Additional Winning Strategies in Two-Player Games*

Vadim Malvone and Aniello Murano

Università degli Studi di Napoli Federico II

Abstract. We study the problem of checking whether a two-player reachability game admits more than a winning strategy. We investigate this in case of perfect and imperfect information, and, by means of an automata approach we provide a linear-time procedure and an exponential-time procedure, respectively. In both cases, the results are tight.

1 Introduction

Game theory is a powerful mathematical framework largely exploited in computer science and AI [1,9,16]. In the basic setting, we refer to two-player turn-based games where the players, conventionally named $Player_0$ and $Player_1$, play for a finite number of rounds. The states of the arena are partitioned among the players and each player can move only from the states he owns. Solving such a game amounts to check whether $Player_0$ has a *winning strategy*, i.e., a sequence of moves that allows him to achieve his goal, no matter how his opponent plays.

In several game settings having a more precise (quantitative) information about *how many* winning strategies a player has at his disposal turns out to be crucial. For example, in Nash Equilibrium, such an information amounts to solve the challenging question of checking whether the equilibrium is unique [2].

In this paper, we address the quantitative question of checking whether $Player_0$ has more than a strategy to win a finite two-player turn-based game, under the reachability condition. We consider both the cases that the players have perfect or imperfect information about the moves performed by the opponent. For the solution we use an automata-theoretic approach. Precisely, we build an automaton that accepts all tree witnesses for more than one winning strategy for $Player_0$. Hence, we reduce the addressed quantitative question to the emptiness of this automaton. In the perfect information setting, this results in a linear-time upper bound complexity. In the imperfect information setting, instead, it results in an exponential-time solution. In both cases, the results are tight.

Related works. Counting strategies has been deeply exploited in *reactive* systems formal verification by means of specification logics extended with *graded modalities*, interpreted over games of infinite duration [2,6,10]. These works suitably extend preliminary reasonings in closed system verification [3,4,7].

* This work is a communication based on the works [12] and [11].

2 The Game Model

In this section, we present our model and its semantics.

Definition 1. *A* turn-based two-player reachability game with imperfect information *(2TRGI), $Player_0$ vs $Player_1$, is a tuple* $G = <\mathrm{St}, s_I, \mathrm{Ac}, \mathrm{tr}, \mathrm{W}, \cong>$, *where* $\mathrm{St} = \mathrm{St}_0 \cup \mathrm{St}_1$ *is a finite non-empty set of* states, *with* St_i *set of states of* $Player_i$, $s_I \in \mathrm{St}$ *is an* initial state, $\mathrm{Ac} = \mathrm{Ac}_0 \cup \mathrm{Ac}_1$ *is the set of actions,* W *is a set of target states,* $\mathrm{tr} : \mathrm{St}_i \times \mathrm{Ac}_i \to \mathrm{St}_{1-i}$, *for* $i \in \{0, 1\}$ *is a* transition function *mapping a state of a player and its action to a state belonging to the other player, and* $\cong = \cong_0 \cup \cong_1$ *is a set of equivalence relations on* Ac.

Let $i \in \{0, 1\}$ and $a, a' \in \mathrm{Ac}_i$ be two actions. If $a \cong_{1-i} a'$ we say that a and a' are *indistinguishable* to $Player_{1-i}$. By $[\mathrm{Ac}_i] \subseteq \mathrm{Ac}_i$ we denote the subset of actions that are *distinguishable/visible* for $Player_{1-i}$. In particular, for each set of equivalent actions over Ac_i, we put a representative action in $[\mathrm{Ac}_i]$. From any $s \in \mathrm{St}_i$, if $a \cong_{1-i} a'$ then also the reached states are indistinguishable, ie $\mathrm{tr}(s, a) = s'$ and $\mathrm{tr}(s, a') = s''$ are indistinguishable for $Player_{1-i}$. A relation $\cong_i$ is an *identity* if $a \cong_i a'$ iff $a = a'$. A *2TRGI* has perfect information (a *2TRG*, for short) if $\cong$ contains only identities. To give the semantics of *2TRGI*s, we introduce the concepts of track, strategy and play.

A *track* is a finite sequence of states $\rho \in \mathrm{St}^*$ such that, for all $i < |\rho|$, if $(\rho)_i \in \mathrm{St}_0$ then there exists an action $a_0 \in \mathrm{Ac}_0$ such that $(\rho)_{i+1} = \mathrm{tr}((\rho)_i, a_0)$, else there exists an action $a_1 \in \mathrm{Ac}_1$ such that $(\rho)_{i+1} = \mathrm{tr}((\rho)_i, a_1)$, where $(\rho)_i$ denotes the i-st element of ρ. By $last(\rho)$ we denote the last element of ρ and by $\rho_{\leq i}$ the prefix track $(\rho)_0 \dots (\rho)_i$. By $\mathrm{Trk} \subseteq \mathrm{St}^*$ we denote the set of tracks over St and by Trk_i the set of tracks ρ in which $last(\rho) \in \mathrm{St}_i$. For simplicity, we assume that all tracks in Trk start at the initial state $s_I \in \mathrm{St}$.

A *strategy* for $Player_i$ is a function $\sigma_i : \mathrm{Trk}_i \to \mathrm{Ac}_i$ that maps a track to an action. A strategy is *uniform* if it adheres on the visibility (visible actions) of the players. In the rest of the paper we only refer to uniform strategies.

The composition of strategies, one for each player, induces a computation called *play*. Precisely, assume $Player_0$ and $Player_1$ take strategies σ_0 and σ_1, respectively. Their composition *induces* a play ρ such that $(\rho)_0 = s_I$ and for each $i \geq 0$ if $(\rho)_i \in \mathrm{St}_0$ then $(\rho)_{i+1} = \mathrm{tr}((\rho)_i, \sigma_0(\rho_{\leq i}))$, else $(\rho)_{i+1} = \mathrm{tr}((\rho)_i, \sigma_1(\rho_{\leq i}))$.

We can now give the definition of *reachability winning condition*.

Definition 2. *$Player_0$ wins a 2TRGI, under the reachability condition, if he has a strategy that for all strategies of $Player_1$ the resulting induced play will enter a state in* W. *Such a strategy is said* winning *for $Player_0$.*

Given a *2TRGI* and one of its play ρ, it is possible to check who is the winner in ρ by considering only $(\rho)_0, \dots, (\rho)_{|\mathrm{St}|+1}$. In fact, if $(\rho)_i \in \mathrm{W}$ for some $i \in \{0, \dots, |\mathrm{St}| + 1\}$ then $Player_0$ wins the play ρ, otherwise there exists a loop in which $Player_1$ can stay infinitely, and then he wins the game.

We formalize the winning condition by means of a tree structure that we call *schema strategy tree*. To proper introduce it, we provide the concept of *decision tree*. We start with some basic notion about trees.

Let Υ be a set. An Υ-*tree* is a prefix closed subset $T \subseteq \Upsilon^*$. The elements of T are called *nodes* and the empty word ε is the *root* of T. Given a node $v = y \cdot x$, with $y \in \Upsilon^*$ and $x \in \Upsilon$, we define $prf(v)$ to be y and $last(v)$ to be x. For an alphabet Σ, a Σ-labeled Υ-tree is a pair $< T, V >$ where T is an $\Upsilon-$tree and $V : T \to \Sigma$ maps each node of T to a symbol in Σ.

A *decision tree* is the unwinding of the game structure along with all possible combinations of player actions, ie a tree that collects all tracks over the game. A winning strategy can be seen as an opportune mapping, over the decision tree, of a player's "strategy schema" built over the visibility. In other words, the player first makes a decision over a set S of indistinguishable states and then this choice is used in the decision tree for each state in S. This makes the decision tree *uniform*. However, observe that we use synchronous memoryfull strategies. This means that in a decision tree, the set S of indistinguishable states resides at the same level. To make this idea more precise, we now formalize the concept of *schema strategy tree* and *uniform strategy tree*.

Definition 3. *Given a 2TRGI and a uniform strategy σ for $Player_i$, a* schema strategy tree *for $Player_i$ is a $\{\top, \bot\}$-labeled $(\mathrm{Ac}_i \cup [\mathrm{Ac}_{1-i}])$-tree $< T, V >$, with $T \subset (\mathrm{Ac}_i \cup [\mathrm{Ac}_{1-i}])^*$ and V as follows: (i) $V(\varepsilon) = \top$; (ii) for all $v \in T$, if $last(v) \in [\mathrm{Ac}_{1-i}]$ then $V(v) = \top$, else let $\rho = (\rho)_0 \dots (\rho)_{|v|-1}$ be a track from s_I, with $(\rho)_k = \mathrm{tr}((\rho)_{k-1}, last(v_{\leq k}))$ for each $0 \leq k \leq |v| - 1$, if $last(v) = \sigma(\rho)$ then $V(v) = \top$, else $V(v) = \bot$.*

In a schema strategy tree the $\top$ label indicates that $Player_i$ selects the corresponding set of visible states in the decision tree and $\bot$ is used conversely. In particular, the starting node of the game is the root of the schema strategy tree and it is always enabled (condition (i)); all nodes belonging to the adversarial player are always enabled; and one of the successors of $Player_i$ nodes is enabled in accordance with the uniform strategy σ (condition (ii)). Simply, a *uniform strategy tree* is a projection of the decision tree along the schema strategy tree.

3 Additional Winning Strategies for 2TRGI

In this section we provide the main result of this work. Technically, we make use of *alternating tree automata* [14].

An alternating tree automaton (*ATA*) is a tuple $A =< \Sigma, D, Q, q_0, \delta, F >$, where Σ is the alphabet, D a finite set of directions, Q the set of states, $q_0 \in Q$ the initial state, $\delta : Q \times \Sigma \to \mathcal{B}^+(D \times Q)$ the transition function, where $\mathcal{B}^+(D \times Q)$ is the set of all positive Boolean combinations of pairs (d, q) with d direction and q state, and $F \subseteq Q$ the set of the accepting states. An *ATA* recognizes (finite) trees by means of runs. For a Σ-labeled tree $< T, V >$, with $T = D^*$, a run is a $(D^* \times Q)$-labeled $\mathbb{N}$-tree $< T_r, r >$ such that the root is labeled with (ε, q_0) and the labels of each node and its successors satisfy the transition relation. A run is *accepting* if all its leaves are labeled with accepting states. An input tree is accepted if there exists a corresponding accepting run. By $L(A)$ we denote the set of trees accepted by A. We say that A is not empty if $L(A) \neq \emptyset$.

For our purpose, we formalize the concept of *schema additional strategy tree*.

Definition 4. *Given a 2TRGI and two uniform strategies σ and σ' for $Player_i$, a* schema additional strategy tree *for $Player_i$ is a $\{\top,\bot\}$-labeled $(\mathrm{Ac}_i \cup [\mathrm{Ac}_{1-i}])$-tree $< T, V >$, with $T \subset (\mathrm{Ac}_i \cup [\mathrm{Ac}_{1-i}])^*$ and V as follows: (i) $V(\varepsilon) = \top$; (ii) for all $v \in T$, if $last(v) \in [\mathrm{Ac}_{1-i}]$ then $V(v) = \top$, else let $\rho = (\rho)_0 \dots (\rho)_{|v|-1}$ be a track from s_I, with $(\rho)_k = \mathrm{tr}((\rho)_{k-1}, last(v_{\leq k}))$ for each $0 \leq k \leq |v| - 1$, if $last(v) = \sigma(\rho)$ or $last(v) = \sigma'(\rho)$ then $V(v) = \top$, else $V(v) = \bot$.*

Now, we have all ingredients to give the following result.

Theorem 1. *Given a 2TRGI the problem of deciding whether $Player_0$ has more than a uniform winning strategy is ExpTime-Complete.*

Proof. For the lower bound, we recall that 2-player turn-based games with imperfect information is *ExpTime*H [13]. For the upper bound, we use an automata approach. Precisely, we build an *ATA* A that accepts all schema additional strategy trees for $Player_0$. It has as set of states $Q = \mathrm{St} \times \mathrm{St} \times \{\top,\bot\} \times \{0,1\} \times \{ok, split\}$ and alphabet $\Sigma = \{\top,\bot\}$. We use in Q a duplication of states as we want to remember the state associated to the parent node while traversing the tree. The flag *split* is used to remember we have to "enter" two winning strategy paths, so moving to *ok*. The flag $f \in \{1,0\}$ indicates whether along a path we have entered or not a target state. As initial state we set $q_0 = (s_I, s_I, \top, 0, split)$. Given a state $q = (s, s', t, f, \bar{f})$, the transition relation $\delta(q, t')$ is defined as:

$$\begin{cases} \bigwedge_{a_0 \in \mathrm{Ac}_0} (d, (s', s'', \top, f', ok)) & \text{if } s' \in \mathrm{St}_0 \wedge t' = \top \wedge t = \top \wedge \bar{f} = ok \\ \bigwedge_{a_0 \in \mathrm{Ac}_0} \bigvee_{\bar{f}' \in \{ok, split\}} (d, (s', s'', \top, f', \bar{f}')) & \text{if } s' \in \mathrm{St}_0 \wedge t' = \top \wedge t = \top \wedge \bar{f} = split \\ \bigwedge_{a_1 \in \mathrm{Ac}_1} (d, (s', s'', \top, f', \bar{f})) & \text{if } s' \in \mathrm{St}_1 \wedge t' = \top \wedge t = \top \\ false & \text{if } t' = \top \text{ and } t = \bot \\ true & \text{if } t' = \bot \end{cases}$$

if $s' \in \mathrm{St}_0$ then $s'' = \mathrm{tr}(s', a_0)$ and d is in accordance with $|\mathrm{Ac}_1|$, else $s'' = \mathrm{tr}(s', a_1)$ and d is in accordance with $|\mathrm{Ac}_0|$; if $q' \in \mathrm{W}$ then $f' = 1$, otherwise $f' = f$.

The set of accepting states is $\mathrm{F} = \{(s, s', t, f, \bar{f}) : s, s' \in \mathrm{St} \wedge t = \top \wedge f = 1 \wedge \bar{f} = ok\}$. Recall that an input tree is accepted if there exists a run whose leaves are all labeled with accepting states. In our setting this means that an input tree simulates a schema additional strategy tree for $Player_0$. So, if the automaton is not empty then $Player_0$ wins the game with more than one strategy, ie, there exists a schema additional strategy tree for him.

The desired computational complexity follows by considering that: *(i)* the size of the automaton is polynomial in the size of the game, *(ii)* to check its emptiness can be performed in exponential time [5,8].

In case we study a *2TRG* the automaton provided in the above proof sends one copy in each direction. So, the automaton is nondeterministic. By recalling that the emptiness problem in this case is solvable in linear-time [15] and the *PTime*-hardness for alternating reachability games [9] the following result holds.

Theorem 2. *Given a 2TRG the problem of deciding whether $Player_0$ has more than a winning strategy is PTime-Complete.*

4 Conclusion and Future Work

In this paper we have shown an automata-theoretic approach to check whether a player has more than a winning strategy in a two-player reachability game. Our approach works with optimal asymptotic complexity both in the case the players have perfect information about the moves performed by their adversarial or not. We believe that our results can be used as core engine to count strategies in more involved game scenarios and in many solution concepts reasoning. For example, it can be used to solve the *Unique Nash Equilibrium* problem, in an extensive game form of finite duration. As future work, it would be useful to check additional winning strategies in multi-agent concurrent games.

References

1. R. Alur, T. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *Journal of the ACM*, 49(5):672–713, 2002.
2. B. Aminof, V. Malvone, A. Murano, and S. Rubin. Graded strategy logic: Reasoning about uniqueness of nash equilibria. In *AAMAS 2016*, pages 698–706, 2016.
3. B. Aminof, A. Murano, and S. Rubin. On ctl* with graded path modalities. In *LPAR-20*, pages 281–296, 2015.
4. P. Bonatti, C. Lutz, A. Murano, and M. Vardi. The Complexity of Enriched muCalculi. *Logical Methods in Computer Science*, 4(3):1–27, 2008.
5. E. Emerson and C. Jutla. The Complexity of Tree Automata and Logics of Programs (Extended Abstract). In *FOCS'88*, pages 328–337. IEEE Computer Society, 1988.
6. A. Ferrante, A. Murano, and M. Parente. Enriched Mu-Calculi Module Checking. *Logical Methods in Computer Science*, 4(3):1–21, 2008.
7. O. Kupferman, U. Sattler, and M. Vardi. The Complexity of the Graded muCalculus. In *CADE'02*, LNCS 2392, pages 423–437. Springer, 2002.
8. O. Kupferman, M. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM*, 47(2):312–360, 2000.
9. O. Kupferman, M. Vardi, and P. Wolper. Module Checking. *Information and Computation*, 164(2):322–344, 2001.
10. V. Malvone, F. Mogavero, A. Murano, and L. Sorrentino. On the counting of strategies. In *TIME 2015*, pages 170–179, 2015.
11. V. Malvone, A. Murano, and L.Sorrentino. Additional Winning Strategies in Finite Games. Under submission.
12. V. Malvone, A. Murano, and L. Sorrentino. Games with additional winning strategies. In *CILC'15*, CEUR Workshop Proceedings, vol. 1459, pages 175–180. CEUR-WS.org, 2015.
13. J. H. Reif. The complexity of two-player games of incomplete information. *J. Comput. Syst. Sci.*, 29(2):274–301, 1984.
14. G. Slutzki. Alternating tree automata. *Theoretical Computer Science*, 41:305–318, 1985.
15. W. Thomas. Infinite trees and automaton definable relations over omega-words. In *STACS'90*, pages 263–277, 1990.
16. M. Wooldridge. *An Introduction to Multi Agent Systems.* John Wiley & Sons, 2002.