The Game of Fluxx: A Benchmark for Goal Reasoning

Ben Wright BENJAMIN.WRIGHT.CTR@NRL.NAVY.MIL
NRC Postdoctoral Fellow, Navy Center for Applied Research in AI, NRL

Michael W. Floyd MICHAEL.FLOYD@KNEXUSRESEARCH.COM
Knexus Research Corporation

Dustin Dannenhauer DDANNENHAUER@NAVATEKLTD.COM
Research Scientist, Navatek LLC

David W. Aha DAVID.AHA@NRL.NAVY.MIL
Navy Center for Applied Research in AI, NRL

Abstract

Goal Reasoning (GR) agents have been deployed in many domains with varying properties, where each agent has typically been tested in at most a few such domains. This has made it difficult to directly compare GR agents. In this paper we propose the card game Fluxx as a domain that can be used for deploying many GR agents. The base game of Fluxx sets the basic game rules and environment (i.e., the cards) but a large collection of expansions exist that introduce new game mechanics and card types. Thus, a single game provides several environments with differing complexity and properties that can be clearly described and defined. Also, Fluxx offers several other interesting properties for GR agents (e.g., rapid goal changes impact longer-term goal planning). We describe the key properties and benefits of Fluxx, and provide a formal description of the game.

1. Introduction & Motivation

Goal Reasoning (GR) agents may perform complex tasks such as discrepancy detection, goal formulation, goal management, or coordination with teammates. Existing research tends to focus on novel contributions for a subset of these GR tasks and, as such, uses evaluation domains with properties that are best suited for studying these tasks (e.g., complexity, observability, action space, involvement of other agents). While this allows for appropriate empirical evaluations of single agents, it results in a wide variety of domains and environments; no evaluation domains exist for evaluating *all* GR agents. This can make it difficult, and often infeasible, to directly compare the empirical performance of GR agents.

The ability to directly compare GR agents, and establish state-of-the-art performance benchmarks, will be critical for developing mature Goal Reasoning systems. While it may not be feasible to select a single benchmark that will satisfy the needs of all GR agents, having a clear benchmark that is well-defined and extendable would provide a significant benefit. Such a domain would allow for a core evaluation environment (i.e., the baseline environment conditions) and various simplifications (e.g., relaxed rules) and extensions (e.g., additional rules, scenarios, or content), designed such that GR agents could be evaluated in some subset of these variants. Thus, an agent could easily be evaluated to determine in which domain variants it performs well or poorly.

We believe the card game Fluxx can provide such a benchmark. In addition to the core game, variants have been released that provide more cards, new game mechanics, and other fundamental changes. Fluxx also provides several properties that are not present in other GR domains, namely rapid goal changes and the need for long-term goal sequence planning. While Fluxx would make for a reasonable GR benchmark environment, we do not anticipate that it would satisfy the evaluation needs of all GR studies or be the only benchmark used for GR agents. Similarly to how we propose evaluating an agent against a variety of Fluxx domain variants, having a suite of such benchmark domains would provide a more robust overall evaluation of GR agents.

In this paper we propose the $N\text{-}M\text{-}Fluxx$ benchmark and describe why we believe it would be suitable for GR agents. In Section 2, we describe the types of domains that have been used to empirically evaluate GR agents. A description of the rules of Fluxx and why we believe it is a good domain is given in Section 3. We formalize the Fluxx game in Section 4 and provide examples of modeling the game using our formalism in Section 5. Finally, we discuss areas of future work in Section 6.

2. Goal Reasoning Domains

Although GR agents have been tested in many domains that often differ substantially in their properties, they can be roughly classified into three primary groups: *agents that operate in variants of traditional planning domains*, *agents that control autonomous vehicles*, and *game-playing agents*.

Many domains used to test GR agents are variants of, or are highly similar to, popular planning domains such as those used in the ICAPS International Planning Competition (McDermott, 2000). This link between GR research and planning domains is expected given than many GR agents use automated planning algorithms as part of their reasoning process. Also, these domains are often more constrained than open-world autonomous systems domains, with discrete state/action spaces, limited environment sizes, and well-defined action and environment models. Although these domains are often inspired by traditional planning domains, they are usually modified to introduce environmental properties (e.g., adversarial behavior, unexpected events, action failure) that are of interest in GR agent investigations.

The Marsworld domain (Dannenhauer et al., 2016) requires an agent to traverse a 2D environment and activate signals (i.e., light sources), with the possibility of signals being deactivated through natural causes. The Mudworld domain (Molineaux & Aha, 2014) is similar, with an agent that can navigate through a 2D environment that may contain muddy regions. The impact of moving through mud is not provided a priori to agents operating in that domain (i.e., not contained in its environment model) so the reduced movement is unexpected. The Arsonist domain (Paisner et al., 2013) is a variant of the classic Blocksworld domain where an adversarial arsonist may unexpectedly set fire to towers of blocks. Although these are only a sampling of the GR domains inspired by automated planning domains, they illustrate the trend of these domains to introduce a single property (or small set of properties) that align closely with the GR agent components being studied;

GR agents that are not designed to improve performance on those GR tasks would be unlikely to be evaluated using these domains.

Compared to the more constrained domains discussed earlier, GR agents that control autonomous systems often operate in more complex environments. These agents are often motivated by the autonomy needs of defense or search-and-rescue applications, and therefore need to operate in higher-fidelity simulation environments. Also, many of these domains involve other agents, both friendly and adversarial, adding reasoning complexity (i.e., these other agents may not always behave according to GR agent expectations) and collaboration requirements (i.e., if a member of a team).

Some domains are defined by simple state spaces (i.e., discrete and grid-like) but can be characterized by a large space of unexpected events that can occur (e.g., encountering other agents, damaged roads, suspicious/hazardous devices), a large space of goals that the agent can pursue, and the preferences of teammates (Floyd et al., 2018). However, reliance on simple 2D domains is not practical for GR agents that control underwater vehicles, given the degrees of freedom, continuous values, and inherent uncertainty of that domain (Wilson et al., 2014). Such higher-fidelity realistic simulation environments have been used for GR agent applications such as squad-based ground patrol (Gillespie et al., 2015), collaborative air combat (Floyd et al., 2017), and naval mine countermeasure missions (Gogineni et al., 2018). One common thread in these domains is that the type of unexpected events encountered are usually rare but serious (e.g., encountering a mine or enemy ship, being attacked, a rapid goal change by teammates, locating an injured person that needs assistance).

While defense and search-and-rescue domains offer more environmental realism, they may not be widely accessible if they are custom designed, have limited API availability, or have distribution restrictions (e.g., if they are restricted to domestic military research). In contrast, computer games often allow for similarly complex environments while also providing greater accessibility. This is beneficial since it provides other agents to compare against, community-developed tools, and other resources. As such, GR agents have been tested for their ability to control bots in several popular computer games including real-time strategy (e.g., Starcraft (Weber et al., 2010), Wargus (Jaidee et al., 2012)) and rogue-like games (e.g., Dungeon Crawl Stone Soup (Dannenhauer et al., 2019)). One limitation of these domains is that the top-level goals are generally highly constrained. These games have a single objective (e.g., eliminate enemies, reach a specified destination) and any goal changes are in service of achieving that objective. Real-world agents may have multiple concurrent goals that partially conflict with each other, especially when an agent can have both personal and team goals (Coman & Aha, 2018).

After examining the existing GR domains, there are properties of each of the three groupings that we believe are desirable. First, simpler domains, such as those inspire by planning competitions, are well understood environments with clear rules and objectives. Second, the more complex military and search-and-rescue domains are beneficial because they involve both friendly and adversarial agents operating in the environment. Finally, the game-based domains offer accessible and open environments that minimize the barrier of entry. We believe Fluxx, at least partially, also provides each of these desirable properties.

3. Fluxx: The Card Game

Fluxx is a multi-player turn-based card game where the rules, possible actions, and goals change over time. Initially, each player is dealt three cards and the remaining cards are used as the *draw pile*. While some cards are visible to all players, *in play* cards, the cards in players' hands are private. At the start of the game there are no concrete goals that serve as end conditions for the game; goals are later added, removed, or changed by players during the course of play. The game proceeds until a player meets the conditions specified on an active (i.e., in-play) goal card. A player's turn proceeds as follows:

1. **Draw** the number of cards required by the current rules of the game. Initially, this is set to one card. If the draw pile runs out, the cards in the discard pile are shuffled and used as the new draw pile.

2. **Play** the number of cards required by the current rules of the game. Initially, this is also set to one card. Fluxx has a variety of different card types, discussed in more detail below.

3. **Discard** cards if the number of cards in the player's hand is above the current card limit, if such a limit exists.

4. **Comply** with the current limit of the number of *Keeper* cards (see below) that the player has active, if such a limit exists.

5. **Perform** any optional actions that in-play cards allow for.

3.1 Fluxx Card Types

The following card types exist in the base version of Fluxx[1] and cards of any type may be played during a player's turn (i.e., the **Play** step above):

- **New Rule**: These cards add new rules to the game that may drastically change what a player must do on their turn. For example, New Rule cards may change the number of cards that must be drawn, the number of cards that must be played, impose a hand limit, or impose a Keeper limit. Additionally, these cards may add certain conditional benefits or penalties to players (e.g., based on a player's hand size). New Rule cards take effect immediately and, if they contradict an existing New Rule card that is in-play, cause that card to be discarded.

- **Keeper**: These cards represent physical items a player has collected. A player adds the played Keeper to their active Keeper collection, the size of which is only limited by any New Rule cards related to Keeper collection sizes. Keepers can be thought of as resources, since many goals require the possession of certain subsets of Keepers.

1. Numerous variants and expansions exist that add new cards types and features to the base game. However, in this paper we will restrict our discussion to the base game.

- **Goal**: These cards define win conditions that all players should attempt to achieve (i.e., a played Goal card effects all players in the game). When a Goal card is played, the existing Goal card is discarded (unless a New Rule card has added a rule that allows multiple active Goals). Goal cards identify a set of Keeper cards that are necessary to win the game, and the game cannot be won until at least one Goal card has been played.

- **Action**: These cards allow players to perform various actions that influence the game in a variety of ways. The type of actions include draw actions (e.g., draw and play two extra cards), hand actions (e.g., trade hands with another player), Keeper actions (e.g., steal another player's Keeper card), rule actions (e.g., remove a New Rule card from play), turn actions (e.g., take another turn), and discard pile actions (e.g., take a card from the discard pile). Unlike the other three card types that remain in play when played, Action cards are discarded after they are played.

The distribution of cards of each type in the base game of Fluxx is 19 Keepers, 23 Actions, 27 New Rules, 30 Goals, and 1 card showing the default game rules (this card is always on the table and never drawn).

3.2 Example of Play

The following outlines a short sample game given as an example by the Fluxx's creators, Looney-Labs[2]. When discussing the game, we will represent the current game rules as the 5-tuple ⟨*draw rule, play rule, current goal(s), hand limit, Keeper limit*⟩. This example game has two players, *Player 1* and *Player 2*, who have each drawn a hand containing 3 cards, with the initial rules ⟨*Draw 1, Play 1, -, -, -*⟩. The suggested number of players ranges from 2 to 6.

- **Round 1:** *Player 1* draws one card into their hand and plays the Keeper card **Music**. *Player 2* draws one card and plays the Keeper card **Chocolate**.

- **Round 2:** *Player 1* draws one card and plays a New Rule of **Draw 5**. The change to the draw amount takes effect immediately, so *Player 1* draws an additional four cards. The rules are now ⟨*Draw 5, Play 1, -, -, -*⟩. *Player 2* draws five cards and plays the Keeper card **The Sun**.

- **Round 3:** *Player 1* draws five cards and plays the New Rule **Play 3**. That rule takes place immediately, so they are able to play two additional cards. The player also plays the Goal **Great Theme Song** (a goal to have the Keepers: Music and Television) and a New Rule **First Play Random** (the first card played each round must be selected randomly by the player to the left). The rules are now ⟨*Draw 5, Play 3 (the first is random), Great Theme Song, -, -*⟩. *Player 2* draws five cards and first plays the Goal **5 Keepers** (a goal to collect five Keepers), which was selected randomly. Next, they play the Keeper **Television** and the new Goal **Hippyism** (a goal to have the Peace and Love Keepers). The rules are now ⟨*Draw 5, Play 3 (the first is random), Hippyism, -, -*⟩.

2. They provide a video of this game being played at their Youtube channel: https://youtu.be/L7y1h7BQCZw

- **Round 4:** *Player 1* draws five cards and plays a randomly selected card, the Goal **Hearts and Minds** (a goal to have the Love and Brain Keepers). *Player 1* then plays the **Peace** Keeper and the **Milk** Keeper. *Player 2* draws five cards and plays a New Rule **Play All** (players must play all cards in their hand) selected at random. *Player 2*, then plays the Action card **Share the Wealth** which requires all Keepers on the table to be collected, shuffled, and dealt out to the players. The redistribution gives *Player 1* the **Peace**, **The Sun**, and **Milk** Keepers, and *Player 2* the **Music**, **Television**, and **Chocolate** Keepers. Next, *Player 2* plays the New Rule **Inflation** that requires a player to draw one additional card any time they draw. To comply with that rule, they draw an additional card (since they only drew five cards to start their turn, not 5+1). *Player 2* then plays the Action **Everybody Gets 1** which requires one card to be drawn for each player but allows *Player 2* to look at the cards and select which player gets which cards. Since the Inflation rule is in play, four cards are drawn (1+1 for each player) and distributed. Next, they play the Action **Zap a Card** that allows them to take any card that is in play and put it in their hand. They select **The Sun** Keeper from *Player 1* and then immediately play it. Finally, they play the Goal **Squishy Chocolate**. Since that goal involves having **Chocolate** and **The Sun** Keepers, both of which *Player 2* has in play, *Player 2* is declared the winner.

- **Example Summary:** The final two rounds (3 and 4) were considerably more complex than rounds 1 and 2 as new rules were added to the game. The first goal to be introduced was **Great Theme Song** in Round 3 by Player 1. In the same round following Player 1, Player 2 played a new goal card **5 Keepers** followed by another goal card **Hippyism**. Each time a goal card is played, it replaces the previous goal card. The goal to win the game was changed three times in Round 3.

3.3 Why Fluxx?

There are several key properties and complexities of Fluxx that we believe make it an interesting GR evaluation domain:

- **Rapid Goal Changes:** Most prior research on GR considers goal changes to be a result of unexpected events or opportunities, which are rare. In some adversarial domains (i.e., the military domains discussed earlier) goal changes can occur due to the unpredictable nature of adversaries. Fluxx takes goal changes one step further. They are not the exception but instead a common and expected occurrence. As GR agents deploy to more complex and long-term scenarios, the level of uncertainty and non-determinism will increase. This results in more situations requiring goal changes to occur. The frequency at which goals change is proportional to the number of goal cards in a given deck and can be controlled based on deck composition (i.e., variants to encourage short-term or longer-term goal reasoning). The depending factor is when and where goal cards are in a given deck. Lastly, unsatisfiable goals can introduced by the addition of goal cards that do not use the Keepers in the deck.

- **Goal Planning:** Existing GR research typically involves, among other reasoning, *goal selection* (i.e., committing to one or more goals) and *planning* to achieve those goals. Fluxx allows for more complex goal selection that more closely resembles *goal planning*. An agent must plan for the goal changes they wish to occur (and anticipate the goal changes caused by other players). Thus, the planning process involves generating a goal plan as well as an action plan (i.e., playing New Rule, Keeper, and Action cards) to achieve those goals.

- **Future Goal Prediction:** Although there are several examples of GR agents that perform plan and goal recognition on other agents, there has only been limited research into predicting how an agent's goals will change over time (e.g., (Borrajo et al., 2016)). Fluxx provides opportunities for such research. For example, if a player performs goal planning then it is possible to recognize the goal plan they are following and then reason on future goals. Using goal recognition can better predict future goal cards the player will play, as well as what New Rules, Keepers, and Actions they may use to support that goal trajectory. While this opens interesting new research directions, it is not required by GR agents that play Fluxx. We believe agents with these capabilities would play Fluxx better. Instead, agents rely on current observable goals and make no attempt to predict future goal changes.
- **Adversarial Goal Changes:** All agents, both friendly and adversarial, are able to change the goals in Fluxx. In the core game, goal changes by other agents are like situations in the real world where someone in power (e.g., a boss, the government) modifies the goals without all affected members agreeing. In such cases, players can either accept the goal change (i.e., attempt to achieve the goal) or ignore the goal and face the repercussions (e.g., losing the game, losing their job, facing legal consequences). From a GR standpoint, this allows for examining agents that deal with unexpected goal changes and then how to reason and respond to them. However, allowing adversaries such direct control over another agent's goals may not always be practical or realistic. As with the rest of Fluxx, we envision variants of the game that can disallow adversaries to change an agents' set of goals. Such a variant gives players interaction (i.e., adding rules, performing actions) but provides more control over their own goals.
- **Changing Rules:** Similar to goals, the rules of the game are not fixed. All players are able to modify the rules such that the way the game is played can be drastically changed over time. This adds an additional difficulty to planning since it also involves planning to change how the game operates.
- **Partial Observability:** The hand of other players and the draw pile are not observable, adding a high degree of uncertainty.
- **Extendability:** Fluxx variants exist that add more cards and additional gameplay mechanics. The game developers provide the rules and card lists online.

4. Formalizing the Fluxx Problem

This section aims to take the previous discussion of Fluxx, presented earlier in this paper, and define it more formally. This is beneficial because it provides an initial formal representation, defines Fluxx-related notation, and allows for a more detailed modeling of games.

4.1 Simplifying the Fluxx Problem

For our initial discussion, we present a constrained variant of Fluxx referred to as *Simplified Basic Fluxx*. This variant reduces some of the complexities of the game. This is close to the most basic version of the game, with successive extensions increasing the scope and complexity (e.g., the full base game of Fluxx, the base game with some subset of expansions). This simple version can serve as an initial baseline to show an agent's ability to play Fluxx before attempting more complex variants.

Simplified Basic Flux Rules:

- The game **does not use Action cards**. This simplifies the game by allowing an agent to focus on following and adding rules (i.e., New Rule cards), collecting resources (i.e., Keeper cards), and planning to achieve goals (i.e., Goal cards).
- The game is **fully observable**. The players know the cards for all players' hands as well as the cards in the deck. This simplification reduces the hidden information and uncertainty involved in a full game.
- Each player **starts with 1 card in their hand**. This simplification reduces the complexity of the game in early rounds.

4.2 The N-M-$Fluxx$ Problem

We represent a game F of Fluxx as an N-M-$Fluxx$ where N is the number of players in the game and M is the maximum number of cards of each card type. Depending on the variant of Fluxx being played (e.g., Basic Fluxx, Simplified Basic Fluxx), the game will be played with a set of Keeper cards $\mathcal{K}$ ($|\mathcal{K}| \leq M$), Goal cards $\mathcal{G}$ ($|\mathcal{G}| \leq M$), New Rule cards $\mathcal{R}$ ($|\mathcal{R}| \leq M$), and Action cards[3] $\mathcal{A}ct$ ($|\mathcal{A}ct| \leq M$).

The game is represented as the tuple:

$$F = \langle Ag, \mathcal{P}_{Ag}, \mathcal{H}_{Ag}, \mathcal{D}is, \mathcal{D}ra, \mathcal{CG}, \mathcal{CR} \rangle \tag{1}$$

The game tuple contains the following information:

- Ag is the set of players in the game ($|Ag| = N$)
- $\mathcal{P}_{Ag}$ is the set of sets of cards currently in play for all players where $P_{Ag} = \bigcup P_a, \forall a \in Ag$
- $\mathcal{H}_{Ag}$ is the set of sets of cards in the hand of players where $H_{Ag} = \bigcup H_a, \forall a \in Ag$
- $\mathcal{D}is$ is the list[4] of cards in the Discard Pile

3. In Simplified Basic Fluxx $\mathcal{A}ct = \emptyset$ (i.e., $|\mathcal{A}ct| = 0$), whereas other variants of Fluxx may introduce sets for other card types.
4. We switch from sets to lists for some of the state information since the ordering of the cards is necessary for certain rules to be applied to them. The simplest example of this is the draw pile where the cards are randomly sorted and then drawn by players using a fixed ordering (unless the pile is reshuffled).

- $\mathcal{D}ra$ is the list of cards in the Draw Pile
- $\mathcal{CG}$ is the list of current goal cards in play
- $\mathcal{CR}$ is the list of current rules in play

Given this representation, a round of Fluxx played by player a proceeds as follows:

1. **Draw**: The player draws d cards from the top of $\mathcal{D}ra$, where d is specified by the draw rule contained in $\mathcal{CR}$. Those cards are added to $\mathcal{H}_a$.
2. **Play**: The player plays p cards from $\mathcal{H}_a$, where p is specified by the play rule contained in $\mathcal{CR}$. For each card c ($c \in \mathcal{H}_a$) that is played, it is removed from the players hand ($\mathcal{H}_a \setminus c$) and the following occurs depending on the card type:
   - **Keeper**: The card is added to the set of cards currently in play for the player ($\mathcal{P}_a \cup c$)
   - **New Rule**: The card is added to the set of active rules ($\mathcal{CR} \cup c$) and existing active rules may be removed based on any rule limits imposed by $\mathcal{CR}$. Any removed rules from $\mathcal{CR}$ are added to the top of $\mathcal{D}is$.
   - **Goal**: The card is added to the active goals ($\mathcal{CG} \cup c$) and existing active goals may be removed based on any goal limits imposed by $\mathcal{CR}$. Any removed goals from $\mathcal{CG}$ are added to the top of $\mathcal{D}is$.
   - **Action**: The effects of these cards come in many forms and may impact one or more elements of F. After the card is played, it is discarded by adding it to the top of $\mathcal{D}is$.
3. **Check for win**: After each **Play**, if $\exists\ a \in Ag\ s.t.\ \mathcal{P}_a$ satisfies any of the active goals in $\mathcal{CG}$, player a is declared the winner. If it is not obvious that one player wins (i.e., more than one player satisfy a goal) play continues. Otherwise, the game continues.

5. Modeling the Fluxx Problem

Now that we have defined the formalism for N-M-$Fluxx$ games, we will provide several examples to illustrate how the information in the model changes during the course of a game.

5.1 The 1-1-$Fluxx$ Problem

For our first example, we examine a game of 1-1-$Fluxx$ using the Simplified Basic Fluxx rules. Since this is the 1-1 version, it only has one player and one card of each type (and no Action cards), so it is a nearly trivial instantiation of the game. However, we use this for illustrative purposes before turning to more complex games.

A 1-1-$Fluxx$ game has one card of each type. If we assume some arbitrary cards $g1$ (a Goal), $r1$ (a New Rule), and $k1$ (a Keeper), the game is initialized with the sets $\mathcal{K} = \{k1\}$, $\mathcal{G} = \{g1\}$, and $\mathcal{R} = \{r1\}$. Each of those cards is placed in the draw pile, $\mathcal{D}ra = \{g1, k1, r1\}$, and no cards are yet in the discard pile, $\mathcal{D}is = \{\}$. Similarly, since no cards have been played yet there are no active Goal or New Rule cards, $\mathcal{CG} = \{\}$ and $\mathcal{CR} = \{\}$.

The game contains a single player $a1$, resulting in $Ag = \{a1\}$. Before the game starts, the player has no cards in its hand and no cards in play, so $\mathcal{P}_{a1} = \{\}$ and $\mathcal{H}_{a1} = \{\}$.

5.1.1 Game Progression of the 1-1-$Fluxx$ *Problem*

Initial Draw: At the start of the game, using Simplified Basic Fluxx rules, each player draws one card from the draw pile and place it in their hand. Thus, the card $g1$ is removed from the draw pile and placed in the hand of player $a1$, $\mathcal{H}_{a1} = \{g1\}$ and $\mathcal{D}ra = \{k1, r1\}$. Since no cards have been played yet, none of the other state information changes.
Round 1: Since there are no New Rule cards in play, $\mathcal{CR} = \{\}$, the player will follow the default rules of the game and *Draw 1* card and *Play 1* card. The player draws the card *k1* from the draw pile and places it in their hand, resulting in $\mathcal{H}_{a1} = \{g1, k1\}$ and $\mathcal{D}ra = \{r1\}$. The player then plays card $g1$ from their hand, resulting in $\mathcal{H}_{a1} = \{k1\}$ and $\mathcal{CG} = \{g1\}$. Since the player does not have the necessary cards in play to satisfy $g1$ (i.e., $\mathcal{P}_{a1} = \{\}$), the game continues.
Round 2: Again, since there are no New Rule cards in play, $\mathcal{CR}$ is still empty, the default rules are followed. The player draws the last remaining card $r1$ from the draw pile and places it in their hand, $\mathcal{H}_{a1} = \{k1, r1\}$ and $\mathcal{D}ra = \{\}$. The player then plays $k1$ which removes it from their hand and places it in their set of in-play cards, $\mathcal{H}_{a1} = \{r1\}$ and $\mathcal{P}_{a1} = \{k1\}$. Assuming $g1$ is to have $k1$ in play, player $a1$ is declared the winner.

5.2 Excerpt of a 2-30-$Fluxx$ Problem as an Example of Play

As we saw in the previous subsection, 1-1-$Fluxx$ using Simplified Basic Fluxx rules does not allow for terribly interesting games and only limited decisions by players. Now, we turn our attention to the example game we previously discussed in Section 3.2. This is a two player game, $Ag = \{a, b\}$, that uses the full Fluxx deck. The deck has a maximum of 30 cards of each type (since there are 30 Goal cards, as discussed earlier), hence the game being 2-30-$Fluxx$. For brevity, we will not define the full sets of $\mathcal{K}$, $\mathcal{G}$, $\mathcal{R}$, and $\mathcal{A}ct$, but they contain the complete set of Fluxx cards that are provided by the base game.

Also, this is a partially observable game. As an outside observer, we do not know which cards are in either player's hand or the ordering of the draw pile. As such, we will use $\{?\}$ to refer to sets we cannot observe[5]. Thus, $\mathcal{D}ra = \{?\}$, $\mathcal{H}_a = \{?\}$, and $\mathcal{H}_b = \{?\}$. Like with the 1-1-$Fluxx$ game, there are no active New Rule cards or Goal cards and neither player has any cards in play ($\mathcal{CR} = \{\}$, $\mathcal{CG} = \{\}$, $\mathcal{P}_a = \{\}$ and $\mathcal{P}_b = \{\}$).

5.2.1 Game Progression of the 2-30-$Fluxx$ *Problem*

Even though the example game in Section 3.2 is relatively simple, it still involves enough game play that we cannot provide the progression of the entire game. We will jump ahead to the most interesting, and complex, round of the game, Round 4.

State after Round 3: Player a has one in-play Keeper card, $\mathcal{P}_a = \{\text{Music}\}$, and player b has three in-play Keepers, $\mathcal{P}_b = \{\text{Chocolate,The Sun,Television}\}$. There is one active goal, $\mathcal{CG} =$

5. An agent could reason about the contents of these sets based on their own observations and the actions of the agents.

$\{$Hippyism$\}$, and three active New Rule cards, $\mathcal{CR} = \{$Draw 5, Play 3, First Play Random$\}$. Two cards are currently in the discard pile, $\mathcal{D}is = \{$Great Theme Song, 5 Keepers$\}$.

Round 4 - Player a:

- The player draws 5 cards, but the specific cards drawn are unknown due to partial observibility.
- Initially, the player must randomly play one of the cards in their hand (due to the First Play Random rule), the Goal *Hearts and Minds*. This causes the current Goal card to be discarded, $\mathcal{CG} = \{$Hearts and Minds$\}$ and $\mathcal{D}is = \{$Hippyism, Great Theme Song, 5 Keepers$\}$.
- The player then chooses to play the Keeper *Peace*, $\mathcal{P}_a = \{$Peace, Music$\}$.
- The player then chooses to play the Keeper *Milk*, $\mathcal{P}_a = \{$Milk, Peace, Music$\}$.

Round 4 - Player b:

- The player draws 5 cards, which are unobservable.
- The card *Play All* is randomly selected and played. This rule replaces *Play 3* which is discarded, $\mathcal{CR} = \{$Draw 5, Play All, First Play Random$\}$ and $\mathcal{D}is = \{$Play 3, Hippyism, Great Theme Song, 5 Keepers$\}$.
- The player plays *Share the Wealth*. This card collects all Keepers in $\mathcal{P}_a$ and $\mathcal{P}_b$ and randomly redistributes them. This redistribution results in $\mathcal{P}_a = \{$Peace, The Sun, Milk$\}$ and $\mathcal{P}_b = \{$Music, Television, Chocolate$\}$. Afterwards, it is discarded, $\mathcal{D}is = \{$Share the Wealth, Play 3, Hippyism, Great Theme Song, 5 Keepers$\}$.
- The New Rule *Inflation* is played, $\mathcal{CR} = \{$Draw 5, Play All, First Play Random, Inflation$\}$. This also requires the player to draw an additional card but the changes to the player's hand and the draw pile are unobservable.
- The player then plays the Action *Everybody Gets 1*, which does not modify any observable information (i.e., only the draw pile and players' hands) and is discarded after use, $\mathcal{D}is = \{$Everybody Gets 1, Share the Wealth, Play 3, Hippyism, Great Theme Song, 5 Keepers$\}$.
- They play the Action *Zap a Card* that allows them to take any card in play and put it into their hand. They select *The Sun* from player a, $\mathcal{P}_a = \{$Peace, Milk$\}$. While mostly unknown, we are certain that *The Sun* $\in \mathcal{H}_b$. *Zap a Card* is then discarded, $\mathcal{D}is = \{$Zap a Card, Everybody Gets 1, Share the Wealth, Play 3, Hippyism, Great Theme Song, 5 Keepers$\}$.
- They play the Keeper *The Sun*, $\mathcal{P}_b = \{$The Sun, Music, Television, Chocolate$\}$. Now that the card has been played, we know that the card picked up (from the previous action) is no longer in the player's hand, *The Sun* $\notin \mathcal{H}_b$.

- Finally, the player plays the Goal *Squishy Chocolate* causing the previous goal to be discarded, $\mathcal{CG} = \{$Squishy Chocolate$\}$ and $\mathcal{D}is = \{$Hearts and Minds, Zap a Card, Everybody Gets 1, Share the Wealth, Play 3, Hippyism, Great Theme Song, 5 Keepers$\}$.

- The *Squishy Chocolate* Goal card requires a player to have the Keepers $\{$Chocolate, The Sun$\}$. Since $\{$Chocolate, The Sun$\} \subseteq \mathcal{P}_b$, player b is declared the winner.

6. Discussion

In this paper we have argued that Fluxx could serve as a valuable benchmark for evaluating Goal Reasoning agents. This stance was motivated by the relative fragmentation in the evaluation domains used for GR research and the difficulty of comparing GR agents and their algorithms. A primary benefit of Fluxx is its extendability. Along with the basic Fluxx game (and the Simplified Basic Fluxx game we described in this paper), there are many variations of the game including *Zombie* Fluxx, *Pirate* Fluxx, and over a dozen others. While some variants add new cards of the basic types (i.e., New Rule, Goal, Action, and Keeper), others introduce new card types such as *Creepers*, *Surprises*, and *UnGoals*. Yet, as we mentioned earlier, we are not claiming Fluxx to be the perfect domain for GR. It contains several shortcomings, it is a turn-based card game (i.e., does not provide continuous sensory information), uses fixed decks (i.e., players will know the cards they may encounter), and has limited cooperative game-play mechanics (i.e., under most circumstances the other players are your competitors). It is our position that Fluxx can be part of a larger suite of well-established benchmark domains used to test GR agents, not the exclusive evaluation domain.

We presented a Fluxx formalism that models the game state and specifies the game parameters. For example, researchers can state that they were experimenting with a 2-30-$Fluxx$ using the core deck, a 4-25-$Fluxx$ using the Zombie Fluxx cards, or a 1-1-$Fluxx$ using Simplified Basic Fluxx rules. This removes the need to infer specific domain conditions other researchers use and sets clear state-of-the-art benchmark results for specific game configurations. Also, we presented several examples of how the Fluxx state, using our formalism, would change during the course of a game. This domain can use similar evaluations and metrics to compare individual reasoning components and agent performance as done previously.

We have also implemented a simple, proof-of-concept GR agent that plays Fluxx. In future work, we plan to improve that agent, describe its implementation and reasoning, and report initial performance results. Our goal is to encode the game play of Fluxx, provide a simple interface to allow other agents to participate in games, and make the software freely available. Our agent will provide an initial performance benchmark.

As mentioned earlier, Fluxx provides several key properties not found in existing GR domains. In future work we plan to develop agents that can operate in domains with those properties (i.e., that can perform long-term goal planning, integrated with existing goal selection and action planning techniques). Also, we will explore the deceptive aspects of Fluxx that arise from partial observability. For example, when a player plays a Goal card, reasoning could assist with predicting whether that is a goal they plan to pursue, a temporary goal (i.e., because they plan to perform another goal change soon), or an attempt to deceive other players into thinking that is their intended goal (or deceiving them into thinking an intended goal is actually a temporary goal).

References

Borrajo, D., Fuentetaja, R., & de la Rosa, T. (2016). Anticipatory Search as Partial Satisfaction Planning with State Dependent Costs. *Proceedings of the 4th Workshop on Goal Reasoning (IJCAI'16)*.

Coman, A., & Aha, D. W. (2018). AI Rebel Agents. *AI Magazine*, *39*, 16–26.

Dannenhauer, D., Floyd, M. W., Decker, J., & Aha, D. W. (2019). Dungeon Crawl Stone Soup as an Evaluation Domain for Artificial Intelligence. *Proceedings of the Workshop on Games and Simulations for AI (held at the 33rd AAAI Conference on AI)*.

Dannenhauer, D., Munoz-Avila, H., & Cox, M. T. (2016). Informed Expectations to Guide GDA Agents in Partially Observable Environments. *Proceedings of the Int'l Joint Conf. on AI* (pp. 2493–2499).

Floyd, M. W., Karneeb, J., Moore, P., & Aha, D. W. (2017). A Goal Reasoning Agent for Controlling UAVs in Beyond-Visual-Range Air Combat. *Proceedings of the 26th Int'l Joint Conf. on AI* (pp. 4714–4721). AAAI Press.

Floyd, M. W., Roberts, M., & Aha, D. W. (2018). Hybrid Goal Selection and Planning in a Goal Reasoning Agent Using Partially Specified Preferences. *Proceedings of the Workshop on Goal Reasoning (held at the 27th Int'l Joint Conf. on AI and the 23rd European Conf. on AI)*.

Gillespie, K., Molineaux, M., Floyd, M. W., Vattam, S. S., & Aha, D. W. (2015). Goal Reasoning for an Autonomous Squad Member. *Proceedings of the Workshop on Goal Reasoning (held at the 3rd Conf. on Advances in Cognitive Systems)* (pp. 52–67).

Gogineni, V. R., Kondrakunta, S., Molineaux, M., & Cox, M. T. (2018). Application of Case-Based Explanations to Formulate Goals in an Unpredictable Mine Clearance Domain. *26th Int'l Conf. on Case-Based Reasoning: Workshop Proceedings - Case-based Reasoning for the Explanation of Intelligent Systems* (pp. 42–51).

Jaidee, U., Munoz-Avila, H., & Aha, D. W. (2012). Learning and Reusing Goal-Specific Policies for Goal-Driven Autonomy. *Proceedings of the 20th Int'l Conf. on Case-Based Reasoning* (pp. 182–195).

McDermott, D. M. (2000). The 1998 AI Planning Systems Competition. *AI Magazine*, *21*, 35–55.

Molineaux, M., & Aha, D. W. (2014). Learning Unknown Event Models. *Proceedings of the Twenty-Eighth AAAI Conference on AI* (pp. 395–401).

Paisner, M., Maynord, M., Cox, M. T., & Perlis, D. (2013). Goal-Driven Autonomy in Dynamic Environments. *Goal Reasoning: Papers from the ACS Workshop* (pp. 79–94).

Weber, B. G., Mateas, M., & Jhala, A. (2010). Applying Goal-Driven Autonomy to Starcraft. *Sixth AI and Interactive Digital Entertainment Conference* (pp. 101–106).

Wilson, M. A., McMahon, J., & Aha, D. W. (2014). Bounded Expectations for Discrepancy Detection in Goal-Driven Autonomy. *AI and Robotics: Papers from the AAAI Workshop*.