The Complexity of Deciding Legality of a Single Step of Magic: the Gathering

Krishnendu Chatterjee and **Rasmus Ibsen-Jensen**[1]

Abstract. Magic: the Gathering is a game about magical combat for any number of players. Formally it is a zero-sum, imperfect information stochastic game that consists of a potentially unbounded number of steps. We consider the problem of deciding if a move is legal in a given single step of Magic. We show that the problem is (a) coNP-complete in general; and (b) in P if either of two small sets of cards are not used. Our lower bound holds even for single-player Magic games. The significant aspects of our results are as follows: First, in most real-life game problems, the task of deciding whether a given move is legal in a single step is trivial, and the computationally hard task is to find the best sequence of legal moves in the presence of multiple players. In contrast, quite uniquely our hardness result holds for single step and with only one-player. Second, we establish efficient algorithms for important special cases of Magic.

1 Introduction

Magic: the Gathering (henceforth, Magic) is a collectible card game where each player has the role of a mighty wizard that can cast spells like fireballs or summon for instance dragons, angels, or demons. To do so the wizards uses mana they get from lands or in other ways. The objective of each player is to win over the other wizards by killing them.

Legality of a move in a single step. We are interested in establishing complexity bounds for deciding whether a move is legal in a single step of a Magic game. Formally, Magic is a stochastic, imperfect information, zero-sum game over an unbounded number of turns. Each turn is divided into some number of phases and each phase consists of some number of steps, in which typically only one player has a choice. The only exception to the turn-based nature is that some cards require all players to perform some action at the same time.

Cards. Each card in the game represents a specific spell, creature, or land, and has in general very different properties. We describe the typical properties of Magic relevant for this work, and there are many slight deviations which we omit. Each card has a name, a color or colors (or is colorless) and consists of a drawing, illustrating the effect of the card, and a text box stating what the card does. Whenever we mention a card in the text we will write its name in bold font and include a figure with the card. Each card can be in different *zones*. The zones important for this paper are:

1. Nearly all cards start in some **library** from which cards can be drawn. The cards in any library are not visible to any player.
2. The cards in a **hand** of a player consists of the cards drawn from the library but not yet played. Cards in a hand can be played for some cost. The cards in a hand is only visible to that player.
3. The **stack** consists of cards and effects that have been played or triggered but are waiting to take effect. Whenever a card or effect is put on the stack, each player has, in turn, the opportunity to play more cards or effects. Whenever no player wants to add more to the stack, the top card or effect on the stack takes effect. Alternately, if the stack is empty, the next phase or turn starts. The content of the stack is visible to all players.
4. The cards on the **battlefield** consists of the cards that have a direct influence on the game currently and are visible to all players.
5. The cards in the **graveyard** consists of cards that have been used and are visible to all players.

Each card also has a card type. The important card types are:

1. **Instant/sorcery** which, when taking effect, has some immediate or short-term effect on the game and immediately goes to the graveyard.
2. **Artifact/enchantment** which, when taking effect, enters the battlefield and have some long-term effect.
3. **Planeswalker** which, when taking effect, enters the battlefield, with some amount of loyalty. On each of their turns, the controller of the planeswalker can pick an effect, which typically includes an increase or decrease (but not below 0) of loyalty. Planeswalkers can also be attacked to decrease their loyalty. Whenever the loyalty of a planeswalker falls below 1, they go to the graveyard.
4. **Creature** which, when taking effect, enters the battlefield. Most creatures can attack and block in the combat phase and all creatures have a power x and a toughness y, denoted x/y on the cards. The power is the amount of damage the creature would do and the toughness is how much damage a typical creature can take before dying and going to the graveyard. Especially, a creature with 0 toughness dies immediately.
5. **Land** which does not use the stack, but can only be played when the stack is empty. Each player can only play lands in his own turn and only one land on each turn. Most lands are used to generate mana which is often used in the cost of the other card types.

Trivia. Magic is a card game initially published in 1993, played by around twenty million players worldwide and there exist 15.919 different Magic cards currently. Magic has the fourth largest real-life tournament, and other than Poker variants, it has the largest tournament. Magic is the first collectible card game, a category including quite a few different games nowadays. While Magic is a single game, there exists many variants of the game, called formats, with currently

[1] IST Austria, Austria, email: {Krishnendu.Chatterjee,RIbsen}@ist.ac.at

22 different formats receiving some official support. Different formats differs in which cards are legal, basic rules, number of players and their relations (i.e. adversarial or team) and even how the decks of cards played with are constructed. The value of cards can vary greatly from basically nothing to \$27.302 for a single card in regular print.

One player variant: Goldfish. None of the official formats in Magic are single-player, because the main purpose is real-world tournaments with multiple players. However, the relevance of one-player version is that any lower bound on this simple version represents a stronger hardness result. We consider the most well-known one-player variant:

- **Goldfish.** In goldfish, the lone player is playing against a "player" that starts with no cards, never makes any choice and generally never does anything unless forced to – a so called goldfish. Along with the relevance mentioned above, the variant is used to provide a benchmark on how fast a deck can win against a player that does nothing relevant.

Research on Magic. Previous research works have focused on various aspects of Magic, for instance, considering different strategies for collecting Magic-cards ([3]) or used online auctions for Magic cards to test revenue for various auction types ([16]). [5] have shown that for Magic games with at least 6 players, there exist decks of cards, such that for all Turing machines, there exists a polynomial-length sequence of actions by the players, such that after the sequence, a player eventually wins iff the Turing machine terminates[2]. While previous results on complexity consider multiple steps of the game, the problem we consider (i.e., the complexity of a single step) has not been considered before.

Research on other real-life games. There has been some research on the complexity of other real-life games. For instance, [12, 4] considered the popular real-time strategy computer game Star Craft. Also [11, 2] considered Bridge. Complexity of various generalized games have also been considered. For instance, checkers is EXPTIME-complete [18] and Othello PSPACE-complete [15], when the games are generalized to n by n boards. See also [14] for a survey about games and complexity. For the game Phutball, [10] showed that deciding if a legal move that wins immediately exists is NP-complete.

The rules of Magic. Magic has a very complex set of rules: The simple introductory rules used for playing the first few games is 16 pages long, but this does not suffice for the purpose of this article. The full set of rules is 210 pages long and is without pictures ([19]). Therefore, explaining the rules of the game is not part of the goal of this article. Instead, we will explain informally the key rules relating to where the complexity is coming from.

Result of this paper. We present three main results.

1. *Classification of requirements and restrictions.* We classify the requirements and restrictions on Magic cards into a few classes. (Blockers must be declared such that all restrictions are satisfied and as few requirements as possible are broken, see Section 2).
2. *Complexity in general.* We show that the legality of a move in a single step can be decided in coNP in general and is coNP-hard even in *goldfish* games (the single-player variant) measured in the number of objects.
3. *Efficient algorithms for special cases.* For two relatively small and nearly disjoint sets of cards rarely used in tournaments, consisting of 0.5% and 0.3% of all cards respectively, we show that finding a legal move in a single step in Magic games *not* using those cards are in DL (deterministic log-space) and P (PTIME) respectively. It follows from our results that in these cases deciding the legality of a move in a single step also has the same complexity.

[2] The construction has later been improved, see [6, 7, 8], however these texts are posts on forums and less polished

Technical contributions. The key technical aspects are:

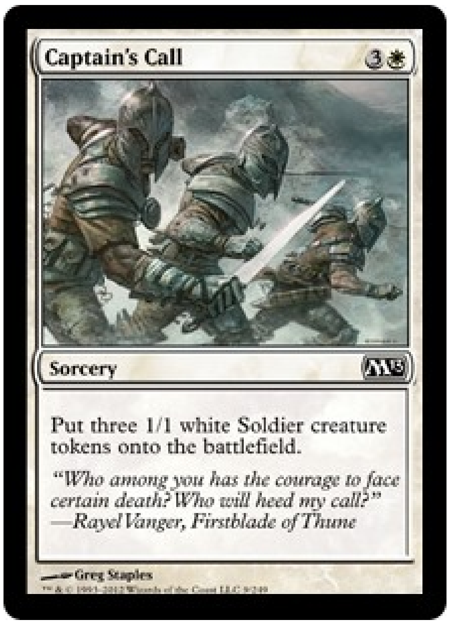
Captain's Call
Sorcery
Put three 1/1 white Soldier creature tokens onto the battlefield.
"Who among you has the courage to face certain death? Who will heed my call?"
—Rayel Vanger, Firstblade of Thune
Greg Staples

Figure 1. Card making tokens: **Captain's Call** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Greg Staples. Image used with permission

1. Non-trivial complexity. Note that we consider games with a constant number of players with a fixed and finite deck. Such restriction in other games lead to trivial complexity (e.g., normal chess is trivial in the sense that only a finite number of configurations can be achieved and an optimal strategy can be found in a constant, but very large, amount of time). The reason this does not happen in Magic is the concept of tokens and the possibility of replaying cards. A token is an object that is not a card. For instance **Captain's Call** creates 3 creatures, though it is only a single card. It is possible to then get **Captain's Call** back to hand (without losing other cards) allowing it to be played again, using some combinations of cards. By playing it many times, any number of tokens can be created.

2. Relationship with classical graph algorithm. A key technical contribution is that we establish a close relationship between the problem we consider and the classical graph algorithm of *min-cost-flow*. While the min-cost-flow problem can be solved in P, we show that the problem we consider requires a natural generalization: along with the standard capacity constraints (which specify that the flow is at most the capacity), we require new constraints that specify that the flow is either 0 or at least a given threshold. Our result shows that this natural generalization makes the problem coNP-complete. On the other hand, if certain cards are not used, we present a many-to-one reduction from our problem to the min-cost-flow problem.

Significance. The main significant implications are:

1. *Magic-specific result.* There exists an online tool for playing Magic, Magic Online ([20]), that checks the legality of a step. However, the tool considers a special case of Magic where at most 200 tokens are allowed per player (i.e., it gives an algorithm for the special case when the size of the input problem is small). In contrast, we first establish complexity of the general problem, showing that it is coNP-hard (even for single-player) and no efficient (polynomial-time) algorithm exists in general (unless $\mathsf{coNP} = \mathsf{P} = \mathsf{NP}$). Second, we present efficient (polynomial-time) algorithms for two special cases, where a small number of cards are not used, but there is no restriction on the number of tokens. Thus we present efficient solution to orthogonal special cases as compared to the online tool.
2. *Uniqueness of complexity.* We consider games with a constant number of players where each has a *fixed and finite* deck. Moreover, we consider the legality of a single step. In most real-life games, either of the above restrictions would lead to trivial computational complexity. The first restriction typically means that the problem size is constant. The second restriction implies that we do not consider strategic choices over multiple steps. In most real-life games the complexity comes from strategic choices (i.e., that maximizes the probability to win over several steps). Quite uniquely our lower bound is for legality of a single-step in a single-player variant of Magic, and the complexity we obtain is neither from the fact that the game is stochastic or imperfect information.
3. *General graph algorithm.* Our technical contribution considers a generalization of the classic min-cost-flow problem on graphs with additional constraints that either the flow is zero or at least a threshold value. We show that such problem is coNP-complete, and show that for real-life game problems (such as special cases of Magic) the classical min-cost-flow algorithm can be used.

2 The complexity of a single step

In this section we consider the complexity of a single step of a Magic game. Concretely we consider the complexity of deciding if in a given step, a given choice is legal. For most types of steps, besides the declare blockers step of the combat phase, this can easily be done in log-space (hence in polynomial time). The reason for the log-space complexity is that the legality of a choice (other than declaration of blockers) is *local*, in the sense that legality can be decided by only considering pairs of objects (i.e. a card c might target another card c' and is legal iff c' can be targeted by c) and counts (i.e. a card c might target k cards, which would be legal if c can have k targets). This can all be done in DL (deterministic log-space). We will thus focus on the declaration of blockers and argue that it is coNP-complete to find a legal declaration of blockers. We describe the problem using the min-cost-flow terminology (since min-cost flow is in P, see [9], and our problem is not, some parts require generalization of min-cost flow).

Min-cost flow. The min-cost flow problem consists of a directed graph $G = (V, E)$, a *capacity function* $c : V \cup E \to \mathbb{N}$ and a weight function $w : E \to \mathbb{Z}$. Also, there is a designated *source state* s and a designated *sink state* t. For a map $M : E \to \mathbb{N}$, let $\Delta_M : V \to \mathbb{Z}$ be $\Delta_M(v) = \sum_{(u,v)\in E} M(u,v) - \sum_{(v,u)\in E} M(v,u)$. A *flow* is a map $f : E \to \mathbb{N}$ such that (1) $\Delta_f(s) \geq 0$; (2) $\Delta_f(t) \leq 0$; and (3) for all $v \in (V \setminus \{s,t\})$ the number $\Delta_f(v)$ is 0 (intuitively, there is a flow from s to t and for other vertices the incoming flow matches the outgoing flow). A flow f is *feasible* if (1) for all $v \in (V \setminus \{s,t\})$ we have $\sum_{(u,v)\in E} f(u,v) \leq c(v)$; and (2) for all $e \in E$ we have $f(e) \leq c(e)$. The *value* of a flow f is $\mathrm{val}(f) = \sum_{e\in E} f(e) \cdot w(e)$. The solution to a min-cost flow problem is a feasible flow with maximum value among all feasible flows. This problem is in P ([9]).

The declaration of blockers step. The declaration of blockers step is a part of the combat phase of a turn. In the combat phase of a turn first a set of creatures A, controlled by P, is declared as *attacking* P' by P (in general with more players each of the creatures can attack different players other than P). For the goldfish format, the player P is the goldfish player (in our lower bound), and there exist cards that ensure that all creatures must be declared attacking (this is the only option for the goldfish player). The creatures controlled by P' is the set B. After this step, if A is not empty, the defending player P' (in general, a player is defending if he is attacked by some creature) can, for each creature $a \in A$ that attacks him, declare that some, perhaps empty, subset $B^a \subseteq B$ is going to block a. We will let B^A be the set $\bigcup_{a\in A} B^a$ of blocking creatures. For any $b \in B$, we also define b^A as $\{a \mid b \in B^a\}$, i.e. the creatures b blocks. The complexity of finding a legal declaration of blockers comes from rule 509.1c ([19]), which states, paraphrased, that a declaration of blockers is legal if it satisfies all *restrictions*, while maximizing the number of satisfied *requirements*.

The base case: No restrictions or requirements. If there are no requirements or restrictions involved, each creature in A can be blocked by any number of creatures in B, but each creature in B can only block one creature of A and any such choice of blockers is legal. Using terminology from min-cost flow, the graph G is a bipartite graph, with the source on the right side and the sink on the left. Each state (besides the sink) on the left side corresponds to a creature in B and each state (besides the source) on the right side corresponds to a creature in A, and the states on the left side has capacity 1 (besides the sink, which has infinite capacity) while the states on the right have infinite capacity. The source has an outgoing edge to each state in B, each state in B has an outgoing edge to each state in A and each state in A has an outgoing edge to the sink. The edges between A and B each have capacity 1 and the edges to/from the sink/source have each infinite capacity. Also, each edge has weight 0.

Restrictions and requirements. The complexity of the problem comes from the requirements and restrictions. We will explain requirements and restrictions in how they modify the min-cost flow instance from the base case.

Restrictions. Rule 509.1b ([19]) states that for a declaration of blockers to be legal, every restriction must be satisfied. A restriction is statement that says that a specific creature cannot block/be blocked unless some condition is met. This, in itself, does not cause any complexity, since for a fixed declaration of blockers, checking if a fixed restriction is satisfied is easy. We now consider the following five types of restrictions[3]:

1. **Local restrictions.** Local restrictions are restrictions of the form: $a \in A$ cannot be blocked by $b \in B$ (i.e. a local restriction is satisfied if it is satisfied for each pair of affected creatures). Concretely, a could have flying and b could have neither reach or flying. A local restriction between $a \in A$ and $b \in B$ can be modeled in our min-cost flow instance by removing the edge (a, b). For instance,

[3] To our knowledge, all restrictions fit into precisely one type

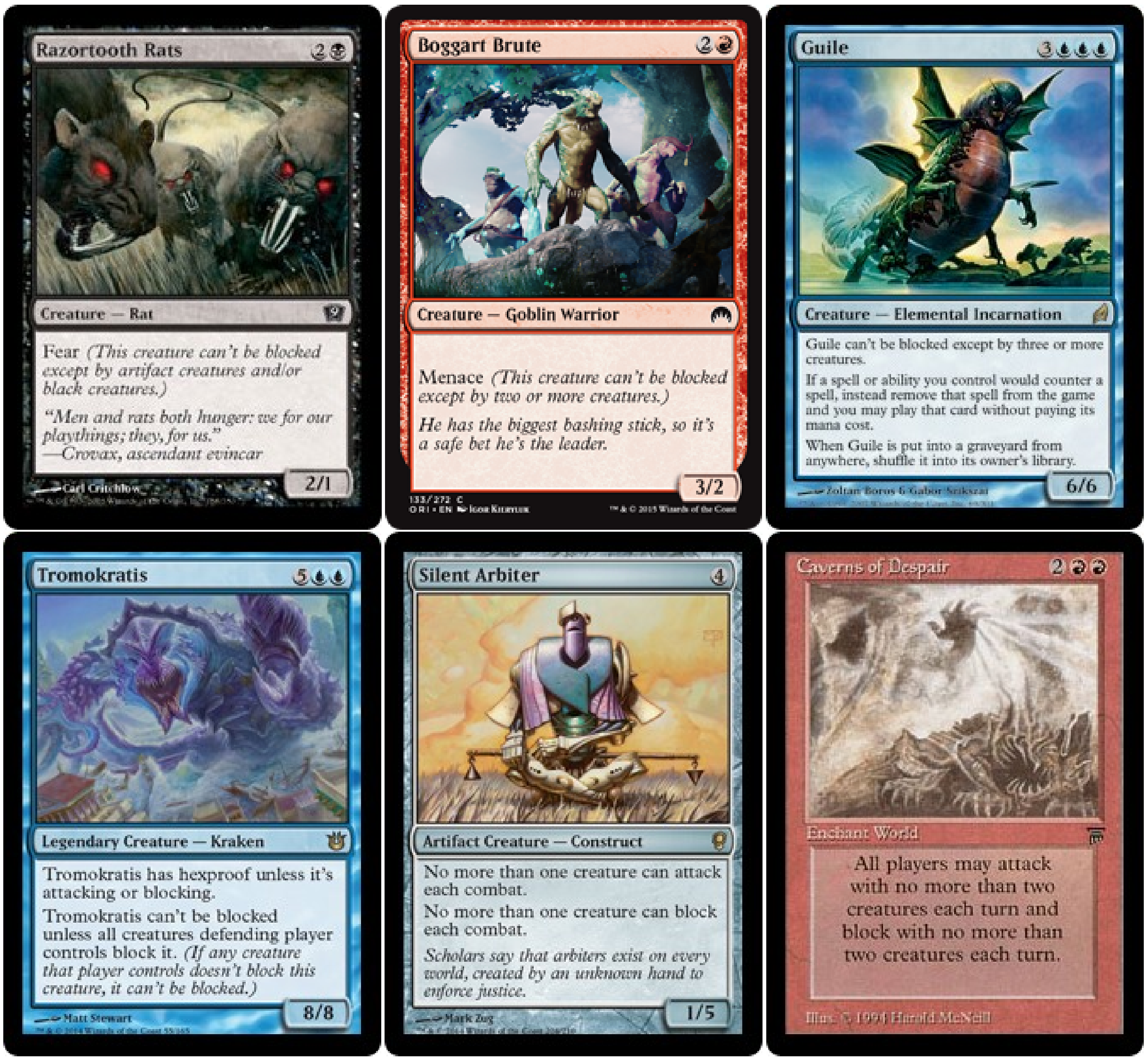
Razortooth Rats
Creature — Rat
Fear (This creature can't be blocked except by artifact creatures and/or black creatures.)
"Men and rats both hunger: we for our playthings; they, for us." —Crovax, ascendant evincar
2/1
Boggart Brute
Creature — Goblin Warrior
Menace (This creature can't be blocked except by two or more creatures.)
He has the biggest bashing stick, so it's a safe bet he's the leader.
3/2
Guile
Creature — Elemental Incarnation
Guile can't be blocked except by three or more creatures.
If a spell or ability you control would counter a spell, instead remove that spell from the game and you may play that card without paying its mana cost.
When Guile is put into a graveyard from anywhere, shuffle it into its owner's library.
6/6
Tromokratis
Legendary Creature — Kraken
Tromokratis has hexproof unless it's attacking or blocking.
Tromokratis can't be blocked unless all creatures defending player controls block it. (If any creature that player controls doesn't block this creature, it can't be blocked.)
8/8
Silent Arbiter
Artifact Creature — Construct
No more than one creature can attack each combat.
No more than one creature can block each combat.
Scholars say that arbiters exist on every world, created by an unknown hand to enforce justice.
1/5
Caverns of Despair
Enchant World
All players may attack with no more than two creatures each turn and block with no more than two creatures each turn.

Figure 2. The cards are sorted from left to right. First row - cards with various types of non-capacity restrictions:
(1) Local restriction: **Razortooth Rats** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Carl Critchlow. Image used with permission;
(2) Local counting restriction $k = 2$: **Boggart Brute** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Igor Kieryluk. Image used with permission;
(3) Local counting restriction $k = 3$: **Guile** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Zoltan Boros & Gabor Szikszai. Image used with permission;
Second row - continued:
(4) Local counting restriction $k = |B|$: **Tromokratis** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Matt Stewart. Image used with permission;
(5) Global counting restriction $k = 1$: **Silent Arbiter** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Mark Zug. Image used with permission;
(6) Global counting restriction $k = 2$: **Caverns of Despair** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Harold McNeill. Image used with permission.

Razertooth Rats creates a local restriction with all non-artifact non-black creatures.

2. **Local counting restrictions.** Local counting restrictions are restrictions of the form: $a \in A$ cannot be blocked except by k or more creatures. Here, concretely, there exists cards for which k is 2 (i.e. cards with the ability **Menace**, like **Boggart Brute**), 3 (there are six cards, for instance **Guile**) and $|B|$ (the card **Tromokratis**). Local counting restrictions have no nice interpretation in min-cost flow terminology (note that there must be something for the problem to be coNP-complete, since min-cost flow can be solved in P, see [9]). A local counting restriction on $a \in A$ corresponds to saying that a have either flow 0 or flow greater than k in min-cost flow.

3. **Global counting restrictions.** Global counting restrictions are restrictions of the form: no more than k creatures can block. Here, concretely, there exists cards for which k is 1 (the cards **Silent**

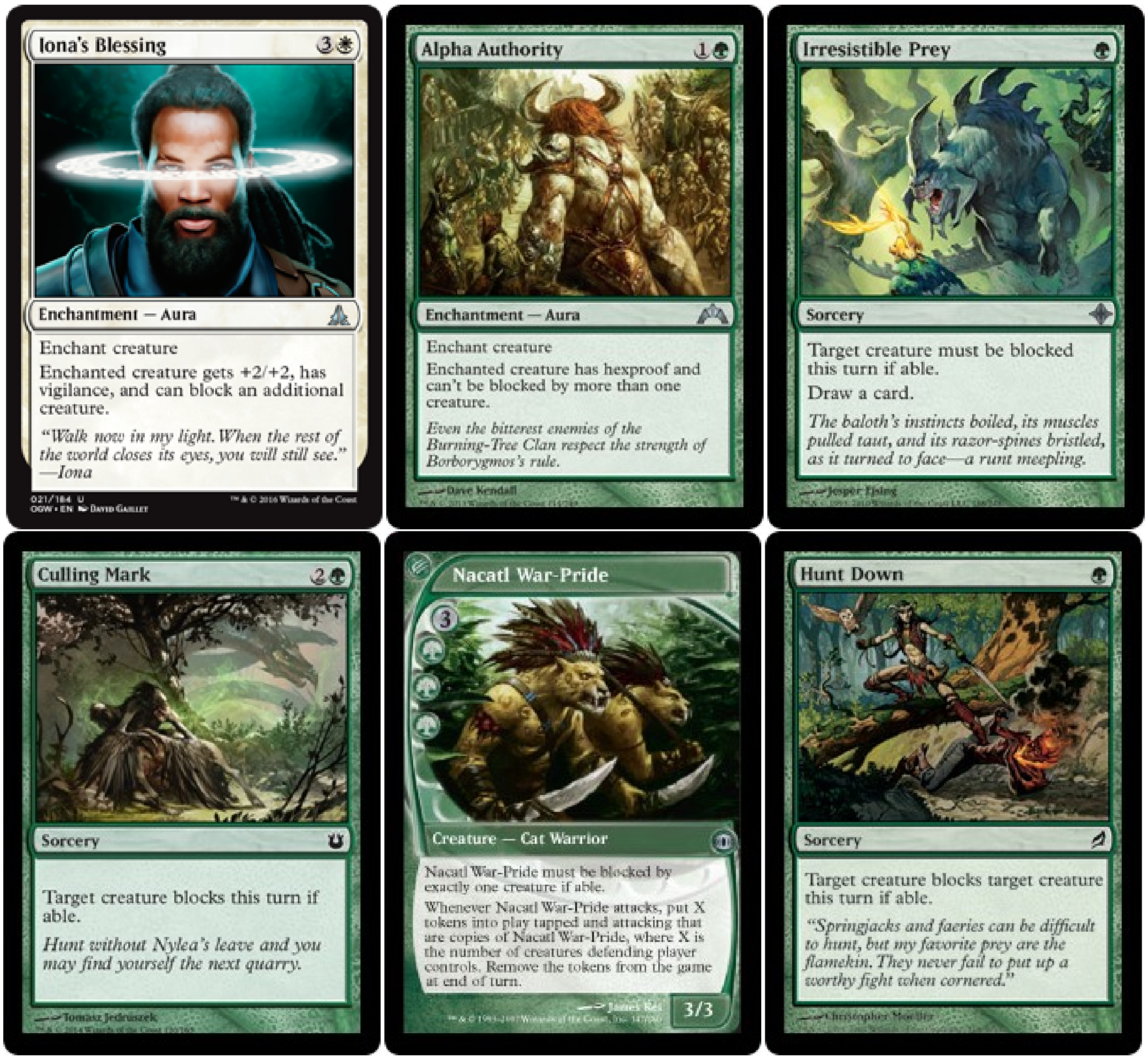
Iona's Blessing
Enchantment — Aura
Enchant creature
Enchanted creature gets +2/+2, has vigilance, and can block an additional creature.
"Walk now in my light. When the rest of the world closes its eyes, you will still see." —Iona
Alpha Authority
Enchantment — Aura
Enchant creature
Enchanted creature has hexproof and can't be blocked by more than one creature.
Even the bitterest enemies of the Burning-Tree Clan respect the strength of Borborygmos's rule.
Irresistible Prey
Sorcery
Target creature must be blocked this turn if able.
Draw a card.
The baloth's instincts boiled, its muscles pulled taut, and its razor-spines bristled, as it turned to face—a runt meepling.
Culling Mark
Sorcery
Target creature blocks this turn if able.
Hunt without Nylea's leave and you may find yourself the next quarry.
Nacatl War-Pride
Creature — Cat Warrior
Nacatl War-Pride must be blocked by exactly one creature if able.
Whenever Nacatl War-Pride attacks, put X tokens into play tapped and attacking that are copies of Nacatl War-Pride, where X is the number of creatures defending player controls. Remove the tokens from the game at end of turn.
3/3
Hunt Down
Sorcery
Target creature blocks target creature this turn if able.
"Springjacks and faeries can be difficult to hunt, but my favorite prey are the flamekin. They never fail to put up a worthy fight when cornered."

Figure 3. Two first cards of first row - cards that can create capacity restrictions:
(1) Increase blocker capacity by 1: **Iona's Blessing** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by David Gaillet. Image used with permission;
(2) Set capacity of attacker to 1:
Alpha Authority ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Ron Spencer. Image used with permission.
Last card on first row and cards in second row - cards with various kinds of requirements:
(1) Generic attacker requirement: **Irresistible Prey** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Jesper Ejsing. Image used with permission;
(2) Generic blocker requirement: **Culling Mark** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Tomasz Jedruszek. Image used with permission;
(3) Special generic requirement: **Nacatl War-Pride** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by James Kei. Image used with permission;
(4) Specific requirement: **Hunt Down** ©2016 Wizards of the Coast LLC in the USA & other countries. Illustration by Christopher Moeller. Image used with permission.

Arbiter and **Dueling Grounds**) and 2 (the card **Caverns of Despair**). While there is no straightforward interpretation in min-cost flow, the global counting restrictions are easy to handle, by simply considering each creature or pair of creatures in B (depending on whether k is at least 1 or 2) as the set of creatures that can block and then solving the problem with only those creatures in B (except that any **Tromokratis** cannot be blocked in this case, assuming that $|B| > 2$ initially).

4. **Blocker capacity restriction.** Blocker capacity restrictions are restrictions of the form: $b \in B$ must be such that $|b^A| \leq k$. By default there is a blocker capacity restriction on each creature for $k = 1$, but k can be changed to any natural number[4], using for instance, a sufficiently large number of **Iona's Blessing**. A blocker capacity restriction on $b \in B$ with $k \in \mathbb{N}$ is modeled in min-cost

[4] It can also be infinite. However, there is no difference between infinite and $k \geq |A|$

flow by having a capacity of k on b.

5. **Attacker capacity restriction.** Attacker capacity restrictions are restrictions of the form: $a \in A$ must be such that $|B^a| \le k$. Currently k can only be 1, which can be achieved by for instance using **Alpha Authority**. A attacker capacity restriction on $a \in A$ with $k \in \mathbb{N}$ can be modeled in min-cost flow with capacity of k on a.

Requirements. Rule 509.1c ([19]) states that the declaration of blockers must maximize the number of requirements satisfied, while not breaking any restrictions. A requirement is an ability that says that a specific creature must block/be blocked (perhaps by a specific set of creatures). In min-cost flow the requirements can be modeled as weights on edges[5]. We focus on three requirements[6]:

1. **Generic attacker/blocker requirements.** Generic attacker (resp. blocker) requirements are parameterised by an attacker $a \in A$ (resp. blocker $b \in B$) and are satisfied if a is blocked (resp. if b blocks). Generic attacker (resp. blocker) requirements can be modeled in min-cost flow by having two edges pointing to a from the source (resp. away from b to the sink), one with unbounded capacity and 0 weight and another with capacity 1 and weight equal to the number of times the creature is affected by the requirement. The requirement could for instance be caused by **Irresistible Prey** (resp. **Culling Mark**).

2. **Special generic requirements** are parametrised by an attacker $a \in A$ and are satisfied if a is blocked by exactly one creature. The requirement has no simple interpretation in min-cost flow, but any flow that has a flow of 1 through a satisfies the requirement. The requirement is only in effect if a is a **Nacatl War-Pride**.

3. **Specific requirements** are parametrised by an attacker $a \in A$ and blocker $b \in B$ and are satisfied if $b \in B^a$. Specific requirements could for instance be caused by **Hunt Down**. We can model k specific requirements between $a \in A$ and $b \in B$ by having a weight of k on the edge from a to b in min-cost flow.

Remark 1 Finding the categories. *Categorizing the requirements and restrictions of all cards in Magic is a demanding task (because of the number of cards). To do so we made a case analysis with many thousands of cases. We do not explicitly include the very technical case analysis, but only the outcome, because of the huge size of the case analysis.*

2.1 Complexity bound in general case

Inclusion in coNP. For a fixed declaration of blockers, it is easy to check that all restrictions are satisfied. In case any restriction is not satisfied, it is easy to find a declaration of blockers satisfying all (concretely, not declaring any blockers always satisfies all restrictions). It is also easy to count the number of requirements satisfied. Hence, the given declaration of blockers is legal precisely if no declaration can be found that satisfies more requirements while satisfying all restrictions. Note that a declaration of blockers can be described as the sets B^a for each a. This can be done in $O(|A| \cdot |B|)$ space, giving a polynomial sized witness. Note that even in case of many players, since the declaration of blockers is a turn-based step, the coNP upper bound holds.

coNP-hardness. Our reduction will be from exact cover by 3-sets. The exact cover by 3-sets problem is as follows: A set of elements E and a set $S \subseteq 2^E$ of sets of elements of E is given, such that $|s| = 3$ for each $s \in S$, and the problem is to decide if there exists a subset S' of S such that $\bigcup_{s \in S'} s = E$ and $s \cap s' = \emptyset$ for all $s, s' \in S'$. The exact cover by 3-sets problem is NP-complete ([13]).

Consider an instance of the exact cover by 3-sets problem and we will construct a combat instance and a declaration of blockers which is not legal iff there exists a satisfying set S' for the exact cover by 3-sets instance. There is a special attacker **Tromokratis**[7], which has, together with some arbitrary creature $b \in B$, been targeted by **Hunt down** $|E| - 1$ times (hence, blocking it with all creatures in B will satisfy $|E| - 1$ requirements). Besides that the attackers will play the role of the sets of S and the blockers the elements of E. Each attacker a, besides the **Tromokratis**, is a **Guile**[8] and will correspond to a set $(e_1, e_2, e_3) \in S$ and **Hunt Down** has been cast on a and the blocker b_1 corresponding to e_1 (so that there is a specific requirement between a and b_1). Similarly for a and e_2 and a and e_3. Hence, blocking a with the creatures corresponding to e_1, e_2 and e_3 will satisfy 3 requirements. Each blocker can block only 1 creature (as is default) and each attacker can be blocked by any number of creatures (as is default). Blocking **Tromokratis** with all creatures is not legal iff there exists an exact cover by 3-sets.

Note that we are giving creatures to the goldfish player. This is sometimes done even in real-life tournament Magic games, for instance it often happens that a player of the Oath of Druids deck will give creatures to his opponent.

Remark 2 How to make the setup in a concrete game of Magic: *This remark is meant for people familiar with the rules of Magic and we do not include illustrations of the cards used. Play* **Imperious Perfect**, **Intruder Alarm**, *2* **Birds of Paradise** *(these first cards gives an arbitrary amount of mana),* **Djinn Illuminatus**, **Tromokratis**, **Concordant Crossroads**, **Vedalken Orrey**, **Fumiko the Lowblood** *and* **Guile**. **Djinn Illuminatus** *allows us to copy spells an arbitrary number of times. On the opponents turn use* **Spitting Image** *to create enough* **Guile***s,* **Hunt Down** *to make the requirements and* **Donate** *to give the attackers away (the cards can be played at that point because of* **Vedalken Orrey***).* **Fumiko the Lowblood** *and* **Concordant Crossroads** *makes the creatures attack, without choice.*

Theorem 1 *The complexity of a single step in a Magic game is coNP-complete, and the hardness result holds even in the special case of goldfish.*

2.2 Special cases with better complexities

In this section we consider several special cases, and show that they have much better complexity. Note that the upper bounds we get in this section is for any number of players because the declare blockers step is turn-based.

Restriction 1: Neither local counting restrictions (except for Tromokratis) nor Nacatl War-Pride. As argued above besides lo-

[5] Except for the requirement caused by **Nacatl War-Pride**, which does not have a simple interpretation in min-cost flow.

[6] To our knowledge, all requirements fit into precisely one type

[7] which must either be left unblocked or blocked by all

[8] which must either be left unblocked or blocked by 3 or more creatures

cal and global counting restrictions and **Nacatl War-Pride**, every requirement and restriction can be encoded in the standard min-cost flow problem. We explain how to handle **Tromokratis** and global counting restrictions.

- *Handling* **Tromokratis**. For any number $k \leq |A|$, we can find the optimal solution such that precisely k **Tromokratis** are blocked, because of the following: For each **Tromokratis** $a \in A$, the number of requirements satisfied by blocking a with all creatures in B, is independent of how else the creatures in B block. Therefore, we can sort the **Tromokratis** after how many requirements are satisfied, block the first k with all creatures in B, and then solve the sub-problem where the set of attackers is A, except for the **Tromokratis**, and each creature in B can block k less creatures.

- *Handling global counting restrictions.* We can handle global counting restrictions parametrised with k (even in the presence of local counting restrictions) straightforwardly since $k \in \{1, 2\}$ (it is not clear if the problem is in polynomial time for k as a parameter, but k is at most 2 for global counting restrictions in Magic). That is, we simply guess the creatures in B^A and then solve the problem with only those creatures. We can, in the presence of global counting restrictions also handle local counting restrictions easily.

For an in depth complexity analysis see Section 2.3, where we show the complexity is $O(nm^2 \log n + mn^2 \log^2 n + n^2 \log C)$ time, where, in our min-cost flow instance, n is the number of states and m the number of edges (i.e. $n = 2 + |A| + |B|$ and $m \leq |A| + |B| + |A| \cdot |B|$), and C is the greatest weight on an edge (and thus C is at most the number of requirements). Hence, the problem is in polynomial time if none of the 37 cards (of which 31 have or grants local counting restriction 2, i.e. Menace) that have or grants a local counting restriction, besides **Tromokratis**, but also including **Nacatl War-Pride**, is in the game. None of these 37 cards, which is less than 0.3% of the distinct Magic cards, are heavily used. Menace is a new ability that might be used in the future.

Restriction 2: Removing requirements. Another simple way to make the problem easier is to remove the cards that have or grants requirements. There are 63 cards, which is less than 0.5% of all distinct Magic cards, that causes requirements on blocking, none of which is used often. Note that only the card **Nacatl War-Pride** is in the 63 cards removed from this special case as well as the 37 cards removed in the above considered special case. A declaration of blockers is then legal iff it satisfies all restrictions. For each type of restriction, as describe above, it is easy to check in DL if the restriction is satisfied. Hence, this special case is in $\mathsf{DL} \subseteq \mathsf{P}$.

Restriction 3: Removing tokens. Another simple way is to remove all roughly 830 producers of tokens (i.e. creatures which are not cards), which corresponds to less than 6.1% of all distinct Magic cards, some of which are used heavily. At this point, the number of creatures in play is bounded by the number of Magic cards in existence and the problem can be solved by considering each possible declaration of blockers. Since the problem is bounded, it is trivial. (We consider this restriction, since it is a reasonably obvious way to make the problem simpler – it does not lead to an efficient algorithm though). A variant of Restriction 3 is used in the online implementation of Magic ([20]) where at most 200 tokens can be created per player (after which no more tokens appear when one tries to make any).

Theorem 2 *The complexity of a single step in a Magic game with any of the Restriction 1-3, no matter the number of players, can be solved in polynomial time.*

Also note that for Restriction 1-3 it follows from our results that the legality of a given move in a single step can be decided in the same complexity as mentioned above.

2.3 In depth analysis of Restriction 1

We will, in this section, give a more in depth complexity analysis, in case there are neither local counting restrictions (except for **Tromokratis**) nor **Nacatl War-Pride** (that is, of Restriction 1). To get a better complexity bound we consider the min-cost flow instance created in this case in Section 2.2. Observe that it has $n = |A| + |B|$ states. Also, there is an edge from $a \in A$ to $b \in B$ unless there is a local restriction between them. Hence, $m = |A| \cdot |B| - c$, where c is the number of local restrictions between different pairs. Parametrising like this, Orlin's strongly polynomial algorithm for minimum cost flow ([17]) runs in time $O(m^2 \log n + mn \log^2 n)$.

We consider three cases:

1. If there is a global counting restriction of k (which is either 1 or 2), we can as explained in Section 2.2, consider each creature or pair of creatures in B in turn and solve the resulting instance when B only consists of those creatures. This takes n^2 times the time for the case when B consists of two creatures.

   Remark 3 *Observe that if the case where $|B| = 2$ can be solved in $O(n \cdot f(n) + g(n))$ time, where $f(n)$ is a sum of terms, each of which is at least constant, then this case can be solved in time $O(nm \cdot f(n) + n^2 g(n))$. This is because the time for a pair of states u, v is then $O((d_u + d_v) \cdot f(d_u + d_v) + g(d_u + d_v))$, where d_u and d_v is the degree of u and v respectively. Hence,*

$$\begin{aligned}
&\sum_{u,v} O((d_u + d_v) \cdot f(d_u + d_v) + g(d_u + d_v)) \\
&\quad \leq O(n^2 g(n)) + f(n) \cdot \sum_{u,v} O((d_u + d_v)) \\
&\quad = O(n^2 g(n)) + f(n) \cdot \sum_{u} O(n \cdot d_u + m) \\
&\quad = O(nmf(n) + n^2 g(n))
\end{aligned}$$

   This shows that using Orlin's method ([17]), this case would use $O(m^2 n^2 \log n + m^2 n \log^2 n) = O(m^2 n^2 \log n)$ time.

2. If **Tromokratis** is not in the game and there are no global counting restrictions, then it is easy to solve it in time $O(m^2 \log n + mn \log^2 n)$, since it is a standard flow problem.

3. Otherwise, if there is no global counting restrictions and **Tromokratis** is in play then we can guess the number Δ of **Tromokratis** each creature should block (observe that this is bounded by the number of creatures in A), by simply trying all options. We can then find the Δ **Tromokratis** that, if blocked, will give the most satisfied requirements, by sorting them after how many requirements will be satisfied (note that the number of satisfied requirements is independent of which other creatures a creature block, and thus doing it greedily will find the best set). We can then reduce the capacity of each blocker by Δ, remove the **Tromokratis** and solve the resulting instance in

$O(m^2 \log n + mn \log n)$ using Orlin's algorithm ([17]). This requires $O(nm^2 \log n + mn^2 \log^2 n)$ time.

Observe that the worst case time, if we use Orlin's algorithm ([17]) in the first case, will be the time for the first step. That said, there exists specialized algorithms for this case, where one side is of constant size, see [1], using time $O(n + \log C)$ where C is the greatest weight. Specifically, C is in this case the maximum number of times any single state (resp. pair of states) are effect by a generic (resp. specific) requirement. This will ensure that the time for case 1 is then $O(nm + n^2 \log C)$ and the total time is $O(nm^2 \log n + mn^2 \log^2 n + n^2 \log C)$.

3 Conclusion

In this work we establish the complexity of deciding the legality of a move in a single step of Magic. In sharp contrast to existing real-life games, where legality of a move in single step is trivial, we establish coNP-completeness for legality of a move in a single-step of a single-player variant of Magic, which is quite unique and should be of wide interest. Moreover, our result is established by showing a close connection with a generalization of the min-cost-flow problem, which is also of independent and general interest. While we show that in general the legality of a move in a single-step is coNP-complete, we present efficient algorithms for special cases which are different from existing tools but these special cases are widely used. Hence our algorithms are also practically relevant for automated tools to analyzed important special cases in Magic.

References

[1] R. K. Ahuja, J. B. Orlin, and C. Stein. Improved Algorithms for Bipartite Network Flow. *SIAM J. Comput.*, 23(5):906–933, 1994.

[2] E. Bonnet, F. Jamain, and A. Saffidine. On the Complexity of Trick-Taking Card Games. In *IJCAI 2013*, 2013.

[3] R. A. Bosch. Optimal Card-Collecting Strategies for Magic: The Gathering. *The College Mathematics Journal*, 31(1):pp. 15–21, 2000.

[4] M. Buro and A. Kovarsky. Concurrent Action Execution with Shared Fluents. In *AAAI 2007*, pages 950–955, 2007.

[5] A. Churchill. Magic: the Gathering is Turing Complete. http://tinyurl.com/olg28d5, 2012.

[6] A. Churchill et. al. Magic: the Gathering is Turing Complete (forum page 1). http://tinyurl.com/pv3n2lg, 2012.

[7] A. Churchill et. al. Magic: the Gathering is Turing Complete (forum page 2). http://tinyurl.com/naq9jmc, 2012.

[8] A. Churchill et. al. Magic: the Gathering is Turing Complete (forum page 3). http://tinyurl.com/ow7obh6, 2014.

[9] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[10] E. D. Demaine, M. L. Demaine, and D. Eppstein. Phutball Endgames are Hard. *CoRR*, cs.CC/0008025, 2000.

[11] I. Frank and D. A. Basin. A theoretical and empirical investigation of search in imperfect information games. *Theor. Comput. Sci.*, 252(1-2):217–256, 2001.

[12] T. Furtak and M. Buro. On the Complexity of Two-Player Attrition Games Played on Graphs. In *AIIDE 2010*, 2010.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[14] R. A. Hearn and E. D. Demaine. *Games, Puzzles, and Computation*. A. K. Peters, Ltd., Natick, MA, USA, 2009.

[15] S. Iwata and T. Kasai. The Othello game on an n x n board is PSPACE-complete. *Theoretical Computer Science*, 123(2):329 – 340, 1994.

[16] D. Lucking-Reiley. Using Field Experiments to Test Equivalence between Auction Formats: Magic on the Internet. *The American Economic Review*, 89(5):pp. 1063–1080, 1999.

[17] J. Orlin. A Faster Strongly Polynomial Minimum Cost Flow Algorithm. In *STOC '88*, pages 377–387, 1988.

[18] J. M. Robson. N by N Checkers is Exptime Complete. *SIAM Journal on Computing*, 13(2):252–267, 1984.

[19] Wizards of the Coast. Magic: the Gathering Comprehensive Rules. http://tinyurl.com/qf6hlx4, 2015.

[20] Wizards of the Coast. Magic: the Gathering Online. http://tinyurl.com/h95z67n, 2016.