Strategic Sorcery: Automated Planning for ‘Magic: The Gathering’

Nicholas Tillo*, Raksha Rehal*, Christian Muise

Queen’s University
Kingston, Canada
{ tillo.nicholas, raksha.rehal, christian.muise}@queensu.ca

Abstract

‘Magic: The Gathering’ (MTG) is a strategic card game that includes complex interactions and competitive play, using a catalogue of over 20,000 unique cards. In this demonstration, we explore how numeric planning can be used to tackle an interesting subset of the game mechanics. Strong gameplay strategies are produced using the ENHSP planner, and our work serves as both an interesting testbed for numeric planners and a foundation for more elaborate strategic card-game settings modelled in PDDL.

1 Introduction

Many trading card games, such as Magic: The Gathering (MTG)[1], provide rich and complex play experiences; they often challenge new players with game states that require nuance and experience to navigate. While MTG games can often be modelled in a simple state space representation, the resulting complexity arising from the actions taken in any single state can make it challenging for players to determine the best strategies to guarantee optimal play. Automating this process using planners can provide valuable insights into the best strategies that may be employed during gameplay and the mechanisms of the game itself (an example intermediate state of the game is shown in Figure 1). It is this motivation that leads us to explore how we might model MTG using planning technology.

The goal of creating an abstract planning model of MTG is to give answers on optimizing play and allowing new players to receive streamlined feedback. The cards demonstrate a variety of complex features that may be tricky to grasp and work with, especially for beginners to the game. Hence, this model aims to teach newcomers what actions are the best to take while using high-level abstractions of core MTG gameplay. It also offers a rich setting for planners to be tested on.

Because much of the game involves numeric aspects (including card strength, player health, and creature attack levels), we opted to use the numeric fragment of PDDL 2.1 (Haslum et al. 2019). In this demo, we cover some of the high-level ideas of the model, along with specific examples of the game dynamics in PDDL.

*These authors contributed equally.

[1]https://magic.wizards.com/

Figure 1: The starting state of the fourth problem file, showing hand, board state, and health.

2 A Model for MTG

We modelled a simplified version of MTG in PDDL in order to generate an optimized sequence of actions such that each move will lead the player to win the game. The model showcases a simple game between (1) the player and (2) the opponent. A ‘win’ for the player is modelled as a PDDL goal that requires the planner to get the opponent’s life total to 0. We are using a very simplified model of the game, which includes changes to traditional MTG gameplay such as restricting possible actions for both the player and opponent, having a limited subset of cards, simplifying the turn structure for both players, and eliminating the spells that are meant to occur instantly.

The game starts at an arbitrary state for the opponent, and an empty or minimally set-up board for the player. The player has a hand of a few cards that they are able to play out in order to defend themselves, and chip away at their opponent’s health at the same time.

The opponent will be represented as having no cards in hand, no lands played, and starting with all their creatures on their board (with summoning sickness). The opponent is

```
(:action attack_opponent
    :parameters (?creature - creature)
    :precondition
      (and
        (> (current_life_total_player) 0)
        (is_owned_by_player ?creature)

        ; checking that it is able to tap
        (not (is_tapped ?creature))
        (not (is_summoning_sick ?creature))

        (current_phase attack)
        (not (is_in_hand ?creature))
        (not (is_dead ?creature))
      )
    :effect
      (and
        (when
          (not (has_vigilance ?creature))
          (is_tapped ?creature)
        )
        (is_attacking ?creature)
        (not (is_blocked ?creature))
      )
)
```

Figure 2: PDDL snippet for the action to attack an opponent.

not able to affect the game state at all, outside of performing routine attacks and blocks as much as possible during their delegated phases. Routine attacks are defined as the opponent attacking with every creature possible on their turn, and routine blocking is defined as blocking each creature possible. It will be a very one-sided fight since the goal of this approach is to model a sequence of actions on the player's end using the core gameplay features of MTG. Hence, the planner will attempt to create a path to victory for the player from the given game state. This is a simplified model of the 2-player aspect to the game, but still provides a compelling setting for planners.

Figure 2 showcases one such action that a correct path may take. This action is a representation of the player's creatures attacking the opponent. The planner may decide whether or not to attack if the corresponding preconditions are true – those being that the player is not dead and the creature is ready to attack (i.e. not in hand, not summoning sick, and not dead). The effects of a creature attacking the opponent is that it gains the status of 'attacking', and it either taps or remains untapped if it has the keyword 'vigilance'. Additionally, this action specifies that the creature is not yet blocked after they are declared to be attacking. This action is just one example of the scope of the model.

A new game state can be defined for solving by the creation of a new problem file that outlines the desired game state, which can be run using the ENHSP 2020 planner (Scala et al. 2016) in conjunction with the original domain file. The output will either produce an optimal plan, or no plan at all if there is no such optimal plan. Figure 1 depicts one such problem file showing the cards that start in the player's hand and the current board state of the game.

3 Discussion

The model was able to reliably find present optimal mid-length plans for the player's victory if the opponent played using the level-zero strategy (Stahl 1993) that was encoded for them – i.e., just attacking and defending wherever possible in a deterministic way. The outputs produced by the planner are able to be analyzed for their strategic meaning; for example, the main priority for the planner while in a disadvantaged state was to protect the player's life total. The planner chose to protect the player by using the player's creatures to block (even if this meant that the creature would die) when in a precarious state where the player's life total was less than that of the opponent. However, when met with an advantaged state, the planner decided to forgo blocking, and instead use the creature to attack, trading the player's life for damage on the opponent. This mimics the infamous MTG adage "health is a resource". This unique way of thinking elevates the player's strategies and causes the player to consider more possible actions that would otherwise not be considered. Thus, the plans produced by ENHSP aligned with verifiable strategic lines of play, even when the problems themselves were extremely simplified. Although the model was implemented for low-complexity game-states, the same strategies and thought-processes output by the planner can be applied with strong success to high-complexity game states. Hence, the outputted plans illustrated realistic and feasible approaches that could be analyzed and studied to better help new players make complex decisions. In spite of the complexity of the mechanisms of MTG, this paper has shown that planning models are increasingly capable of solving difficult problems.

One limitation of the model that remains is the definition of routine attacks and blocks for the opponent. The fact that the opponent attacks and blocks using every creature each turn restricts the number of meaningful situations. This factor may affect the model, and a more dynamic enemy choice structure defining when the opponent should attack or not could be implemented.

Additional future work that can be done to further extend analysis could be to utilize metrics in PDDL, e.g. to minimize the damage taken by the player. This would reflect the different types of playing styles and approaches to MTG, highlighting the various ways one could choose their priorities and strategies in the game.

4 Demonstration

During the live demonstration, we will show the full aspects of the PDDL model, along with actual cards that were modelled, explicitly displaying four major problem files, including the state depicted in Figure 1. Live solving and demonstrated solutions will be done using the VSCode PDDL plugin with ENHSP-2020 planner (Scala et al. 2016). The application of the generated plans will be showcased through a combination of the Cockatrice application[2] and physical MTG cards that will be used as a visual example to show the corresponding actions in MTG.

[2]https://cockatrice.github.io/

References

Haslum, P.; Lipovetzky, N.; Magazzeni, D.; and Muise, C. 2019. *An Introduction to the Planning Domain Definition Language*. Morgan & Claypool. ISBN 9781627058759.

Scala, E.; Haslum, P.; Thiébaux, S.; and Ramírez, M. 2016. Interval-Based Relaxation for General Numeric Planning. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, 655–663. IOS Press.

Stahl, D. O. 1993. Evolution of smartn players. *Games and Economic Behavior*, 5(4): 604–617.