Backward Induction is PTIME-complete

Jakub Szymanik*

Institute of Logic Language and Computation, University of Amsterdam
J.K.Szymanik@uva.nl

Abstract. We prove that the computational problem of finding backward induction outcome is PTIME-complete.

Key words: game-theory, backward induction, computational complexity, finite extensive games

1   Introduction

Higher-order reasoning of the form 'I believe that Ann knows that Peter thinks . . . ' is an attractive topic for logical analysis. The logical investigations often go hand in hand with game theory. In this context, one of the common topics among researchers in logic and game theory has been backward induction (henceforth, BI), the process of reasoning backwards, from the end to determine a sequence of optimal actions. BI is a common method for determining sub-game perfect equilibria in the case of finite extensive-form games. BI can be understood as an inductive algorithm defined on a game tree – an algorithm that tells us which sequence of actions will be chosen by agents that want to maximize their own payoffs, assuming common knowledge of rationality.

Games have been also extensively used to design experimental paradigms aiming at studying social cognition, with a particular focus on higher-order social cognition. Often the experimental turn-based games can be modeled as extensive-form games and solved by applying BI. As it is hard to determine what the reasoning strategies used by participants in such games are, formal findings on backward induction have been used to better understand humans' strategic reasonings [1].

Recently, Van Benthem and Gheerbrant [2] have studied the logical definability of BI. They have observed that it can be defined in the first-order logic extended with the least fixed-point operator as well as in a variety of other dynamic epistemic formalisms. Obviously, from the least fixed-point definability result it follows that BI is in PTIME [3]. But is it also hard, and therefore complete for PTIME?

2   Preliminaries

Let us start by recalling that the reachability problem on alternating graphs is PTIME-complete [3].

Definition 1. *Let an alternating graph $G = (V, E, A, s, t)$ be a directed graph whose vertices, V, are labeled universal or existential. $A \subseteq V$ is the set of universal vertices. $E \subseteq V \times V$ is the edge relation.*

Definition 2. *Let $G = (V, E, A, s, t)$ be an alternating graph. We say that t is reachable from s iff $P_a^G(s, t)$, where $P_a^G(x, y)$ is the smallest relation on vertices of G satisfying:*

(1) $P_a^G(x, x)$
(2) If x is existential and $P_a^G(z, y)$ holds for some edge (x, z) then $P_a^G(x, y)$.
(3) If x is universal, there is at least one edge leaving x, and $P_a^G(z, y)$ holds for all edges (x, z) then $P_a^G(x, y)$.

The idea here is that for t to be reachable from an existential node x there must exist a path from x to t, while the condition for a universal node y is stronger: t is reachable from y if and only if every path from y leads to t. One can think about alternating reachability in terms of a competitive game, where the player controlling existential vertices wants to get to t and the player controlling universal vertices is trying to prevent that. For example, in the alternating graph of Fig. 1, t is not reachable from s (i.e., there is no winning strategy for the existential player). To see it just imagine that the first player will move from s to v. Then the second player has only one choice leading to the dead-end. It means, that not every move of the first player controlling the universal node s is on the path to t.
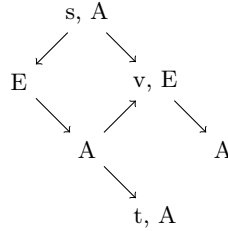

Fig. 1. t is not reachable from s.

Now, we can define the alternating reachability problem, that is a class of alternating graphs in which t is reachable from s. One can think about that as a decision problem: given an alternating graph G and nodes s, t check whether t is reachable from s.

Definition 3. $REACH_a = \{G|P_a^G(s,t)\}$

The following computational complexity result will be crucial for us.

Theorem 1 ([4]). *$REACH_a$ is PTIME-complete via first-order reductions.*

Proof. The original proof of Immerman simulates directly an alternating Turing machine (ATM) to show that the problem is complete for ATM logarithmic space, known to be equal to P [5].

As the proof of Theorem 1 simulates ATM computation tree it follows that:

Corollary 1 *$REACH_a$ is PTIME-complete on trees.*

Note, that given a game tree T and an existential node s, the problem $REACH_a$ over T intuitively corresponds to the question: 'Is s a winning position for the first player in the zero-sum game T, i.e., can the first player force the game from node s towards node t against all possible counterstrategies of the second player?' (see [6], Problem A.11.1).

3   Backward Induction Problem

Now we are ready to define the computational decision problem corresponding to BI for extensive form, non zero-sum games. Intuitively: can the first player force the game from node s towards node t against all possible *rational* (= pay-off maximizing) counterstrategies of the second player? The difference here is that we consider only rational strategies as the non zero-sum games do not have to be strictly competitive.

Definition 4. *A two-player finite extensive form game $T = (V, E, V_1, V_2, V_{end}, f_1, f_2, s, t)$, where V is the set of nodes, $E \subseteq V \times V$ is the edge relation (available moves). For $i = 1, 2$, $V_i \subseteq V$ is the set of nodes controlled by Player i, and $V_1 \cap V_2 = \emptyset$. V_{end} is the set of end nodes. Finally, $f_i : V_{end} \longrightarrow \mathbb{N}$ assigns pay-offs for Player i.*

Without loss of generality let us restrict attention to *generic games*:

Definition 5. *A game T is generic, if for each player, distinct end nodes have different pay-offs.*

Definition 6. *Let T be a two-player game. We define the backward induction accessibility relation on T. Let $P_{bi}^T(x, y)$ be the smallest relation on vertices of T such that:*

(1) $P_{bi}^T(x, x)$
(2) Take $i = 1, 2$. Assume that $x \in V_i$ and $P_{bi}^T(z, y)$. If the following two conditions hold, then also $P_{bi}^T(x, y)$ holds:
*   (a) $E(x, z)$;*
*   (b) there is no w, v such that $E(x, w)$, $P_{bi}^T(w, v)$, and $f_i(v) > f_i(y)$.*

For example, in the tree of Fig. 3 t is not a backward induction solution for the game starting from s. Player 2 will rather start the game by going to the state w than v. And, t is not reachable from w.

We can again define the corresponding decision problem – whether in the game represented by tree T and starting in node s the first player can force the output t – as a class of game trees where s and t belong to the backward induction accessibility relation on T?
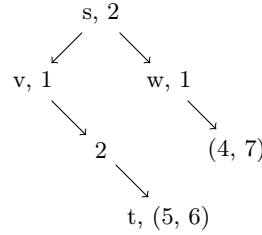
s, 2
v, 1
w, 1
2
(4, 7)
t, (5, 6)

Fig. 2. t is not reachable from s.

Definition 7. $\mathbb{BI} = \{T | P_{bi}^T(s, t)\}$

The problem $\mathbb{BI}$ intuitively corresponds to the question whether t is a sub-game perfect equilibrium in game T starting at node s [7].

4 Complexity of BI

The definability result of Van Benthem and Gheerbrant implies that it can be decided in polynomial time whether node t is a subgame perfect equilibrium of the game, i.e., the result of a gameplay following a BI strategy. First of all, note that it also follows that given an arbitrary finite extensive game with starting node s one can find a BI solution of the game in polynomial time. Simply, it is enough to run the polynomial decision algorithm for every node of the game. In this section we prove that computing backward induction relation is not only in PTIME but it is actually a PTIME-complete problem.

Theorem 2. $\mathbb{BI}$ *is PTIME-complete via first-order reductions.*

Proof. First of all, $\mathbb{BI}$ is in PTIME by providing FO(LFP) definition [3]. Now, it suffices to show PTIME-hardness. For that we will reduce the $REACH_a$ problem on trees (cf. Corollary 1) to the $\mathbb{BI}$ problem. We take any alternating tree $T = (V, E, A, s, t)$. Without loss of generality let us assume that s is existential. We construct a two player game, $T' = (V', E', V_1, V_2, V_{end}, f_1, f_2, s', t')$, where: $V = V'$, $t \in V_{end} = \{$end nodes of V$\}$, $E = E'$, $V_1 = V - A$, $V_2 = A$, $s = s$, $t = t$, and for every $v \in V'$, if $v \neq t$, then $f_1(t) > f_1(v)$ and $f_2(t) < f_2(v)$.

Now, we need to prove that $T \in REACH_a$ iff $T' \in \mathbb{BI}$. Assume, that $T \in REACH_a$. It means that whatever Player 2 does in the game T', Player 1 has a strategy to force outcome t. As t gives strictly the best pay-off for Player 1, then $P_{bi}^{T'}(s,t)$. Hence, $T' \in \mathbb{BI}$. For the other direction, assume for contradiction that $T \notin REACH_a$. This means that there is a node $v \neq t$ such that Player 2 can guarantee the game T' to end in v. From the pay-off construction for T', v is more attractive to Player 2 than t. Therefore, it is not the case that $P_{bi}^{T'}(s,t)$. Hence, $T' \notin \mathbb{BI}$.

What does this tell us about the complexity of backward induction? First of all, problems in PTIME are usually taken to be tractable [8], so relatively easy to solve, also for humans [9]. Furthermore, given assumptions on non-collapse, PTIME-completeness suggests that the problem of deciding whether a given node is a sub-game perfect equilibrium of the game is difficult to effectively parallelize (it lies outside NC[1]) and solve in limited space (it lies outside LOGSPACE).

References

1. Szymanik, J., Meijering, B., Verbrugge, R.: Using intrinsic complexity of turn-taking games to predict participants' reaction times. In: Proceedings of the 35th Annual Conference of the Cognitive Science Society. (2013)
2. van Benthem, J., Gheerbrant, A.: Game solution, epistemic dynamics and fixed-point logics. Fundamenta Informaticae **100**(1-4) (2010) 19–41
3. Immerman, N.: Descriptive Complexity. Texts in Computer Science. Springer, New York, NY (1998)
4. Immerman, N.: Number of quantifiers is better than number of tape cells. Journal of Computer and System Sciences **22**(3) (1981) 384 – 406
5. Chandra, A.K., Kozen, D.C., Stockmeyer, L.J.: Alternation. J. ACM **28**(1) (1981) 114–133
6. Greenlaw, R., Hoover, J.H., Ruzzo, W.L.: Limits to Parallel Computation: P-Completeness Theory. Oxford University Press, USA (1995)
7. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. The MIT Press, Cambridge, MA (1994)
8. Edmonds, J.: Paths, trees, and flowers. Canadian Journal of Mathematics **17** (1965) 449–467
9. Frixione, M.: Tractable competence. Minds and Machines **11**(3) (2001) 379–397

---

[1]A problem is in NC if there exist constants c and k such that it can be solved in time $O(log^c n)$ using $O(n^k)$ parallel processors.