance of the first two strategies and that of the last two strategies, which shows that using epistemic logic to choose actions significantly increases the chances to win in Go Fish. The concealing strategy that the second-order knowledge adds to the last strategy makes its performance slightly better than the strategy involving only first-order knowledge. The ranking of the strategies holds in every combination of player strategies, which shows that the performance of strategies does not depend on the strategies of opponents.

6 Conclusion

In this project the game *Go Fish!* is analyzed using formal logic and Kripke semantics. The use of Kripke models helps to find efficient strategies for winning the game. A small experiment in which four different strategies were tested against each other showed that using first- or second-order knowledge improves a player's chances of winning significantly. The experiment was performed in the game application that was developed for this project. In the online version of the application a human user can play the game against two AI agents, of which the difficulty level (i.e. the strategy) can be chosen before starting the game. During the game, the user has the option of showing the first-order or second-order knowledge in order to help him choose a good action.

7 Discussion and future research

Due to limitations to our HTML and JavaScript programming abilities, the UI of the application is something that could be improved upon. However, we did manage to visualize the most important features of the game and make the online web application playable.

For future research, it could be interesting to extend the game with extra rules and whether that influences the performance of the four strategies in section 4.1 or if new strategies can be discovered. These extra rules could be one of the following:

- *Asking another player for cards is done by showing a card to that player, instead of asking for it out loud.* This rule results in a more dynamic game in the underlying epistemic logic, in the sense that not every piece of knowledge can be considered common knowledge anymore. Instead, one player may know about cards of another player that a third player knows nothing of.
- *Each card that is drawn from the deck has to be shown to every other player.* That way, it is not the case anymore that every round the knowledge about what a player does *not* have is removed. Instead, all knowledge that already existed about another players hand is still valid. Furthermore, concealing information about the cards that a player has from the beginning of the game becomes more important.
- *The game ends when a player runs out of cards; any cards other players still have count as deduction.* This rule changes a lot about the efficiency of strategies. A very aggressive strategy may be to run out of cards as soon as possible with only one complete book, by revealing a lot of hand cards to other players, in order to make them ask for your cards. Strategies will possibly perform well or badly depending on the strategies of the other players.
- *Players may ask for cards that they do not have.* Although this rule decreases the amount of certain knowledge that players have during the game, asking for cards that one does not have might not be a good strategy, as the player is further away from obtaining a complete book. Players could also tryo to make a guess about whether the player is following a truthful or deceptive strategy.

Another aspect that will improve this *Go Fish!* game analysis is to involve probability theory. The game depends greatly on chance. Therefore, involving probability theory to calculate the success probability of actions would make the analysis more interesting. Instead of removing knowledge about a player having none of a certain card after that player draws a card from the deck, one could change this knowledge into a probability that the new card is that specific card. That probability would depend on the size of the closed deck, and how much cards of that book are already known to be in the hands of players.