ゲーム木と戦略的複雑度

無藤　望

本稿では，ゲーム理論で登場するゲーム木の上で定義される戦略の複雑度を導入し，複雑度を考慮するときゲームの帰結がどのように変化するかを議論する．特に，逐次手番囚人のジレンマの無限回繰り返しゲームにおいて協力が達成されるかを見るためには，戦略の複雑度を慎重に定義する必要があることを示す．

キーワード：ゲーム理論，ゲーム木，囚人のジレンマ，繰り返しゲーム，戦略的複雑度，複式複雑度

われわれは日々いろいろな意思決定を行っている．その帰結として表れる行動は，自分だけでなく他者にもしばしば影響を与える．逆に他者の行動から，自分の意思決定が影響を受けることも多い．例えば，自動車を運転するとき，多くの他人が自分と同じ道を通行しようとすると，渋滞になってより時間がかかってしまう．

このように各人の行動が相互に影響し合う場面が，ゲーム理論の考察対象となる状況である．

1. ゲーム木と展開形ゲーム

1.1 戦略形ゲーム

冒頭に述べたような，複数の主体の行動が相互に影響し合う状況をゲーム理論では「ゲーム」と呼ぶ．そして，そのゲームに関わる主体を「プレーヤー」と呼ぶ．

一般には，プレーヤーはさまざまなタイミングで意思決定を行うことがあるが，単純な場合として，全プレーヤーが一斉に行動を選択するゲームを「戦略形ゲーム」または「標準形ゲーム」という．各プレーヤーの行動がすべて定まると，ゲームの結果が決まる．この結果を各プレーヤーが評価した実数値を「利得」と呼ぶ．

今，n 人のプレーヤーがいるとし，$1,\ldots,n$ という自然数で表す．それぞれのプレーヤー i $(i=1,\ldots,n)$ は行動集合 A_i から行動を選択するとしよう．行動の組 $(a_1,\ldots,a_n)\in A_1\times\cdots\times A_n$ が選択されたとき，プレーヤー i の利得は，あらかじめ与えられた実数値の利得関数 $u_i : A_1\times\cdots\times A_n\to\mathbb{R}$ によって $u_i(a_1,\ldots,a_n)$ と定まる．これらの 3 つの要素をまとめた $\bigl(\{1,\ldots,n\},(A_1,\ldots,A_n),(u_1,\ldots,u_n)\bigr)$ が戦略形ゲームである．

有名な戦略形ゲームに「囚人のジレンマ」と呼ばれるものがある．これは，2 人のプレーヤー 1, 2 が協力する（C）か裏切る（D）かを個別に選択するゲームで，行動集合は $A_i=\{C_i,D_i\}$ で表される（$i=1,2$）．2 人の行動によって，ここでは表 1 のように利得が定まるとする．この表では，括弧内の左の数字がプレーヤー 1 の利得，右がプレーヤー 2 の利得を表す．例えば，2 人とも協力するなら 2 人とも 2 の利得を得られるが，プレーヤー 2 が協力しているときにプレーヤー 1 が裏切ると，プレーヤー 1 の利得は 3 となってより大きな利得をえられるのに対し，2 の利得は –1 と大きく下がってしまう．そして，2 人とも裏切ると，利得はともに 0 となってしまう．これは，自分だけが協力したときよりは大きいが，2 人が協力したときよりも小さい利得である．このように，互いに裏切るよりも，互いに協力することが 2 人にとって望ましいのに，裏切りがより大きな利得をもたらすために，協力状態を維持できないというのが囚人のジレンマの特徴である．

表 1　囚人のジレンマの利得表の例

	C_2	D_2
C_1	$(2,2)$	$(-1,3)$
D_1	$(3,-1)$	$(0,0)$

相手が裏切っているならば，自分が協力しようとしても損をしてしまうため，互いに裏切る状態はそのまま維持される．このように，1 人のプレーヤーが行動を変えてもそのプレーヤーが得をできないとき，その行動の組が「ナッシュ均衡」であるという．

定義 1. ある行動の組 $(a_1,\ldots,a_n)$ は次の性質をみたすときナッシュ均衡であるという．

- すべての $i=1,\ldots,n$ とすべての $a_i'\in A_i$ に対して $u_i(a_1,\ldots,a_i,\ldots,a_n)\geq u_i(a_1,\ldots,a_i',\ldots,a_n)$ が成り立つ．

ナッシュ均衡は，[8] で導入された概念で，ゲーム理論の中核を成すものである．

むとう　のぞむ
一橋大学大学院経済学研究科
〒 186–8601 東京都国立市中 2–1

1.2　戦略形と展開形

ゲーム理論の教科書では，上記の戦略形ゲームと「展開形ゲーム」を区別して解説することが多い．展開形ゲームでは，各プレーヤーはゲームのルールによって定められたタイミングで，行動を 1 回あるいは複数回選択していく．このように行動のタイミングを考えていることは，戦略形ゲームにはない特徴である．

展開形ゲームのルールは「ゲーム木」で表される．ゲーム木を数学的に定義するのは煩雑になるため，例を挙げることにより解説する[1]．ゲーム木によって規定された順序に従って各プレーヤーが行動を選択した結果に実数値の利得を対応させる関数が，展開形ゲームにおける利得関数である．展開形ゲームは，プレーヤーの集合にゲーム木と利得関数を組み合わせたものである．

ゲーム木の例として，囚人のジレンマの行動順序を逐次的にした展開形ゲームを考えてみよう．プレーヤー 1 が C_1, D_1 のいずれかを選択した後，プレーヤー 2 が C_2, D_2 のいずれかを選択する．これは図 1 のゲーム木で表される．ここでは，ゲームの時間的推移をゲーム木における上から下への移動として表す．まず，一番上の始点でプレーヤー 1 が C_1, D_1 のいずれかを選択する．もし C_1 を選ぶと，ゲームは左の点に移り，プレーヤー 2 が C_2, D_2 を選択する．D_1 を選ぶと，2 は右の点で選択することになる．このように，2 人の選択によって，ゲーム木の 4 つの終端点のいずれかに到達する．この例では，利得関数として表 1 のものを与えておき，図 1 とこの利得関数を組み合わせて得られる展開形ゲームを Γ と表すことにする．

Γ では後から行動するプレーヤーが，それ以前に選択された行動を知ることができる状況だった．このような展開形ゲームを「完全情報ゲーム」という．

完全情報でないゲーム木の例として図 2 を考えてみよう．最初にプレーヤー 1 が C_1, D_1 のいずれかを選ぶと，ゲームは中段の点に移る．ここで 2 つの点が円弧で結ばれているのは，行動の手番となるプレーヤー 2 にとって，現在どちらの点にいるかを知り得ないことを表す．つまり，プレーヤー 2 はプレーヤー 1 の行動がわからないまま C_2, D_2 を選択しなくてはならないという点で，図 1 のゲーム木とは異なる．このようにゲーム木上のいくつかの点を円弧で結ぶことで表される点の集合を「情報集合」と呼ぶ．2 点以上からなる情報集合のあるゲームは完全情報でない．

[1] ゲーム木の概念は [10] にすでに表れている．展開形ゲームに詳しい教科書として [7], [11] を挙げておく．

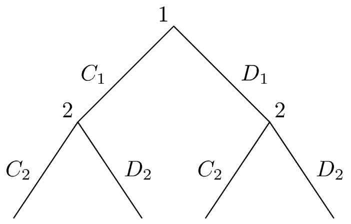

図 1　逐次手番囚人のジレンマ Γ のゲーム木

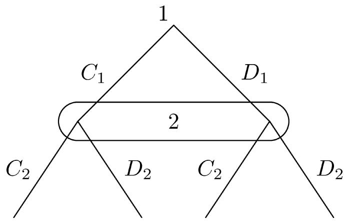

図 2　同時手番囚人のジレンマ G のゲーム木

図 2 のゲーム木では，プレーヤー 2 は 1 の選択を知らないまま自分の行動を選ぶ．これは 2 人が同時に行動する戦略形ゲームとしての囚人のジレンマと同じである．したがって，図 2 は同時手番囚人のジレンマのゲーム木といえる．任意の戦略形ゲームは，このようなゲーム木によって展開形ゲームとして表すことができる．

図 2 のゲーム木と表 1 の利得関数を組み合わせた展開形ゲームを G で表すことにする．

1.3　展開形ゲームにおける戦略

図 1 の逐次手番囚人のジレンマで，各プレーヤーがどのように行動できるかを考えよう．プレーヤー 1 は C_1, D_1 を単純に選択するだけだが，プレーヤー 2 は 1 の行動を見てから C_2, D_2 を選択できる．したがって，

1 が C_1 を選んだら C_2 を選ぶが，D_1 なら D_2 を選ぶ　(1)

のように，相手の行動に応じて異なる行動を選ぶことが可能である．それに対し，

1 が C_1, D_1 のどちらを選んでも D_2 を選ぶ　(2)

というように，相手の行動によらずに行動を決めることも可能である．上記のように，プレーヤー 2 は 1 の行動に応じて自身の行動を選択する．この選択のしかたを「戦略」と呼ぶ．したがって，プレーヤー 2 の戦略は $\{C_1, D_1\}$ から $\{C_2, D_2\}$ への写像として表される．プレーヤー 1 が D_1 を選ぶ場合，上記の戦略 (1) と戦略 (2) は同じ結果をもたらすが，これらは異なる戦略である．戦略は，実際には選択されない行動の後の選択もすべて規定するものであることに注意が必要

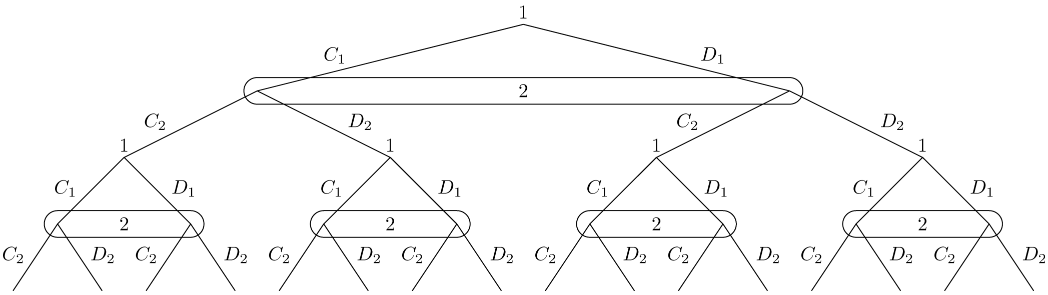

図 3　2 回繰り返し同時手番囚人のジレンマのゲーム木

である．

各プレーヤーの戦略が与えられると，ゲームの結果としてどの終端点に到達するかが確定する．展開形ゲームの利得関数が与えられているとすると，上記のように戦略を定めることにより，定義 1 と同様にして展開形ゲームにおけるナッシュ均衡を定義できる．

2. 繰り返しゲーム

前節では，それぞれのプレーヤーが 1 度ずつ行動する単純な例を見たが，一般には 1 人のプレーヤーが何度も行動するゲームもある．図 2 の同時手番囚人のジレンマを，2 回繰り返すゲームが一例である．このゲームは，図 3 に示すようにより大きなゲーム木を持つ．はじめに同時手番囚人のジレンマがプレーされ，その結果を両プレーヤーが確認したあとに 2 度目の同時手番囚人のジレンマが行われる．このように同じゲームを繰り返すゲームを「繰り返しゲーム」と呼ぶ．図 3 では 2 回繰り返しを示したが，繰り返し回数を無限とする「無限回繰り返しゲーム」を考えることも多い．

1.1 節で，囚人のジレンマにおいて協力が達成されないことを指摘した．しかし，現実にはさまざまな場面で協力行動が観察できる．この不整合に関して，ゲーム理論が提供するひとつの説明は，現実には同じような状況が繰り返し出現する場合があることに基づく．そこで，同時手番囚人のジレンマの無限回繰り返しゲーム G^∞ を考えよう．

このゲームは第 1 期から始まり，$t = 1, 2, \ldots$ と無限に続く．第 t 期にプレーヤー 1 が a_1^t（$= C_1, D_1$）を，2 が a_2^t（$= C_2, D_2$）を選択したとき，プレーヤー $i = 1, 2$ の利得は

$$(1-\delta)\sum_{t=1}^{\infty}\delta^{t-1}u_i(a_1^t, a_2^t)$$

で与えられると定義する．ここで，u_1, u_2 は表 1 の利得関数であり，δ は $0 < \delta < 1$ をみたす実数で，「割引因子」と呼ばれる．例えば，毎期 (C_1, C_2) が選択され，両者が $u_i(C_1, C_2) = 2$ の利得を得たとすると繰り返しゲーム全体での利得も

$$(1-\delta)\sum_{t=1}^{\infty}\delta^{t-1}\cdot 2 = (1-\delta)\cdot\frac{1}{1-\delta}\cdot 2 = 2$$

となる．

このとき次のような「切り替え戦略（trigger strategy）」が考えられる：「最初は協力する．相手が協力する限り自分も協力するが，もし相手が裏切ったらその後はずっと裏切り続ける」

相手が切り替え戦略をとっているときに，自分がどのように行動すると利得を最大化できるか考えてみよう．もし自分がずっと協力すると，相手も協力を続けるので全体で 2 の利得が得られる．一方，裏切るとその期の利得は 3 に増えるが，その後は相手も裏切ってくるので利得は 0 に落ちる．全体としては

$$(1-\delta)\left(3+\sum_{t=2}^{\infty}\delta^{t-1}\cdot 0\right) = 3(1-\delta)$$

の利得が得られる．これらを比較すると，$\delta \geq 1/3$ ならば毎期協力することが最適になることがわかる．

毎期協力するという行動はそのプレーヤーの切り替え戦略によっても実現されるので，相手が切り替え戦略をとっているとき，自分も切り替え戦略をとることで自分の利得を最大化できることがわかる．これは相手にとっても同じなので，互いに切り替え戦略をプレーすることにより，無限回繰り返しゲームで協力がナッシュ均衡として達成されることがわかった．

このように，無限回繰り返しゲームでは，1 回限りのゲームでは達成できないような高い利得がナッシュ均衡として実現可能である．これは広い範囲のゲームで成立する結果であり，「フォーク定理」と呼ばれている．

3. 繰り返しゲームにおける戦略的複雑度

前節での議論は広汎に成り立つ重要な結果であるが，それに対する批判も少なからず存在する．ここではそのひとつとして，戦略の複雑さを取り上げる．

前節で，相手が切り替え戦略をとっているとき「毎期協力するという行動はそのプレーヤーの切り替え戦略によっても実現される」と書いた．しかし，相手が切り替え戦略をとっているとわかっているのであれば，自分は裏切るべきかどうかを気にせずに，単に毎期協力するという戦略をとっていればよいのではないかという疑問が考えられる．何が起ころうとも毎期協力するという戦略は，相手が裏切ったら自分も裏切るという切り替えのある戦略よりも単純と考えるのは自然だろう．同じ利得が得られるならより単純な戦略を選ぶとすれば，相手が切り替え戦略をとっているときに，自分は切り替え戦略をとらずに，毎期協力する戦略をとったほうが良い．したがって，プレーヤーがより単純な戦略を選好する場合，互いに切り替え戦略をプレーするのは均衡とは言えない．

以下では，戦略の複雑さとは何であるかを同時手番囚人のジレンマの無限回繰り返しゲーム G^∞ において定義し，その複雑さを加味したナッシュ均衡を導入する．

3.1 戦略の有限オートマトンによる表現

繰り返しゲームにおける戦略は，一般には，前期までに選択された行動の列に今期の行動を対応させる写像として定義される．しかし，この記法は煩雑になりがちなので（有限）オートマトンによる表現がよく用いられる[2]．例えば，図 4 はプレーヤー 1 の切り替え戦略を表現するオートマトンである．切り替え戦略は 2 つの「状態」によって表現される．図 4 において，短い矢印のついた左の円は第 1 期におけるゲームの状態（初期状態）を表す．第 t 期にこの状態にあるとき，プレーヤー 1 は協力し C_1 を選択する．2 が第 t 期に C_2 を選択すると，1 はそれを第 t 期の終わりに観察し，状態は矢印に従って左側にとどまる．そして第 $t+1$ 期も協力する．しかし，2 が第 t 期に D_2 を選択すると，状態は矢印に従って右の円に遷移し，1 は第 $t+1$ 期には D_1 を選択することになる．いったん右の状態に遷移すると，その後は C_2, D_2 のいずれを 2 が選んでもずっと右の状態にとどまり続け，1 は裏切り続けることになる．

[2] オートマトン一般については [4] を参照のこと．

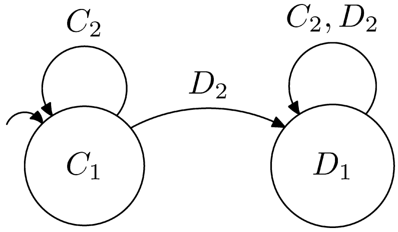

図 4 切り替え戦略を表現する有限オートマトン

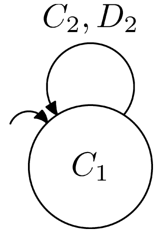

図 5 毎期協力する戦略を表現する有限オートマトン

一方，毎期協力するプレーヤー 1 の戦略を表現するオートマトンは図 5 で表される．状態は 1 つしかなく，プレーヤー 2 がどう行動しても C_1 を選択する．

このように考えると，状態と遷移のしかたを定義すれば，どんな戦略でも表現可能であることがわかる．

3.2 戦略の複雑度

このようにオートマトンで戦略を表現すると，切り替え戦略のほうが毎期協力する戦略よりも複雑であることが見てとれるようになる．Aumann [2] は，オートマトンの状態数（図にある円の個数）によってその戦略の複雑度を定義することを提案した．この定義に従うと，図 4 の複雑度は 2 であるので，複雑度 1 の図 5 よりも複雑であると言える．そして，[1] はその定義に基づき，一般の 2 人同時手番の無限回繰り返しゲームで，プレーヤーが単純な戦略を選好する場合のナッシュ均衡を分析した．

定義 2. 戦略の組が定義 1 の意味でナッシュ均衡であり，さらに，どのプレーヤーも，利得を保ちながらより複雑度の低い戦略に逸脱できないとき，その戦略の組を「複雑度込みのナッシュ均衡」と呼ぶ．

定義 2 でも，定義 1 と同様にプレーヤーは利得を最大化している．複数の戦略が等しい利得をもたらすときにのみ，第 2 の選択基準として戦略の複雑度を用いる．

本節の冒頭で解説したとおり，切り替え戦略は複雑度込みのナッシュ均衡を構成しない．しかし，[1] は図 6

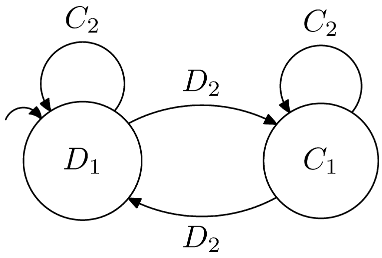

図 6　「鞭と飴」戦略を表現する有限オートマトン

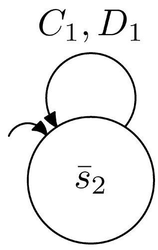

図 7　Γ^∞ におけるプレーヤー 2 の毎期 $\bar{s}_2$ をとる戦略を表現する有限オートマトン

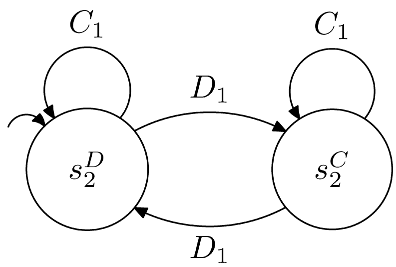

図 8　Γ^∞ におけるプレーヤー 2 の「鞭と飴」戦略を表現する有限オートマトン

のオートマトンで表現される「鞭と飴」(tat for tit) 戦略を互いにとることが,（割引因子が高いとき）複雑度込みのナッシュ均衡となることを示した[3]．この戦略では，第 1 期に裏切るが，そのとき相手も裏切ったことを確認すると，右の状態に遷移しその後は相手が協力する限り協力を続ける．したがって，どちらも鞭と飴戦略をとっていると，第 1 期は互いに裏切るが，第 2 期以降はずっと協力が続く．このように裏切りと協力がどちらも実際の選択として実現されるようにするには，最少でも 2 つの状態が必要なので，より少ない状態数で同じ利得を実現することはできない．

このように，最初に裏切りが起こるという限定はあるものの，無限に協力が続くという意味で，複雑度込みのナッシュ均衡でも協力が達成できることがわかった．

4.　逐次手番の繰り返しゲーム

前節では同時手番囚人のジレンマの無限回繰り返しゲーム G^∞ を考えた．次に，逐次手番囚人のジレンマの無限回繰り返しゲーム Γ^∞ を考えよう．逐次手番の場合，プレーヤー 1 の戦略集合は同時手番の場合と変わらないが，1.3 節で議論したように，プレーヤー 2 はその期の 1 の行動を観察してから行動できるので，戦略の集合がより大きくなる．そのため，戦略をオートマトンで表現するとき，その表現方法は 1 通りではない．ここでは [9] に従い，1 回限りのゲーム Γ での戦略を状態に対応させ，遷移は同時手番のときと同様に扱うことにする．

1 回限りのゲーム Γ での戦略に関して記号を導入しよう．1.3 節の (1) の戦略を $\bar{s}_2$ と書き，(2) の戦略を s_2^D と書くことにする．また，プレーヤー 1 の行動にかかわらず協力する戦略を s_2^C と書く．このとき，例えば，Γ^∞ において毎期 $\bar{s}_2$ を選択するプレーヤー 2 の戦略は図 7 で表現される．また，このゲーム Γ^∞ での鞭と飴戦略は図 8 のオートマトンで表される．

プレーヤー 1 が図 6 の鞭と飴戦略をとり，プレーヤー 2 が図 8 の鞭と飴戦略をとっているとする．このとき，前節で議論したとおり，第 1 期に互いに裏切った後，第 2 期以降は無限に協力が続く．ここでプレーヤー 2 が鞭と飴戦略の代わりに，図 7 の戦略をとったとしよう．それでも，1 が鞭と飴戦略をとっているなら，第 1 期に裏切り，第 2 期以降は協力するという行動の列は同じになる．よって，プレーヤー 2 の利得は同じになる．しかし，図 7 のオートマトンの状態数は 1 であり，図 8 の状態数 2 よりも小さい．よって，このオートマトンの状態数を複雑度と定義するならば，互いに鞭と飴戦略をとるのは複雑度込みのナッシュ均衡ではないことになる．

[9] は，オートマトンの状態数を複雑度と定義した上で，一般の 2 人完全情報ゲームの無限回繰り返しゲームにおいて上と同様の議論を行い，1 回限りのゲームにおけるひとつのナッシュ均衡を毎期繰り返すことだけが複雑度込みのナッシュ均衡となることを示した．1 回限りの逐次手番囚人のジレンマ Γ においては互いに裏切ることだけがナッシュ均衡となるので，Γ^∞ では，この複雑度込みのナッシュ均衡において協力は全く達成されないことになる．

[3] 図 6 の戦略は文献でさまざまな呼び方がされる．有名な「飴と鞭 (tit for tat)」戦略とは逆に，この戦略では最初に裏切りがあった後協力に移行することから，ここでは「鞭と飴」という名を採用した．

5. 複式複雑度

前節では，逐次手番囚人のジレンマの無限回繰り返しゲーム Γ^{∞} において，協力が達成されないという否定的な結果が導かれた．しかし，これはオートマトンの状態数を戦略の複雑度と仮定したことに依存している．

1 回限りの逐次手番囚人のジレンマ Γ において，(1) の戦略 $\bar{s}_2$ と必ず協力する戦略 s_2^C を比較しよう．前者では相手の行動に依存して自分の行動を決める必要があるが，後者では相手がどう行動しても無関係に協力すればよい．その点で，前者のほうが後者よりも複雑であると言えないだろうか．オートマトンの状態数を考えるだけでは，このような，1 回限りのゲームにおける戦略の複雑さの違いを無視してしまうことになる．

そこで，[6] は一般の 2 人逐次手番ゲームの無限回繰り返しゲームにおいて，「複式複雑度」という新しい複雑度を定義した．この複雑度は，単に状態数を数えるのではなく，1 回限りのゲームにおける戦略の複雑さによって，その戦略と対応づけられた状態に重み付けをしたうえで足し合わせる．例えば，図 7 のオートマトンではひとつの状態で $\bar{s}_2$ がとられている．この $\bar{s}_2$ は，プレーヤー 1 の行動に応じて C_2, D_2 という 2 つの行動を選ぶ可能性があるので，状態は 1 つでも複雑度は 2 と定義する．一方，図 8 のオートマトンでは状態は 2 つあるが，どちらの状態でも s_2^D, s_2^C という単一の行動しか選択しない戦略が対応づけられているので，単に足し合わせて複雑度は $1+1=2$ と数える．逐次手番囚人のジレンマにおけるプレーヤー 2 の一般のオートマトンに関しては，次のように定義される．

定義 3. プレーヤー 2 の戦略を表現するオートマトンにおいて，1 の行動に応じて C_2, D_2 のどちらもとり得るような状態の個数を q，片方しか取り得ないような状態の個数を r とすると，複式複雑度は $2q+r$ で定義される．

[6] は，この複雑度を採用したうえで，一般の 2 人逐次手番ゲームの繰り返しゲームにおいて複雑度込みのナッシュ均衡の特徴付けを行った．単に状態数を複雑度とする前節の議論と異なり，図 7 と図 8 は共に複式複雑度 2 となるため，図 8 から図 7 に変えてもより単純になることはない．したがって，互いに鞭と飴戦略をとり合うことが複雑度込みのナッシュ均衡となる．

よって，逐次手番囚人のジレンマの無限回繰り返しゲーム Γ^{∞} において協力が達成されないという否定的な結果は，オートマトンの状態数を複雑度とする単純な解釈に依存するものであり，複式複雑度の下では協力が達成可能であることがわかった．

6. おわりに

本稿では，繰り返しゲームにおける戦略の複雑度を導入したうえで，囚人のジレンマの繰り返しゲームで，ナッシュ均衡において協力が達成可能であるというフォーク定理で一般にいわれる結果は，複雑度込みのナッシュ均衡を考えても同様に成り立つことを見た．ただし，逐次手番囚人のジレンマでは，複雑度の定義を慎重に考える必要がある．

より一般の繰り返しゲームでも，[2] 以降活発な議論が行われてきた．さらに近年の研究では，交渉ゲームや市場ゲームなど，繰り返しゲームでない展開形ゲームにも戦略的複雑度の概念が適用され，成果を上げている．これらの研究については，[3]，[5] のサーベイを参照されたい．

参考文献

[1] D. Abreu and A. Rubinstein, The Structure of Nash Equilibrium in Repeated Games with Finite Automata, *Econometrica*, **56**, 1259–1281, 1989.

[2] R. J. Aumann, Survey of Repeated Games, in *Essays in Game Theory and Mathematical Economics in Honor of Oskar Morgenstern*, V. Boehm and H. Nachthanp, eds., Wissenschaftsverlag Bibliographisches Institut, 1981.

[3] K. Chatterjee and H. Sabourian, Game Theory and Strategic Complexity, in *Encyclopedia of Complexity and System Science*, R. A. Meyers, eds., Springer, 2009.

[4] J. E. Hopcroft, R. Motwani and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (2nd ed.), Addison Wesley, 2000. 野崎昭弘，高橋正子，町田元，山崎秀記訳，『オートマトン言語理論 計算論 I, II〔第 2 版〕』，サイエンス社，2003.

[5] E. Kalai, Bounded Rationality and Strategic Complexity in Repeated Games, in *Game Theory and Applications*, T. Ichiishi, A. Neyman, and Y. Tauman, eds., Academic Press, 1990.

[6] N. Muto, Strategic Complexity in Repeated Extensive Games, Working Paper (2012).

[7] R. B. Myerson, *Game Theory: Analysis of Conflict*, Harvard University Press, 1997.

[8] J. Nash, Non-cooperative Games, *Annals of Mathematics*, **54** (1951), 286–295.

[9] M. Piccione and A. Rubinstein, Finite Automata Play a Repeated Extensive Game, *Journal of Economic Theory* (1993), **61**, 160–168.

[10] J. von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 1944 (reprinted in 2007). 銀林浩，橋本和美，宮本敏雄監訳，『ゲームの理論と経済行動 1, 2, 3』，ちくま学芸文庫，筑摩書房，2009.

[11] 岡田章，『ゲーム理論〔新版〕』，有斐閣，2011.