Automatic Synthesis of Generalized Winning Strategies of Impartial Combinatorial Games*

Extended Abstract

Kaisheng Wu, Yong Qiao, Kaidong Chen, Fei Rong, Liangda Fang, Zhao-Rong Lai, Qian Dong
Dept. of Computer Science, Jinan University
Guangzhou, China

Liping Xiong
Dept. of Computer Science
South China Normal University
Guangzhou, China

ABSTRACT

One of the challenging problems of impartial combinatorial games (ICGs) is to construct generalized winning strategies for possibly infinitely many states. In this paper, we investigate synthesizing generalized winning strategies for ICGs. To this end, we first propose a logical framework to formalize ICGs based on linear integer arithmetic. We then propose an approach to generating the winning strategy for ICGs. Experimental results on several games demonstrate that our approach is effective in most of these games.

KEYWORDS

Impartial combinatorial games; Strategy synthesis

ACM Reference Format:
Kaisheng Wu, Yong Qiao, Kaidong Chen, Fei Rong, Liangda Fang, Zhao-Rong Lai, Qian Dong and Liping Xiong. 2020. Automatic Synthesis of Generalized Winning Strategies of Impartial Combinatorial Games. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

Strategy representation and reasoning has recently received much attention in artificial intelligence, particularly, multi-agent systems and game theory [3, 10, 13]. In the area of game theory, one class of the elementary and fundamental games is impartial combinatorial games (ICGs) where two players alternate moving with perfect information [6].

One of the challenging problems of ICGs is to synthesize the strategy for the player who can force to win. Given a state, the winning strategy can be computed via off-the-shelf backward algorithms [13]. However, this strategy holds for only one state but not (infinitely) many states. Synthesizing a generalized winning strategy that works for possibly infinitely many states is of interest. Unfortunately, this problem is notoriously difficult even for simple games, and undecidable in general [7].

In this paper, we concentrate on ICGs, and investigate synthesizing generalized winning strategies. The contributions of this paper are as follows: (1) We first propose a logical framework to formalize ICGs based on linear integer arithmetic. (2) We then propose an approach to synthesizing the arithmetic formula, called the winning formula, which exactly captures the states in which the player forces to win. (3) Furthermore, we give a method to synthesize generalized winning strategies for impartial combinatorial games. (4) Finally, we evaluate our approach on several games, and experimental results demonstrate the effectiveness and scalability of the proposed approach.

2 IMPARTIAL COMBINATORIAL GAMES

In this section, we briefly introduce impartial combinatorial games and a logical framework to represent it. A game that satisfies the following conditions is called an ICG [6]: (1) There are two players and possible states such that the player can move from one state to another one. (2) Two players alternate moving and have the same choice of moving. (3) The game ends when it moves to an *ending state* in which no player has a possible move, and always ends in a finite number of moves. There are two play rules: *normal* and *misère*. Under the normal rule, the last player to move wins. By contrast, the last player loses under the misère rule.

By Zermelo's Theorem, there always exists a winning strategy for one player in ICGs [12].

For an ICG, we classify states into two types: *winning* and *losing states*. A winning state is a state winning for the player. By contrast, a losing state is a state where the player cannot force to win. The formal definition is as follows:

Definition 2.1 ([6]). In an ICG, under the normal rule, winning and losing states are recursively defined as follows:

(1) All ending states are losing states.
(2) All states such that there is at least one move to a losing state are winning states.
(3) All states such that the only possible moves are to winning states are losing states.

We then present a logical framework for describing ICGs based on linear integer arithmetic (LIA). Let $\mathcal{N}$ be the set of integers, $\mathcal{V}$ a set of variables and $\mathcal{X} \subseteq \mathcal{V}$ a finite set of state variables. The syntax of linear integer arithmetic is defined as follows. The sets of *arithmetic expressions* (Exp), *literals* (Lit) and *formulas* (Form) is defined by the following grammar:

$$e, e' \in \mathtt{Exp} ::= c \mid v \mid e + e' \mid e - e'$$

$$l \in \mathtt{Lit} ::= e = e' \mid e \neq e' \mid e < e' \mid e > e' \mid e \leq e' \mid e \geq e' \mid e \equiv_{c'} c \mid e \not\equiv_{c'} c$$

$$\phi, \phi' \in \mathtt{Form} ::= l \mid \phi \wedge \phi' \mid \phi \vee \phi' \mid \forall v \phi \mid \exists v \phi$$

*Liangda Fang is the corresponding author. This work was supported by the Natural Science Foundation of China (Nos. 61603152, 61602210, 61703182, 61772232, 61773179, 61877029 and U1811263).

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

where $c, c' \in \mathcal{N}$ and $v \in \mathcal{V}$.

We remark that the literal $e \equiv_c c'$ denotes that e and c are congruent modulo c' (*i.e.*, $e - c$ are divisible by c'), and its negation is $e \not\equiv_c c'$. We use $\mathtt{Form}_{\mathcal{X}}$ for the set of formulas of which free variables are state variables.

Definition 2.2. An ICG is defined as a tuple $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ where

- $\mathcal{X}$: a finite set of state variables.
- $\mathcal{A}$: a finite set of actions.
- C: a formula of $\mathtt{Form}_{\mathcal{X}}$ denoting all legal states.
- $\mathcal{E}$: a formula of $\mathtt{Form}_{\mathcal{X}}$ denoting all ending states.

We use an arithmetic formula to represent all winning states, called *the winning formula*.

Definition 2.3. Let $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ be an ICG and $\phi \in \mathtt{Form}_{\mathcal{X}}$. We call ϕ the winning formula of Π, if

(1) For any ending state s s.t. $s \models \mathcal{E}$, we have $s \models \neg\phi$;
(2) For any legal state s s.t. $s \models \phi$, there is an action a applicable in s s.t. $do(a, s) \models \neg\phi$;
(3) For any legal state s and action a s.t. $s \models \neg\phi$ and a is applicable in s, we have $do(a, s) \models \phi$.

We hereafter define winning strategies as a set of pairs of formulas and actions over state variables.

Definition 2.4. Let $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ be an ICG, and ϕ the winning formula of Π. A winning strategy δ is a set of pairs $\{(\psi_1, a_1), \cdots, (\psi_n, a_n)\}$ where each $\psi_i \in \mathtt{Form}_{\mathcal{X}}$ and a_i is a semi-ground action, if

(1) $\bigvee_{i=1}^{n} \psi_i \equiv \phi$;
(2) $\psi_i \not\equiv \bot$ for $1 \leq i \leq n$;
(3) For all winning states s and $a \in \delta(s)$, $do(a, s)$ is a losing state.

Here, $\delta(s) = \{a_i \mid s \models \psi_i\}$.

3 SYNTHESIS OF WINNING STRATEGIES

In this section, we provide a synthesis approach to winning strategies. Our approach consists of three steps: (1) *synthesizing the winning formula*, (2) *refining the winning formula*, and (3) *synthesizing the winning strategy*. The first and third steps are based on the enumerative algorithm proposed in [9]. The enumerative algorithm aims to synthesize objects satisfying a set of specifications that is represented in LIA. For details, please refer to [9].

Synthesizing the winning formula. We give the constrains for the winning formula based on the definition of winning formulas.

Definition 3.1. Let $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ be an ICG. The constraints for the winning formula ϕ of Π are as follows:

(1) $\mathcal{E} \rightarrow \neg\phi$;
(2) $(C \land \phi) \rightarrow \exists \mathcal{X}'[\mathcal{T}(\mathcal{A}) \land \neg\phi[\mathcal{X}/\mathcal{X}']]$;
(3) $(C \land \neg\phi) \rightarrow \forall \mathcal{X}'[\mathcal{T}(\mathcal{A}) \rightarrow \phi[\mathcal{X}/\mathcal{X}']]$.

where $\phi[\mathcal{X}/\mathcal{X}']$ is the formula obtained by replacing every occurrence of $v \in \mathcal{X}$ in ϕ with v', and $\mathcal{T}(\mathcal{A})$ is the transition formula for all actions a in $\mathcal{A}$ that reflects the relation between states before and after performing a. The three constraints correspond to Items 1 - 3 of Definition 2.3 respectively.

The following is the completeness and soundness of the enumerative algorithm that synthesizes the winning formula according to the above constraints.

Theorem 3.2. *Let $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ be an ICG.*

Soundness *If the enumerative algorithm synthesizes the formula ϕ satisfying the constraints illustrated in Definition 3.1, then ϕ is the winning formula of Π.*

Relatively Completeness *If the winning formula of Π is LIA-definable (i.e., it can be represented in by a LIA formula), then the enumerative algorithm terminates with a winning formula of the smallest size.*

Refining the winning formula. We hereafter present a syntactic method to refine the cover of the winning formula ϕ facilitating synthesis of winning strategies. The method involves two syntactic operations. We first obtain an equivalent formula ϕ' by replacing in ϕ every occurrence of numeric literals of the form $e_1 \neq e_2$, $e_1 \leq e_2$, $e_1 \geq e_2$ and $e_1 \not\equiv_{e_3} e_2$ by the disjunction of arithmetic literals. Then, we obtain a formula of the form $\bigvee_{i=1}^{n} \psi_i$ where ψ_i is a numeric term by transforming it via the distributive law and removing contradictory arithmetic terms.

Synthesizing the winning strategy. Similarly to Step (1), we also synthesize the winning strategy from the refinement of winning formula $\{\psi_1, \cdots, \psi_n\}$ via the enumerative algorithm based on the following two constraints:

(1) $C \land \psi_i \rightarrow \mathtt{pre}(a[\mathcal{Y}/\Sigma])$;
(2) $C \land \psi_i \rightarrow \forall \mathcal{X}'[\mathcal{T}(a[\mathcal{Y}/\Sigma]) \rightarrow \phi[\mathcal{X}/\mathcal{X}']]$.

Intuitively, the first condition requires that $a[\mathcal{Y}/\Sigma]$ is applicable over every legal state s satisfying ψ_i while the second one ensures that performing $a[\mathcal{Y}/\Sigma]$ from any legal state s satisfying ψ_i leads to a losing state.

Finally, we end with the soundness and completeness theorem for synthesizing the winning strategy.

Theorem 3.3. *Let $\Pi = \langle \mathcal{X}, \mathcal{A}, C, \mathcal{E} \rangle$ be an ICG. Let ϕ be the winning formula of Π and Ψ the cover of ϕ.*

Soundness *If the enumerative algorithm synthesizes the winning strategy δ : $\{(\psi_1, a_1[\mathcal{Y}_1/\Sigma_1]), \cdots, (\psi_n, a_n[\mathcal{Y}_n/\Sigma_n])\}$ satisfying the above two constraints for each pair $(\psi_i, a_i[\mathcal{Y}_i/\Sigma_i])$, then δ is the winning strategy of Π.*

Bounded Completeness *Suppose that for each $\psi_i \in \Psi$, there is a semi-ground action $a_i[\mathcal{Y}_i/\Sigma_i]$ s.t. each expression of Σ_i is of at most size m and $\{(\psi_1, a_1[\mathcal{Y}_1/\Sigma_1]), \cdots, (\psi_n, a_n[\mathcal{Y}_n/\Sigma_n])\}$ is the winning strategy of Π. Then, the enumerative algorithm terminates with a winning strategy δ.*

4 EXPERIMENTAL EVALUATION

We have implemented our approach, proposed in the previous section, to a system by using Python and Z3 [4]. We evaluate our system on the following games: 2-rowed and L-shaped Chomp [8], Empty-and-Divide [5], 2-piled Nim [2], the monotonic variation of 2-piled Nim [1], the monotonic 2-diet variation of Wythoff [1], Take-away [6], and Subtraction [11]. Under the normal rule, our approach is able to solve all games except the monotonic 2-diet Wythoff game in a reasonable amount of time (< 250s). This shows the effectiveness and scalability of our approach on a wide range of games under the normal rule. Under the misère rule, our approach only solves the monotonic 2-piled Nim, Take-away and Subtraction games in a fully automated way.

REFERENCES

[1] June Ahn, Benjamin Chen, Richard Chen, Ezra Erives, Jeremy Fleming, Michael Gerovitch, Tejas Gopalakrishna, Tanya Khovanova, Neil Malur, Nastia Polina, and Poonam Sahoo. 2017. On Variations of Nim and Chomp. *arXiv preprint arXiv:1705.06774* (2017).

[2] Charles L. Bouton. 1901. Nim, a game with a complete mathematical theory. *Annals of Mathematics* 3, 1/4 (1901), 35–39.

[3] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. 2010. Strategy logic. *Information and Computation* 208 (2010), 677–693.

[4] Leonardo de Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Proceedings of the Fourteenth International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS-2008)*. 337–340.

[5] Thomas S. Ferguson. 1998. Some chip transfer games. *Theoretical Computer Science* 191, 1-2 (1998), 157–171.

[6] Thomas S. Ferguson. 2018. *Game Theory*. World Scientific.

[7] Kailun Luo and Yongmei Liu. 2019. Automatic Verification of FSA Strategies via Counterexample-Guided Local Search for Invariants. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-2019)*. 1814–1821.

[8] Frederik Schuh. 1952. The game of divisions.

[9] Abhishek Udupa, Arun Raghavan, Jyotirmoy V. Deshmukh, Sela Mador-Haim, Milo M. K. Martin, and Rajeev Alur. 2013. TRANSIT: Specifying Protocols with Concolic Snippets. In *Proceedings of the Thirty-Fourth ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI-2013)*. 287–296.

[10] Liping Xiong and Yongmei Liu. 2016. Strategy Representation and Reasoning for Incomplete Information Concurrent Games in the Situation Calculus. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-2016)*. 1322–1329.

[11] IM Yaglom. 2001. Two games with matchsticks. *Kvant Selecta: Combinatorics, I* 1 (2001), 1–8.

[12] Ernst Zermelo. 1913. Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. In *Proceedings of the Fifth International Congress of Mathematicians*, Vol. 2. Cambridge University Press, 501–504.

[13] Dongmo Zhang and Michael Thielscher. 2015. Representing and Reasoning about Game Strategies. *Journal of Philosophical Logic* 44 (2015), 203–236. Issue 2.