数理解析研究所講究録
1010 巻 1997 年 167-173

Computational Playability of Backward-Induction Solutions and Nash Equilibria:

An Application of Recursion Theory to Game Theory

TASHIRO Hidetoshi

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

E-mail Address: gem5104@srv.cc.hit-u.ac.jp Institute Telephone Number: ++84-425-80-8192

Correspondence Postal Address: Shimizu 1-17-13-203, Suginami, Tōkyō, 167, Japan

June, 1997

Abstract

This paper considers the playability of noncooperative game solutions from the viewpoint of players' computational ability. We construct a two-person two-stage game with perfect information such that payoff functions are computable but no backward induction strategy is computable. Nevertheless we show the decidability of Nash equilibria of the game. These results mean that backward induction solutions cannot be played in practice because it is impossible to supply the players with effective instructions regarding how they should find the solutions, although Nash equilibria are playable.

Keywords: Playability; Backward-induction solution; Computability; Nash equilibrium; Decidability; Kleene's T-predicate

1 Introduction

In this paper we consider the effective playability of backward induction solutions and Nash equilibria of a game under the assumption that the players should compute effectively. For a solution of a game to be playable, it should be computable in the sense that there is an algorithm, *i.e.* a Turing machine, to find it.

In the next section we construct a two-person two-stage game with computable payoff functions Since payoff functions are computable, it may be conjectured that solutions of the game would be also computable. However the conjecture is false. Indeed,

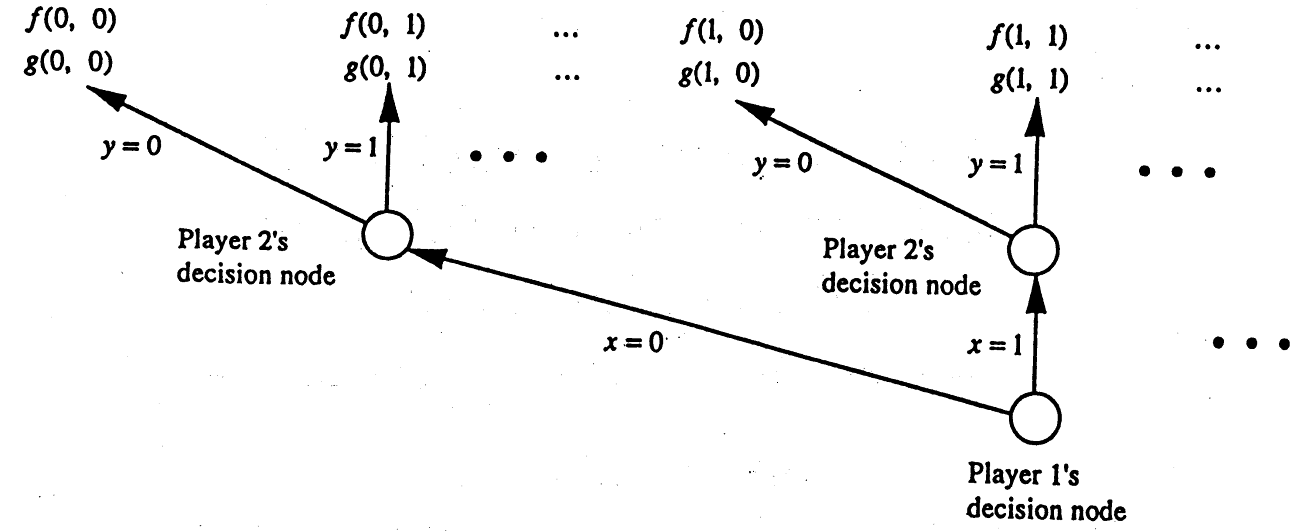

Figure 1: The Game Tree

we prove that while the game has a backward induction solution, no backward induction solution is computable. Nevertheless we show that there exists an algorithm to decide whether or not a given strategy profile is a Nash equilibrium of the game. These results mean that the solutions of the constructed game cannot be played in practice because it is impossible to supply the players with effective instructions regarding how to find the solutions.

2 A two-person two-stage game with computable payoff functions and the computational playability of its solutions

Construction of the Game Both players, 1 and 2, have countably many feasible actions, *i.e.* each player's action space is $\mathrm{N} = \{0, 1, 2, \ldots\}$. 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 $g(x, y)$.

Before defining players 1 and 2's specific payoff functions, we explain Kleene's T-predicate $T_1(z, x, y)$, which is the key to the construction of computable payoff functions. Kleene's T-predicate $T_1(z, x, y)$ is a particular computable predicate (see Kleene (1952, p.281)). 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 (see

Davis (1958, pp.57–58)). 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. Neither of the predicates $\exists y T_1(x,x,y)$ nor $\forall y \neg T_1(x,x,y)$ is computable.[1] By the definition of $T_1(z,x,y)$ (see Kleene (1952, p.281)), $T_1(0,x,y)$ does not hold for any x and any y.

Define players 1 and 2's payoff functions f and g as follows:

$$(1) \qquad f(x,y) := \begin{cases} y & \text{if } T_1(y,y,x), \\ x+y+1 & \text{otherwise;} \end{cases}$$

$$(2) \qquad g(x,y) := \begin{cases} x & \text{if } T_1(x,x,y), \\ x+y+1 & \text{otherwise.} \end{cases}$$

Since $T_1(x,x,y)$ is a computable predicate, f and g are computable functions from $\mathbf{N} \times \mathbf{N}$ to $\mathbf{N}$.

Computational Playability of Backward Induction Solutions 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.

Proposition 1 *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. □

However, the existence of backward induction solutions does not imply their playability. Indeed, we show the following theorem.

Theorem 1 *For any backward induction solution (x^*, ψ) of the game, the backward induction strategy ψ is not computable.*

Proof Let (x^*,ψ) be any backward induction solution of the game. Then

$$(3) \qquad g(\,x,\psi(x)) = \min_y\, g(x,y) = \begin{cases} x & \text{if } \exists y\, T_1(x,x,y), \\ x+1 & \text{otherwise.} \end{cases}$$

Now we prove the following:

$$(4) \qquad \forall x\,[\,\exists y\, T_1(x,x,y) \iff T_1(x,x,\psi(x))\,].$$

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

Consider any x such that $\exists y\, T_1(x,x,y)$ holds. Then by (3) we have $g(x,\psi(x)) = x$. Therefore by (2) we have $T_1(x,x,\psi(x))$. Conversely $T_1(x,x,\psi(x))$ implies $\exists y\, T_1(x,x,y)$. Consequently (4) holds.

Since $\exists y\, T_1(x,x,y)$ is not computable, by (4) the predicate $T_1(x,x,\psi(x))$ is not computable. This implies that ψ is not computable. □

Computational Playability of Nash Equilibria Consider the strategic form of the game defined above. A pair $(x^*,\, y^*)$ of actions is said to be a *Nash equilibrium* of the game iff $\forall x\, [\, f(x^*,\, y^*) \leq f(x,\, y^*)\,]$ and $\forall y\, [\, g(x^*,\, y^*) \leq g(x^*,\, y)\,]$.

There exists a Nash equilibrium of the game defined above. Indeed, the pair $(0,0)$ is a Nash equilibrium. By Theorem 1 it may be conjectured that no Nash equilibrium is decidable. However the conjecture is false. Indeed, we prove the following theorem:

Theorem 2 *There exists an algorithm to decide whether or not a given pair (x,y) of actions is a Nash equilibrium of the game.*

Proof The definition of Nash equilibrium is written as $f(x^*,\, y^*) = \min\limits_x f(x,\, y^*)$ and $g(x^*,\, y^*) = \min\limits_y g(x^*,\, y)$. Now we prove that a given pair (x^*, y^*) is a Nash equilibrium of the game if and only if

(5) $$[T_1(x^*,x^*,y^*),\ T_1(y^*,y^*,x^*),\ x^* \neq 0 \text{ and } y^* \neq 0] \text{ or } [x^* = y^* = 0].$$

By (1),

(6) $$f(x^*,\, y^*) = \begin{cases} y^* & \text{if } T_1(y^*,y^*,x^*), \\ x^* + y^* + 1 & \text{otherwise.} \end{cases}$$

On the other hand, again by (1),

(7) $$\min_x f(x,\, y^*) = \begin{cases} y^* & \text{if } \exists w\, T_1(y^*,y^*,w), \\ y^* + 1 & \text{otherwise.} \end{cases}$$

Similarly, by (2) we have

(8) $$g(x^*,\, y^*) = \begin{cases} x^* & \text{if } T_1(x^*,x^*,y^*), \\ x^* + y^* + 1 & \text{otherwise.} \end{cases}$$

Moreover it follows that

(9) $$\min_x g(x^*,\, y) = \begin{cases} x^* & \text{if } \exists w\, T_1(x^*,x^*,w), \\ x^* + 1 & \text{otherwise.} \end{cases}$$

Suppose $x^* = y^* = 0$. Then, since $T_1(0,0,0)$ does not hold, by (6) we have $f(x^*,\, y^*) = 1$. On the other hand, since $\exists w T_1(0,0,w)$ does not hold, by (7) we have $\min\limits_x f(x,\, y^*) = 1$. Thus $f(x^*,\, y^*) = \min\limits_x f(x,\, y^*)$. Similarly by (8) and (9) $g(x^*,\, y^*) = 1 = \min\limits_y g(x^*,\, y)$. Hence $(0,0)$ is a Nash equilibrium.

Suppose that $x^* \neq 0$ but $y^* = 0$. Then, by (6) we have $f(x^*, y^*) = x^* + 1 \neq 1$, but by (7) we have $\min_x f(x, y^*) = 1$. Thus $f(x^*, y^*) \neq \min_x f(x, y^*)$.

Similarly, if $x^* = 0$ but $y^* \neq 0$, then $g(x^*, y^*) \neq \min_y g(x^*, y)$.

Suppose that $x^* \neq 0$ and $y^* \neq 0$. Then $x^* + y^* + 1 \neq y^* + 1$ and $x^* + y^* + 1 \neq x^* + 1$. Thus, if $f(x^*, y^*) = \min_x f(x, y^*)$, then $T_1(y^*, y^*, x^*)$. On the other hand, if $g(x^*, y^*) = \min_y g(x^*, y)$, then $T_1(x^*, x^*, y^*)$. Hence if $x^* \neq 0$, $y^* \neq 0$ and (x^*, y^*) is a Nash equilibrium, then both $T_1(y^*, y^*, x^*)$ and $T_1(x^*, x^*, y^*)$ hold. Conversely $T_1(y^*, y^*, x^*)$ implies $f(x^*, y^*) = y^* = \min_x f(x, y^*)$, and $T_1(x^*, x^*, y^*)$ implies $g(x^*, y^*) = x^* = \min_y g(x^*, y)$. Consequently a given pair (x^*, y^*) is a Nash equilibrium if and only if (5) holds.

By the computability of $T_1(z, x, y)$, the condition (5) is computable. Therefore it is decidable whether or not a given (x^*, y^*) is a Nash equilibrium. □

3 Concluding remarks

Open Problems It is open how to construct a game with computable payoff functions such that the Nash equilibria are undecidable.

It is also open what constraints on the players' payoff functions suffice for the computational playability of backward induction solutions and Nash equilibria. This problem is closely related to the formulation of bounded rationality.

Moreover, while in this paper we discussed only pure strategies, it is unsolved how to introduce some random device for mixing pure strategies. This problem is connected with the computational complexity of mixed strategies.

Furthermore, assume that players 1 and 2 play repeatedly the game defined in the above section, and that player 1 knows that player 2 uses the same computable response function but player 1 may not know *which* computable strategy player 2 is using all the time. By Theorem 1, since player 2 uses a computable strategy this function is not the backward induction strategy. Hence player 1 cannot deduce this strategy only from player 2's payoff function g. 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. This problem is an extension of one raised by Rabin (1957). Although he proved a positive result, his discussion cannot be directly applied to the game defined in this paper because his discussion was dependent on the win-lose property. This problem is pertinent to the possibility of effective learning.

Rabin (1957)'s Pioneering Work The motive of this paper is to extend Rabin (1957)'s result on the computational playability of winning strategies. He considered a game such that two players, 1 and 2, 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. Since Rabin discussed two-person win-lose games, players 1 and 2's payoff functions f and g can be formulated into the zero-one valued functions as follows:

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

$$f(x,y,z) + g(x,y,z) \; = \; 1.$$

where h is a computable function, and therefore so are both f and g. 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 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 player who in theory can always win, cannot do so in practice because it is impossible to supply him with effective instructions regarding how he should play in order to win."(Rabin (1957, p.148))

The Importance of Computability We employed the concept of computability as a criterion for the playability of game solutions. Indeed, while the other epistemological concepts, *e.g.* computational complexities, are relative, as Gödel (1936) writes, "the notion 'computable' is in a certain sense 'absolute', while almost metamathematical notions otherwise known (for example, provable, definable, and so on) quite essentially depend upon the system adopted."[2] For details, see Gödel (1946), Kleene (1952, § 62) and Odifreddi (1989, Section I.8).

Acknowledgements

The author would like to express his deep thanks to Professor NAGASHIMA Takashi (Hitotsubashi University), with whom the author has talked over all details, for his continual and most efficient suggestions, comments, discussions and encouragement. Furthermore the author would like to acknowledge Professor KANEKO Mamoru (University of Tsukuba) for his invaluable comments, discussions and encouragement. Moreover the author would like to acknowledge Professor SUZUMURA Kôtarô (Hitotsubashi University) for his comments, suggestions and encouragement. The author would also like to thank Ms HASEGAWA Kaori (Hitotsubashi University) for her encouragement, Ms Irene WATSON (Hitotsubashi University) and Ms Alessandra CALOSSO (International Centre for Economic Research, Turin) for their helpful kindness, and further Professor Wiebe VAN DER HOEK (Utrecht University) for his comment. Last, but not least, the author would like to thank Professor HIROKAWA Midori (Hôsei University) for her help and encouragement in the early stages of his study. Of course, any possible errors and misunderstandings are entirely due to the author.

[2] Nevertheless, it is also important to consider the definability and deducibility of game solutions from the viewpoint of players' logical ability. See Kaneko and Nagashima (1996, 1997).

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

[5] Kaneko M. and Nagashima T. (1997) "Game Logic and its Applications II," forthcoming in *Studia Logica.*

[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.