数理解析研究所講究録
1021 巻 1997 年 112-118

Computational Playability of Backward Induction Solutions

Hidetoshi TASHIRO * (田代 秀敏)

Graduate School of Economics, Hitotsubashi University, Kunitachi, Tōkyō, 186, Japan

E-mail Address: gem5104@srv.cc.hit-u.ac.jp

Abstract

This paper considers the computational playability of backward induction solutions, *i.e.* whether or not there is an algorithm to play them. We construct a two-person two-stage game with perfect information, in which both players have countably many feasible actions and their payoff functions are computable. We prove that the backward induction solutions of the game, which are proved to exist, are not computably playable because it is impossible to supply the players with algorithms regarding how they should play the solutions. Moreover, we show that if players' payoff functions are polynomial with rational coefficients, then the backward induction solutions are computably playable.

Keywords: Computational playability; Backward induction solution; Computability; Kleene's T-predicate; Search; Polynomial payoff functions

1 Introduction

A solution of a game is said to be *computationally playable* if the solution strategies are computable in the sense that there is an algorithm, *i.e.* a Turing machine, to compute them.[1] In this paper we consider the computational playability of backward induction solutions of some games.

* The author is grateful to Mamoru Kaneko, Kotaro Suzumura and especially to Takashi Nagashima. Furthermore, he would like to acknowledge Midori Hirokawa, Kaori Hasegawa, Wiebe van der Hoek, Ryo Kashima, John Nash, Itzhak Zilcha, Kin Chung Lo, Luchuan A. Liu and Akihiko Matsui for their comments and discussions. Of course, any possible errors are due to the author.

[1] For the epistemological importance of the concept of computability, see Gödel (1936), Gödel (1946), Kleene (1952, § 62) and Odifreddi (1989, Section I.8). For other epistemological concepts, *e.g.* definability, deducibility, etc., of a game solution, see Kaneko and Nagashima (1996, 1997).

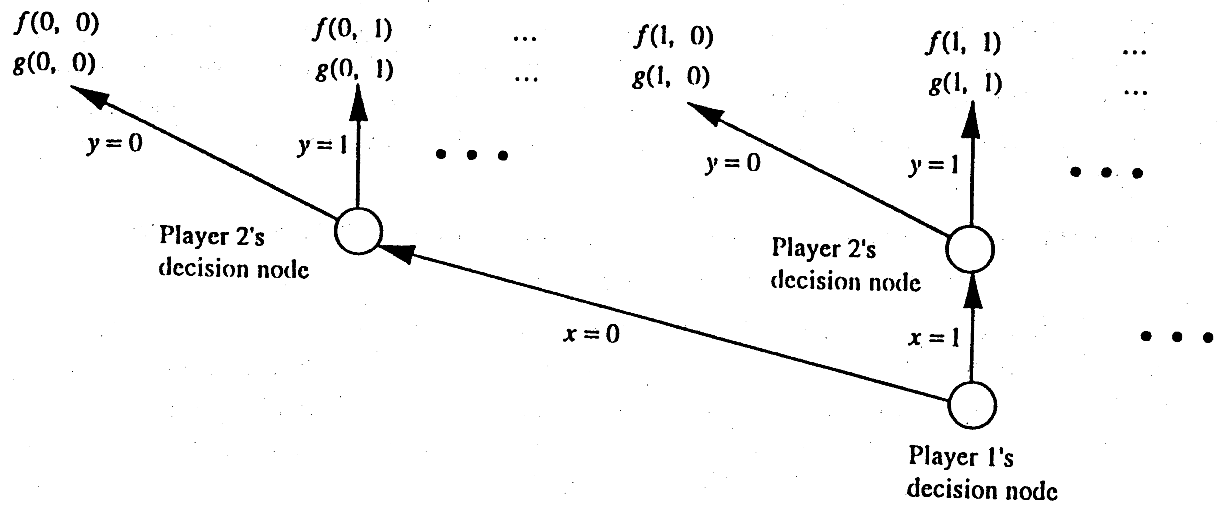

Figure 1: The Game Tree

In the next section we begin by constructing a two-person two-stage game with perfect information, in which both players have countably many feasible actions and their payoffs are natural numbers. First we show that for any payoff functions, there exists a backward induction solution of the game. Then it may be conjectured that if both players' payoff functions are computable, *i.e.* there exists algorithms to compute them, then the backward induction solutions of the game would be computationally playable. However the conjecture is false. Indeed, we prove that while the game has a backward induction solution, no backward induction strategy is computable and further it is not computable whether or not a given action profile is the realization of the backward induction solution. These results mean that the backward induction solutions of the constructed game are not computably playable because it is impossible to supply the players with effective instructions regarding how to play them. Finally, we show that, for any polynomial payoff function of the second moving player and for any backward induction solution, the backward induction strategy is computable. This result implies that if the rules of the game are so simple that players' payoff functions are polynomials, then the backward induction solutions are computably playable.

2 Existence and Playability of Backward Induction Solutions

Both players, 1 and 2, have countably many feasible actions, *i.e.* each player's action space is $\mathbf{N} = \{0, 1, 2, \dots\}$. Players 1 and 2's actions are denoted with x and y, respectively. Furthermore, players 1 and 2's payoff functions are denoted by $f(x, y)$ and $g(x, y)$, respectively. We assume that both players are minimizers.

The rules of the game are, illustrated in Figure 1, as follows: in the first stage player 1 chooses his action x; and, in the second stage player 2 observes x, and then chooses his action y. The pair (x, y) determines players' payoffs $f(x, y)$ and $g(x, y)$.

A pair (x^*, ψ) is said to be a *backward induction solution* of the game iff $\forall x\,[\,f(x^*, \psi(x^*)) \leq f(x, \psi(x))\,]$ and $\forall x \forall y\,[\,g(x, \psi(x)) \leq g(x, y)\,]$. x^* is called the *backward induction action* for player 1 and ψ is called the *backward induction strategy* for player 2.

Theorem 1 *Assume that f and g are any functions from* $\mathbf{N} \times \mathbf{N}$ *to* $\mathbf{N}$. *There exists a backward induction solution of the game defined above.*

Proof Let x be an arbitrarily fixed action. Then the range $\{g(x, y) \mid y \in \mathbf{N}\}$ has a unique minimum element, say, $g(x, y^*)$. Let $\psi(x)$ be the minimum of such y^*. Thus we have a backward induction strategy ψ for player 2, which satisfies $\forall y[g(x, \psi(x)) \leq g(x, y)]$. Now consider the function $f(x, \psi(x))$. Again, the range of this function has a minimum, say, $f(x^*, \psi(x^*))$. This x^* satisfies $\forall x[f(x^*, \psi(x^*)) \leq f(x, \psi(x))]$. Thus, (x^*, ψ) is a backward induction solution of the game. □

The existence of backward induction solutions does not imply their computational playability. Indeed, we can construct player 2's computable payoff function g such that for any backward induction solution (x^*, ψ) of the game, the backward induction strategy ψ is not computable.

To define players 2's specific payoff function, we explain Kleene's T-predicate $T_1(z, x, y)$. Kleene's T-predicate $T_1(z, x, y)$ is a particular computable predicate, as Kleene (1952, p.281) mentions. Intuitively, Kleene's T-predicate $T_1(z, x, y)$ means that z is a code of an algorithm, and that y is the code of a computation on the code x of an input, as Davis (1958, pp.57–58) mentions. In other words, $T_1(z, x, y)$ represents the relation that, given codes z and x of an algorithm and an input, a universal Turing machine, which is an ideal computer, operates the computation whose code is y. The predicate $\exists y T_1(x, x, y)$ is not computable, *i.e.* there exists no algorithm to decide for given x whether or not $\exists y T_1(x, x, y)$ holds.[2]

Theorem 2 *Assume that player 2's payoff function g is as follows:*

$$g(x, y) = \begin{cases} 0 & \textit{if } T_1(x, x, y), \\ 1 & \textit{otherwise.} \end{cases} \tag{1}$$

Then for any backward induction solution (x^*, ψ) of the game, the backward induction strategy ψ is not computable.

Proof Since $T_1(x, x, y)$ is a computable predicate, g is a computable function from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{N}$. Let (x^*, ψ) be any backward induction solution of the game. Then we have

$$g(x, \psi(x)) = \min_y g(x, y) = \begin{cases} 0 & \text{if } \exists y\, T_1(x, x, y), \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

[2] This fact is one of the most fundamental theorems in computability theory. See Kleene (1952, p.301).

Now we prove the following:

$$(3) \qquad \forall x\,[\,\exists y\, T_1(x,x,y) \iff T_1(x,x,\psi(x))\,].$$

Consider any x such that $\exists y\, T_1(x,x,y)$ holds. Then by (2) we have $g(x,\psi(x)) = 0$. Therefore by (1) we have $T_1(x,x,\psi(x))$. Conversely $T_1(x,x,\psi(x))$ implies $\exists y\, T_1(x,x,y)$. Consequently (3) holds.

Since $\exists\, y\, T_1(x,x,y)$ is not computable, by (3) the predicate $T_1(x,x,\psi(x))$ is not computable. This implies that ψ is not computable. □

Theorem 2 means that there are games in which the backward induction solutions of the game, which are proved to exist, are not computably playable because it is impossible to supply the players with algorithms regarding how they should play the solutions.

An action profile $(x^*, \psi(x^*))$ is said to be the *realization* of a backward induction solution (x^*, ψ) of the game. Despite of Theorem 2, it may be conjectured that players can play the realization without computing ψ, because players' payoff functions are computable and the ranges $\{f(x,y) \mid x \in \mathbf{N}\}$ and $\{g(x,y) \mid y \in \mathbf{N}\}$ have unique minimum elements. However, the conjecture is false.

Theorem 3 *Assume that player 2's payoff function g is* (1). *Then for any backward induction solution $(x^*,\ \psi)$ of the game, it is not computable whether or not a given action profile (x,y) is the realization of $(x^*,\ \psi)$.*

Proof Let $(x^*,\ \psi)$ be any backward induction solution of the game. Assume that a given action profile (x,y) is the realization of $(x^*,\ \psi)$. Then we have $\forall w[g(x,y) \le g(x,w)]$. Moreover, suppose that it is computable whether or not (x,y) is the realization of $(x^*,\ \psi)$. We have only to prove the noncomputability of $\forall w[g(x,y) \le g(x,w)]$.

Now we show the following:

$$(4) \qquad \forall w[g(x,y) \le g(x,w)] \iff \left[\exists w T_1(x,x,w) \Longrightarrow T_1(x,x,y)\right].$$

Since g takes only the values 0 and 1, $g(x,y) \le g(x,w)$ is equivalent to $g(x,y) = 0$ or $g(x,w) = 1$. Thus $\forall w[g(x,y) \le g(x,w)]$ is equivalent to $\exists w[g(x,w) = 0] \Longrightarrow g(x,y) = 0$. By (1), $g(x,y) = 0$ if and only if $T_1(x,x,y)$. Thus $\forall w[g(x,y) \le g(x,w)]$ is equivalent to $\exists w T_1(x,x,w) \Longrightarrow T_1(x,x,y)$.

We have only to show the noncomputability of the predicate $\exists w T_1(x,x,w) \Longrightarrow T_1(x,x,y)$, which we abbreviate to $R(x,y)$. Suppose that $R(x,y)$ is computable. Then, by (4) and by the existence of the minimum element of the range $\{g(x,y) \mid y \in \mathbf{N}\}$, given x, players can search the propositions $R(x,0), R(x,1), R(x,2), \ldots$ in succession to look for one that is true. Consider the first y for which $R(x,y)$ is true. Then, since $T_1(x,x,y)$ implies $\exists w T_1(x,x,w)$, we have $\exists w T_1(x,x,w) \iff T_1(x,x,y)$. Hence, since $T_1(x,x,y)$ is computable, $\exists w T_1(x,x,w)$ is also computable. This contradicts its noncomputability. □

Despite of Theorems 2 and 3, if payoff functions belong to a certain class, then the backward induction solutions are computably playable.

Theorem 4 *Assume that player 2's payoff function g is any polynomial with integral coefficients in x and y. Then for any backward induction solution $(x^*,\ \psi)$ of the game, the backward induction strategy ψ is computable.*

Proof By assumption, player 2's payoff function g is represented as follows:

$$\begin{aligned} g(x,y) &= \sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij}x^i y^j \\ &= \left(\sum_{i=0}^{n} a_{in}x^i\right)y^n + \left(\sum_{i=0}^{n} a_{i,n-1}x^i\right)y^{n-1} + \ldots + \left(\sum_{i=0}^{n} a_{i1}x^i\right)y + \sum_{i=0}^{n} a_{i0}x^i, \end{aligned}$$

where n is a natural number and a_{ij} is an integer for $i, j = 0, \ldots, n$.[3] We can choose a constant M such that

$$\forall x \forall y \left[y > M\sum_{j=0}^{n}\left|\sum_{i=0}^{n} a_{ij}x^i\right| \Longrightarrow g(x,y) > \min_y g(x,y). \right]$$

Then for given x, in the finite interval $0 \le y < M\sum_{j=0}^{n}\left|\sum_{i=0}^{n} a_{ij}x^i\right|$, we can find a y^* such that $g(x,y^*) = \min_y g(x,y)$. Therefore, ψ is computable. □

As a corollary of Theorem 4, we can prove that if player 2's payoff function g is any polynomial with rational coefficients in x and y, then for any backward induction solution $(x^*,\ \psi)$ of the game the backward induction strategy ψ is computable.

3 Concluding remarks

Assume that players 1 and 2 play repeatedly the game defined in Theorem 2, and that player 1 knows that player 2 uses the same computable strategy but player 1 may not know *which* computable strategy player 2 is using all the time. By Theorem 2, since player 2 uses a computable strategy this strategy is not the backward induction strategy. Then it is unexplained whether or not player 1 can effectively discover after a finite number of plays a way of proceeding the backward induction to optimize against player 2's computable strategy. Rabin (1957) discussed the similiar problem in the game described in the following paragraph. Although he proved a positive result, his discussion cannot be applied to the game defined in Theorem 2 because his discussion was dependent on the win-lose property.

The motive of this paper is to simplify Rabin (1957)'s work on the computational playability of winning strategies. He considered a game such that two players, 1 and 2,

[3] g is a function from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{Z}$.

choose their actions alternately in three-stages. The rules of the game are such that in the first stage, player 1 chooses his action $x \in \mathbf{N}$; in the second stage, player 2 observes x, and then chooses his action $y \in \mathbf{N}$; and, in the third stage, player 1 observes x and y, and then chooses his action $z \in \mathbf{N}$. Then players' payoffs $f(x,y,z) \in \mathbf{N}$ and $g(x,y,z) \in \mathbf{N}$ are determined. Players 1 and 2's payoff functions f and g are defined as follows:

$$f(x,y,z) = \begin{cases} 0 & \text{if } h(z) = x+y, \\ 1 & \text{otherwise,} \end{cases} \quad \text{and} \quad g(x,y,z) \ = \ 1 - f(x,y,z),$$

where h is a computable function, and therefore so are both f and g. Since both f and g take only the values 0 and 1 and since $f(x,y,z) + g(x,y,z) = 1$ for all x, y and z, the game is a win-lose one. Rabin assumed that the range $h(\mathbf{N})$ is 'simple' in the sense that it is a recursively enumerable set, *i.e.* the range of a computable function, and its complement is infinite and contains no infinite recursively enumerable subsets. (For simple sets, see Davis (1958, p.76).) A winning strategy for player 2 of the above described game is any function $\tau(x)$ such that $\forall x \forall z\, [\, x + \tau(x) \neq h(z)\,]$. Rabin proved the noncomputability of winning strategies of the game. His result means that there are games in which the winning strategies exist but none of them is computably playable.

References

[1] M. Davis (1958) *Computability and Unsolvability*, McGraw-Hill, New York.

[2] K. Gödel (1936) "Über die Länge von Beweisen (On the length of proofs)," *Ergebnisse eines mathematischen Kolloquiums, Heft 7*, pp.23–24; reprinted with English translation in: S. Feferman, J.W. Dawson, Jr., S.C. Kleene, G.H. Moore, R.M. Solovay, and J. van Heijenoort, eds., (1986) *Kurt Gödel Collected Works, Volume I*, Oxford University Press, New York, pp.396–398.

[3] K. Gödel (1946) "Remarks before the Princeton bicentennial conference on problems in mathematics," (unpublished), reprinted in: S. Feferman, J.W. Dawson, Jr., S.C. Kleene, G.H. Moore, R.M. Solovay, and J. van Heijenoort, eds., (1990) *Kurt Gödel Collected Works, Volume II*, Oxford University Press, New York, pp.150–153.

[4] M. Kaneko and T. Nagashima (1996) "Game Logic and its Applications I," *Studia Logica 57*, pp.325–354.

[5] M. Kaneko and T. Nagashima (1997) "Game Logic and its Applications II," *Studia Logica 58*, pp.273–303.

[6] S.C. Kleene (1952) *Introduction to Metamathematics*, North-Holland, Amsterdam.

[7] P. Odifreddi (1989) *Classical Recursion Theory*, North-Holland, Amsterdam.

[8] M.O. Rabin (1957) "Effective Computability of Winning Strategies," in: M. Dresher, A.W. Tucker, and P. Wolfe, eds., *Contributions to the Theory of Games, Volume III*, Princeton University Press, Princeton, pp.147–157.