

オペレーションズ・リサーチ II (6)

田中 俊二

shunji.tanaka@okayama-u.ac.jp

本文書のライセンスは CC-BY-SA にしています



スケジュール

No.	内容
1	導入 (非線形最適化問題, ゲーム理論, 多目的最適化問題)
2	非線形計画 1 (勾配, ヘッセ行列, 凸性, 最適性条件, ニュートン法)
3	非線形計画 2 (最急降下法, 準ニュートン法, 共役勾配法, 信頼領域法)
4	非線形計画 3 (制約つき問題の最適性条件, KKT 条件, ペナルティ関数法, 2 次計画法, 逐次 2 次計画法)
5	ゲーム理論 1 (種々のゲーム, 標準形, 純粋戦略, 混合戦略, ナッシュ均衡)
6	ゲーム理論 2 (展開形ゲーム, 繰り返しゲーム)
7	多目的最適化 (パレート最適性, 重み付け法, ϵ 制約法, 重み付きメトリック法)

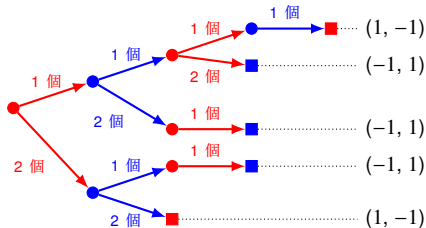
展開形ゲーム

展開形ゲーム (extensive form game) とは？

- ゲームの表現方法の一つ
- 交代手番ゲームや不確定ゲーム，完全情報ゲームなどを表すのに適している
- **ゲームの木** (game tree) を用いる

完全情報ゲームの例：石取りゲーム (交代手番ゲーム)

4 個の石がある．プレイヤー 1，プレイヤー 2，プレイヤー 1，... の順に 1 個または 2 個の石を取っていく．最後に石を取ったプレイヤーが負け (利得 -1)．勝ったプレイヤーの利得は 1．



ゲームの木

- 手番 (プレイヤー 1)
- 手番 (プレイヤー 2)
- 頂点 (プレイヤー 1 の勝利)
- 頂点 (プレイヤー 2 の勝利)

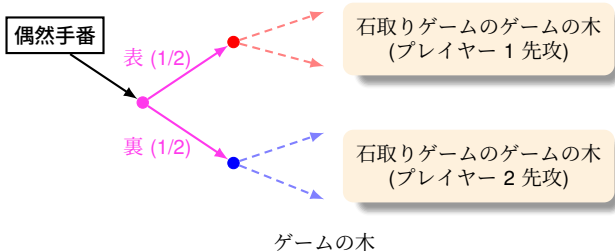
展開形ゲーム：偶然手番

不確定ゲームの例：コイントスで先攻・後攻を決めるゲーム

先ほどの石取りゲームの先攻・後攻をコイントスで決める．表が出たらプレイヤー 1 が先攻，裏が出たらプレイヤー 2 が先攻．

偶然手番 (chance move)

結果が偶然により決まる手番

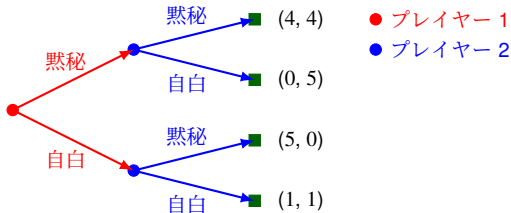


展開形ゲーム：不完全情報ゲーム

囚人のジレンマ改

1 \ 2	黙秘	自白
黙秘	(4, 4)	(0, 5)
自白	(5, 0)	(1, 1)

- 同時手番ゲームは交代手番ゲームに変換
- プレイヤー 2 がプレイヤー 1 の行動に応じて自分の行動を切り替えるのは不可能

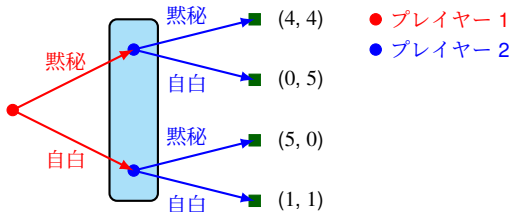


展開形ゲーム：不完全情報ゲーム

囚人のジレンマ改

1 \ 2	黙秘	自白
黙秘	(4, 4)	(0, 5)
自白	(5, 0)	(1, 1)

- 同時手番ゲームは交代手番ゲームに変換
- プレイヤー 2 がプレイヤー 1 の行動に応じて自分の行動を切り替えるのは不可能

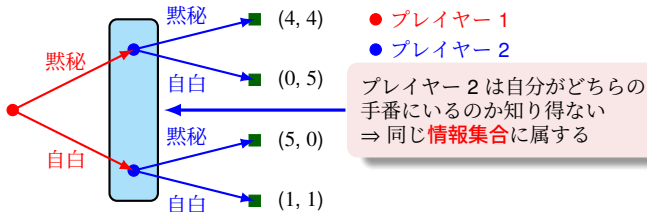


展開形ゲーム：不完全情報ゲーム

囚人のジレンマ改

1 \ 2	黙秘	自白
黙秘	(4, 4)	(0, 5)
自白	(5, 0)	(1, 1)

- 同時手番ゲームは交代手番ゲームに変換
- プレイヤー 2 がプレイヤー 1 の行動に応じて自分の行動を切り替えるのは不可能

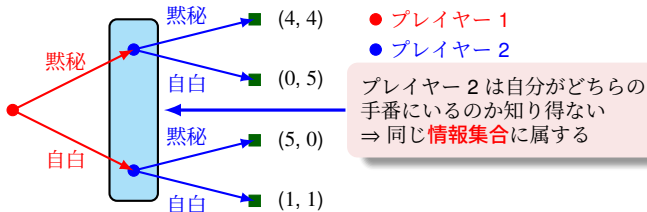


展開形ゲーム：不完全情報ゲーム

囚人のジレンマ改

1 \ 2	黙秘	自白
黙秘	(4, 4)	(0, 5)
自白	(5, 0)	(1, 1)

- 同時手番ゲームは交代手番ゲームに変換
- プレイヤー 2 がプレイヤー 1 の行動に応じて自分の行動を切り替えるのは不可能



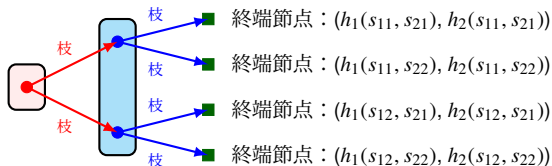
完全情報ゲーム

すべての情報集合がただ 1 つの手番からなるゲーム

展開形ゲームの要素

展開形ゲームの要素

- プレイヤー数： n
- ゲームの木
 - 節点 (node)：手番
 - 終端節点 (terminal node)：ゲーム終了
 - 枝 (edge)：行動
- 各プレイヤー i の手番の集合： P_i (P_0 は偶然手番)
- 偶然手番 (存在するなら) における確率
- 各頂点におけるプレイヤー i の利得関数： $h_i(s_1, \dots, s_n)$
 s_i はプレイヤー i の**戦略** (プレイヤーの行動計画)
- 各プレイヤーの**情報集合**
 - 情報集合同士は交わりを持たない
 - 同じ情報集合に属する手番は同じ数の枝 (行動) を持つ



展開形ゲームにおける戦略

局所戦略 (local strategy)

各情報集合における純粋戦略 (行動) ・ 混合戦略

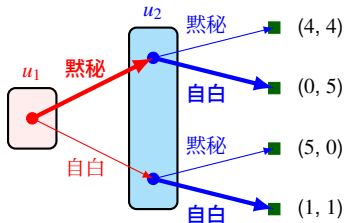
戦略

局所戦略の組

不完全情報ゲームにおける戦略の例：囚人のジレンマ

プレイヤー 1: u_1 における 黙秘, 自白

プレイヤー 2: u_2 における 黙秘, 自白

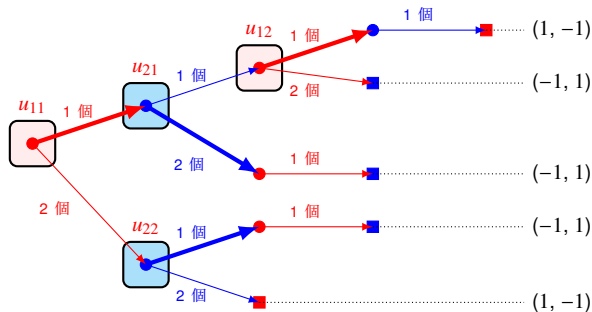


展開形ゲームにおける戦略 (続き)

完全情報ゲームにおける戦略の例：石取りゲーム

プレイヤー 1 : (u_{11} における 1 個・2 個, u_{12} における 1 個・2 個) の組

プレイヤー 2 : (u_{21} における 1 個・2 個, u_{22} における 1 個・2 個) の組



展開形ゲームにおけるナッシュ均衡

期待利得

プレイヤーの純粋戦略の組 (s_1, \dots, s_n) に対する期待利得： $H_i(s_1, \dots, s_n)$

偶然手番がなければ $H_i(s_1, \dots, s_n) = h_i(s_1, \dots, s_n)$

ナッシュ均衡

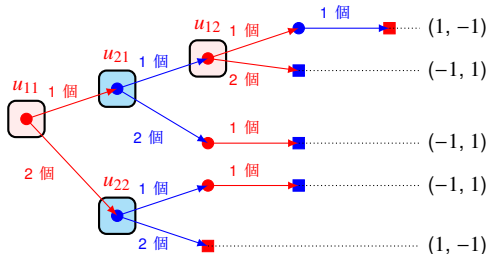
プレイヤーの純粋戦略の組 (s_1^*, \dots, s_n^*) が**ナッシュ均衡** (Nash equilibrium) であるとは、任意の i ($1 \leq i \leq n$) と任意の $s_i \in S_i$ に対して $H_i(s_i, s_{-i}^*) \leq H_i(s_i^*, s_{-i}^*)$ が成り立つことをいう。

混合戦略の場合も同様

後退帰納法 (backward induction)

- **完全情報**展開形ゲームに対する純粋戦略のナッシュ均衡の求め方
- 終端節点から逆方向に各手番の最適な行動を求めていく

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

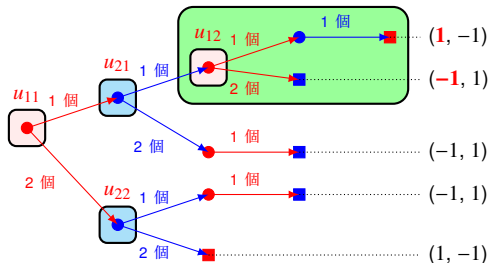
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

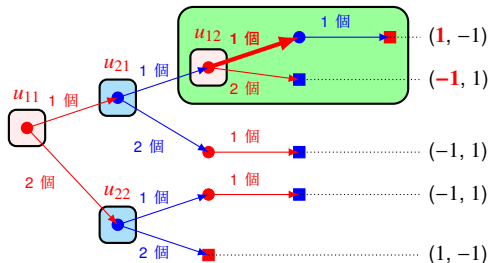
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

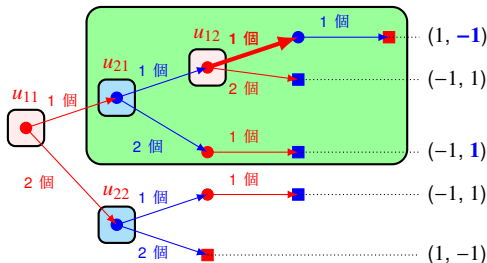
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

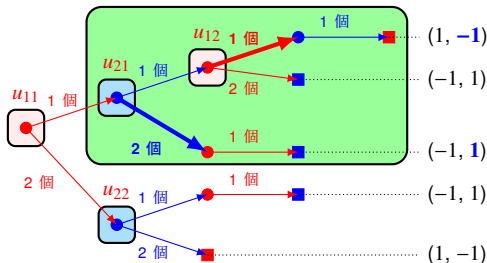
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

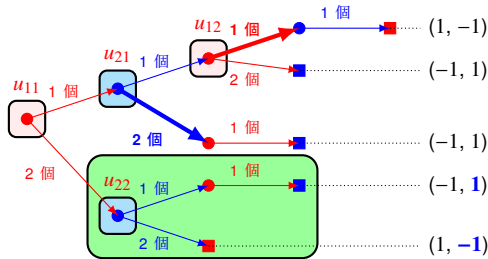
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

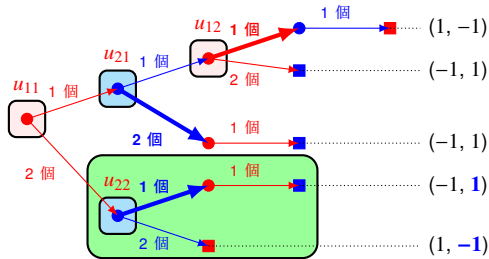
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

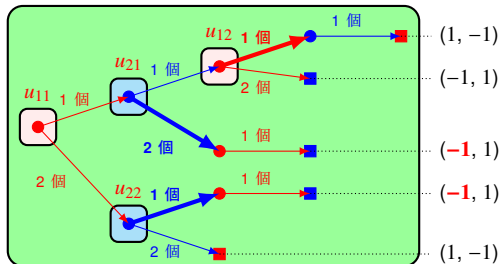
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

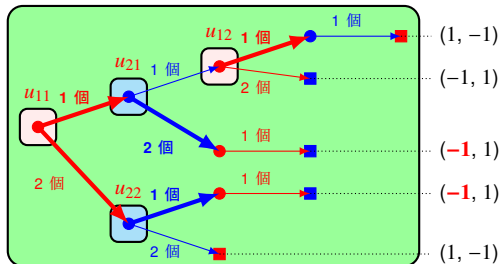
u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の例：石取りゲーム



後退帰納法の例：石取りゲーム

u_{12} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{21} : 1 個 ... 利得 -1, 2 個 ... 利得 1

u_{22} : 1 個 ... 利得 1, 2 個 ... 利得 -1

u_{11} : 1 個 ... 利得 -1, 2 個 ... 利得 -1

純粋戦略のナッシュ均衡 (プレイヤー 2 が必ず勝つ)

プレイヤー 1 : (u_{11} で 1 個, u_{12} で 1 個), (u_{11} で 2 個, u_{12} で 1 個)

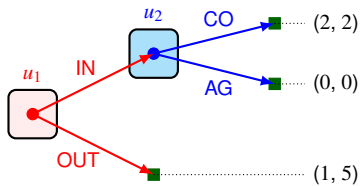
プレイヤー 2 : (u_{21} で 2 個, u_{22} で 1 個), (u_{21} で 2 個, u_{22} で 1 個)

後退帰納法の練習問題：チェーン店ゲーム

チェーン店ゲーム (chainstore game)

- A 市の市場は、大手チェーンストア (プレイヤー 2) の支店が独占
- 同じ商品を販売する別の事業者 (プレイヤー 1) は、A 市に開店する (IN) か、他の小規模な市に開店する (OUT) かを決定
- 他の市に開店する場合,
プレイヤー 1 : 利得 1
プレイヤー 2 : A 市の市場独占による利得 5
- プレイヤー 1 が A 市に開店する場合、プレイヤー 2 は、プレイヤー 1 と協調して価格を維持する (COOPERATIVE) か、値下げ競争を仕掛ける (AGGRESSIVE) かを決定
- 協調する場合,
プレイヤー 1 : 利得 2
プレイヤー 2 : 利得 2
- 競争する場合,
プレイヤー 1 : 利得 0
プレイヤー 2 : 利得 0

後退帰納法の練習問題：チェーン店ゲーム (続き)



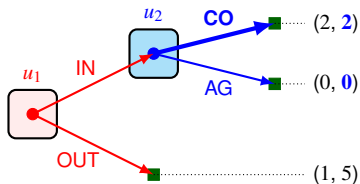
後退帰納法：チェーン店ゲーム

u_2 :

u_1 :

ナッシュ均衡は , そのときの利得は

後退帰納法の練習問題：チェーン店ゲーム (続き)



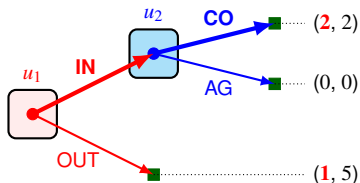
後退帰納法：チェーン店ゲーム

u_2 : CO... 利得 2, AG... 利得 0

u_1 :

ナッシュ均衡は , そのときの利得は

後退帰納法の練習問題：チェーン店ゲーム (続き)



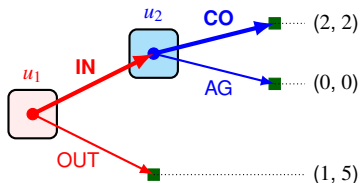
後退帰納法：チェーン店ゲーム

u_2 : **CO**... 利得 2, **AG**... 利得 0

u_1 : **IN**... 利得 2, **OUT**... 利得 1

ナッシュ均衡は , そのときの利得は

後退帰納法の練習問題：チェーン店ゲーム (続き)



後退帰納法：チェーン店ゲーム

u_2 : **CO**... 利得 2, **AG**... 利得 0

u_1 : **IN**... 利得 2, **OUT**... 利得 1

ナッシュ均衡は (**IN**, **CO**), そのときの利得は (2, 2)

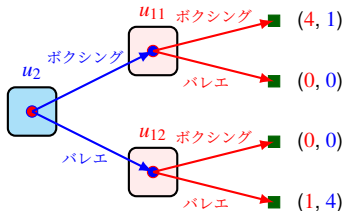
標準形ゲームへの変換

展開形ゲームの標準形ゲームへの変換

純粋戦略は各情報集合における行動の組合せ \Rightarrow すべて列挙

男女の争い（レディーファースト版）

- 同時手番ゲームではなく女 (プレイヤー 2) が先に選択する逐次手番ゲーム
- 他は男女の争いと同じ



利得行列

男 \ 女	ボ	バ
(ボ, ボ)	(4, 1)	(0, 0)
(ボ, バ)	(4, 1)	(0, 0)
(バ, ボ)	(0, 0)	(1, 4)
(バ, バ)	(0, 0)	(1, 4)

標準形ゲーム・展開形ゲームにおけるナッシュ均衡

チェーン店ゲームのナッシュ均衡

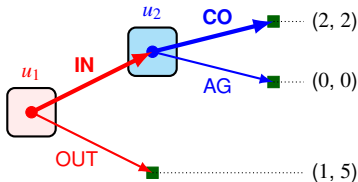
展開形ゲーム： (IN, CO)

標準形ゲーム： (IN, CO), (OUT, AG)

(OUT, AG) がナッシュ均衡であることの確認

- プレイヤー 1 が OUT \Rightarrow IN に変更：利得 $1 \Rightarrow 0$
- プレイヤー 2 が AG \Rightarrow CO に変更：利得 $5 \Rightarrow 5$

いずれも最適応答



利得行列

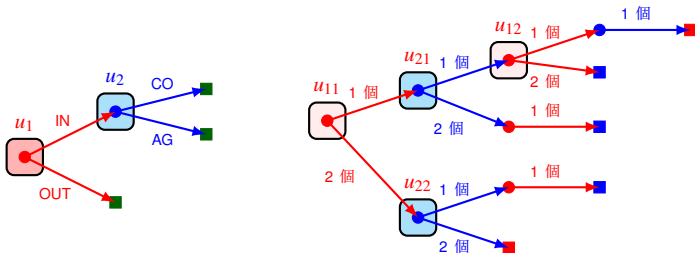
1 \ 2	CO AG	
	CO	AG
IN	(2, 2)	(0, 0)
OUT	(1, 5)	(1, 5)

- プレイヤー 1 の行動が OUT の場合、プレイヤー 2 の手番には到達しない
- 到達しない手番についても最適な行動を想定するのが望ましいが、(OUT, AG) ではそうになっていない \Rightarrow **より条件の厳しい均衡の定義が必要**

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



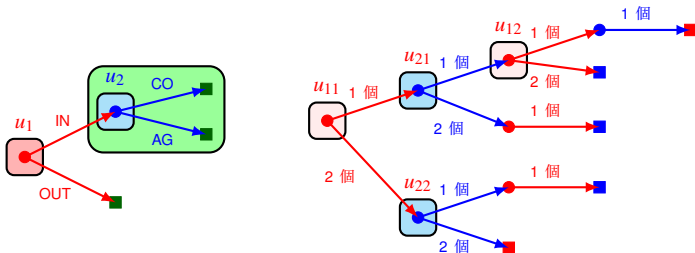
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



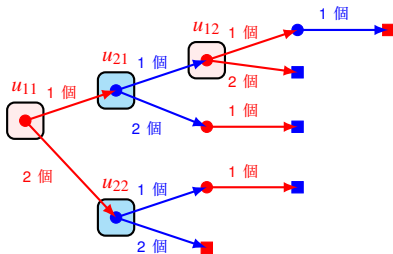
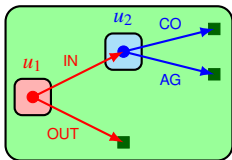
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- **ゲームの木全体も部分ゲームの一つ**
- 後退帰納法：部分ゲームにおけるナッシュ均衡



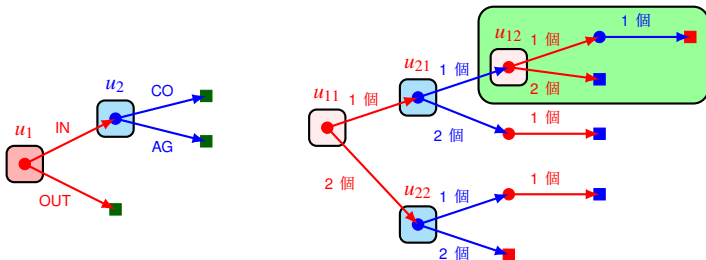
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



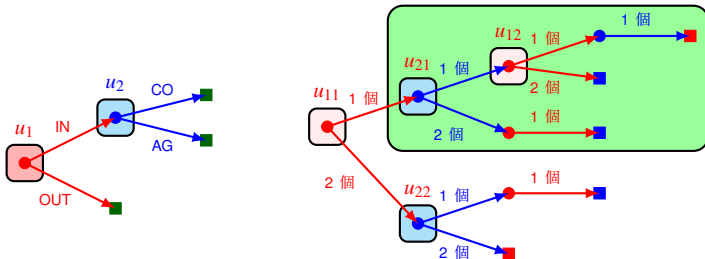
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



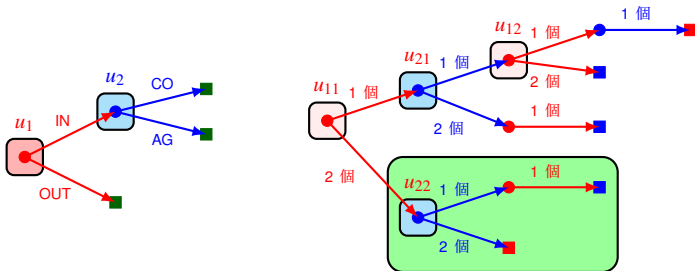
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



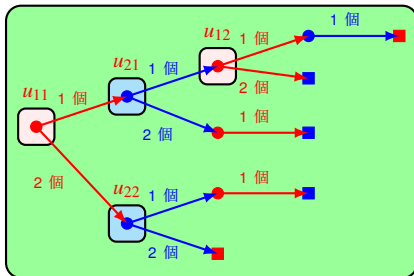
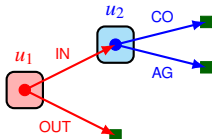
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- **ゲームの木全体も部分ゲームの一つ**
- 後退帰納法：部分ゲームにおけるナッシュ均衡



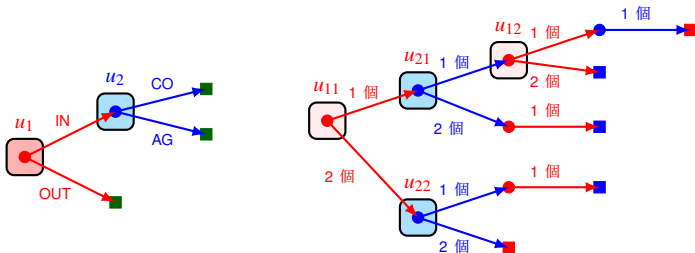
部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての妥当な (proper) 部分ゲームに対してナッシュ均衡

部分ゲーム完全均衡

部分ゲーム (subgame)

- ゲームの木の一部でゲームとして完結しているもの
- 情報集合を分割してはだめ
- ゲームの木全体も部分ゲームの一つ
- 後退帰納法：部分ゲームにおけるナッシュ均衡



部分ゲーム完全均衡 (subgame perfect equilibrium)

- 展開形ゲームに対するナッシュ均衡よりも強い条件
- すべての**妥当な** (proper) 部分ゲームに対してナッシュ均衡

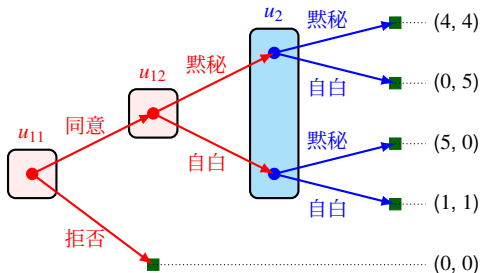
部分ゲーム完全均衡：妥当な部分ゲーム

妥当な部分ゲーム

部分ゲームのうち，開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ，そうでなければ終了



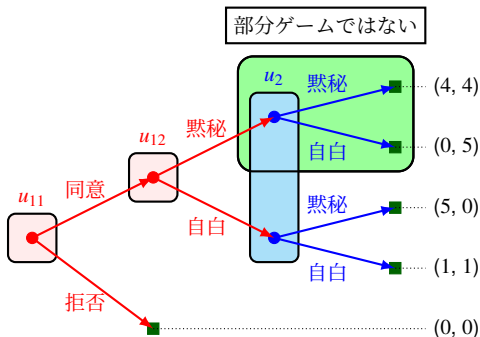
部分ゲーム完全均衡：妥当な部分ゲーム

妥当な部分ゲーム

部分ゲームのうち、開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ、そうでなければ終了



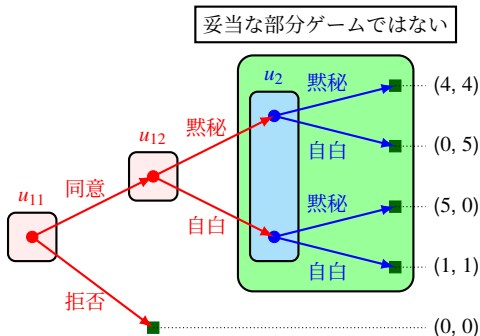
部分ゲーム完全均衡：妥当な部分ゲーム

妥当な部分ゲーム

部分ゲームのうち，開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ．そうでなければ終了



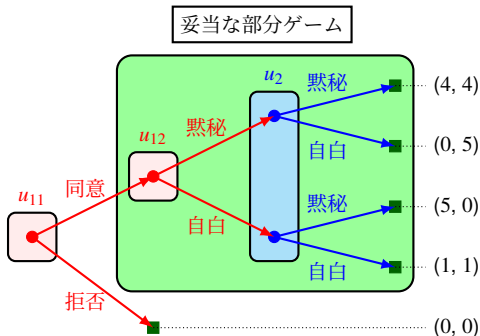
部分ゲーム完全均衡：妥当な部分ゲーム

妥当な部分ゲーム

部分ゲームのうち，開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ，そうでなければ終了



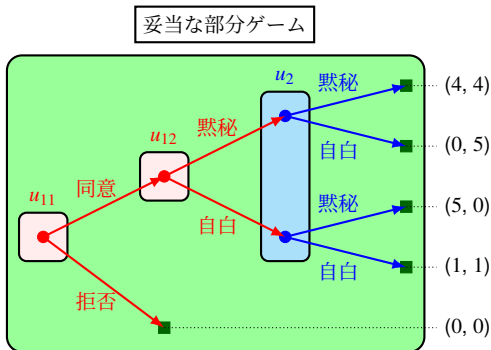
部分ゲーム完全均衡：妥当な部分ゲーム

妥当な部分ゲーム

部分ゲームのうち，開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ．そうでなければ終了



不完全情報ゲームの部分ゲーム完全均衡

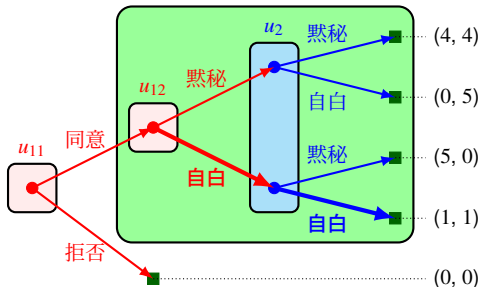
妥当な部分ゲーム

部分ゲームのうち，開始点の手番が 1 つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改 2

プレイヤー 1 が同意すれば囚人のジレンマをプレイ．そうでなければ終了

囚人のジレンマのナッシュ均衡は (自白, 自白)



不完全情報ゲームの部分ゲーム完全均衡

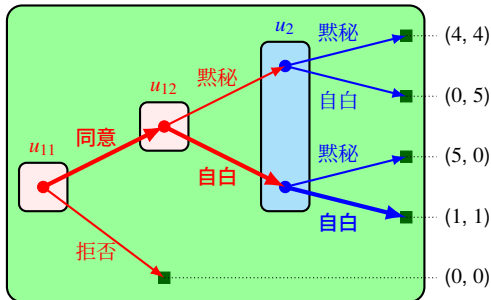
妥当な部分ゲーム

部分ゲームのうち、開始点の手番が1つだけのもの

妥当な部分ゲームの例：囚人のジレンマ改2

プレイヤー1が同意すれば囚人のジレンマをプレイ。そうでなければ終了

部分ゲーム完全均衡は (同意・自白, 自白)



練習問題：ボランティアのジレンマ

ボランティアのジレンマ (volunteer's dilemma)

- プレイヤーのいずれかがコスト 1 を負担すれば、全員がより大きな利得 3 を得る
コストを負担したプレイヤーの利得は $3 - 1 = 2$
- コストを負担する人数が増えても利得は変わらない
- 誰もコストを負担しなければ利得は 0

利得行列

C は協調 (cooperation) ・ 負担, D は裏切り (defection) ・ 負担しない

1\2	C ₂	D ₂
C ₁	(2, 2)	(2, 3)
D ₁	(3, 2)	(0, 0)

ナッシュ均衡

練習問題：ボランティアのジレンマ

ボランティアのジレンマ (volunteer's dilemma)

- プレイヤーのいずれかがコスト 1 を負担すれば、全員がより大きな利得 3 を得る
コストを負担したプレイヤーの利得は $3 - 1 = 2$
- コストを負担する人数が増えても利得は変わらない
- 誰もコストを負担しなければ利得は 0

利得行列

C は協調 (cooperation) ・ 負担, D は裏切り (defection) ・ 負担しない

1\2	C ₂	D ₂
C ₁	(2, 2)	(2, 3)
D ₁	(3, 2)	(0, 0)

ナッシュ均衡

- (D₁, D₂) : D₁ から C₁ に変更すると、利得は 0 から 2 に増加 ⇒ D₁ は最適応答ではない
- (C₁, D₂) : C₁ から D₁ に変更すると、利得は 2 から 0 に減少. D₂ から C₂ に変更すると、利得は 3 から 2 に減少. **ナッシュ均衡**
- (D₁, C₂) : (C₁, D₂) と同様. **ナッシュ均衡**
- (C₁, C₂) : C₁ から D₁ に変更すると、利得は 2 から 3 に増加. C₁ は最適応答ではない.

練習問題：ボランティアのジレンマ

ボランティアのジレンマ (volunteer's dilemma)

- プレイヤーのいずれかがコスト 1 を負担すれば、全員がより大きな利得 3 を得る
コストを負担したプレイヤーの利得は $3 - 1 = 2$
- コストを負担する人数が増えても利得は変わらない
- 誰もコストを負担しなければ利得は 0

利得行列

C は協調 (cooperation) ・ 負担, D は裏切り (defection) ・ 負担しない

1\2	C ₂	D ₂
C ₁	(2, 2)	(2, 3)
D ₁	(3, 2)	(0, 0)

ナッシュ均衡

(D₁, D₂) : D₁ から C₁ に変更すると、利得は 0 から 2 に増加 ⇒ D₁ は最適応答ではない

(C₁, D₂) : C₁ から D₁ に変更すると、利得は 2 から 0 に減少. D₂ から C₂ に変更すると、利得は 3 から 2 に減少. **ナッシュ均衡**

(D₁, C₂) : (C₁, D₂) と同様. **ナッシュ均衡**

(C₁, C₂) : C₁ から D₁ に変更すると、利得は 2 から 3 に増加. C₁ は最適応答ではない.

タダ乗りするのが得 (n 人の場合は $n - 1$ 人がタダ乗りするのがナッシュ均衡！)

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ．2 回目はコストが 1 から 2 に増加

利得行列 (1 回目)

1\2	C ₂₁	D ₂₁
C ₁₁	(2, 2)	(2, 3)
D ₁₁	(3, 2)	(0, 0)

利得行列 (2 回目)

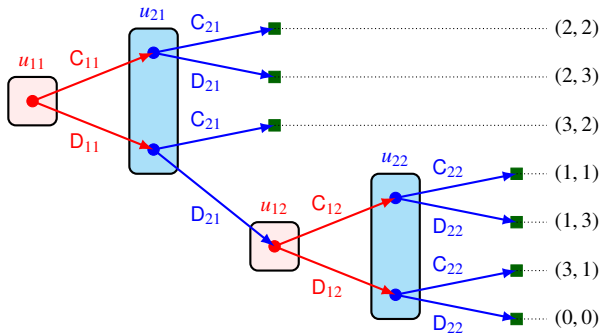
1\2	C ₂₂	D ₂₂
C ₁₂	(1, 1)	(1, 3)
D ₁₂	(3, 1)	(0, 0)

ゲームの木

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 に増加



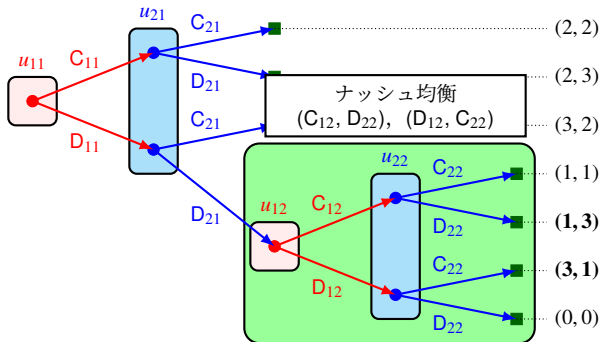
ゲームの木

部分ゲーム完全均衡

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 が増加



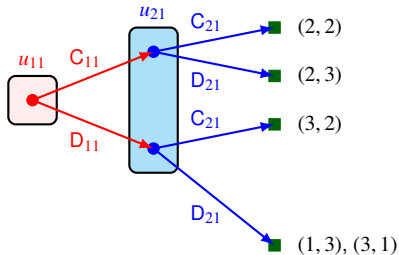
ゲームの木

部分ゲーム完全均衡

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 に増加



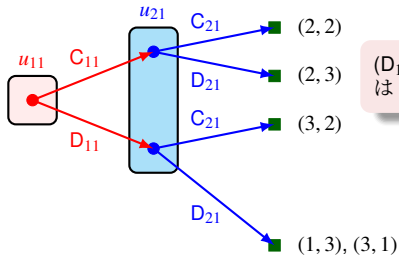
ゲームの木

部分ゲーム完全均衡

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 に増加



(D_{11}, D_{21}) は, (C_{11}, D_{21}) あるいは (D_{11}, C_{21}) に支配される

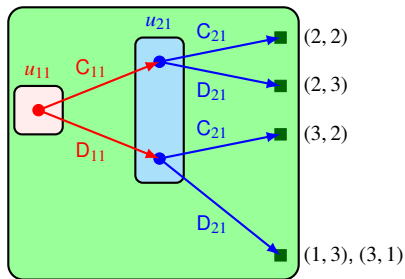
ゲームの木

部分ゲーム完全均衡

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 に増加



ナッシュ均衡
 $(C_{11}, D_{21}), (D_{11}, C_{21})$

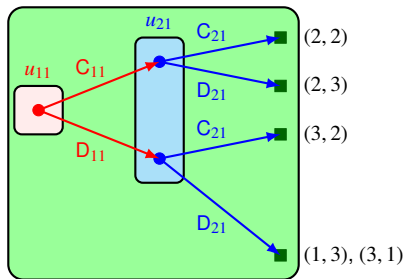
ゲームの木

部分ゲーム完全均衡

練習問題：ボランティアのジレンマ (続き)

ボランティアのジレンマ ×2

1 回目のプレイでいずれも D ならば 2 回目をプレイ. 2 回目はコストが 1 から 2 に増加



ナッシュ均衡
 $(C_{11}, D_{21}), (D_{11}, C_{21})$

ゲームの木

部分ゲーム完全均衡

$(C_{11}, D_{21}), (D_{11}, C_{21})$ と $(C_{12}, D_{22}), (D_{12}, C_{22})$ の組合せ (4 通り)

繰り返しゲーム

繰り返しゲーム (repeated game)

同一のゲームを繰り返しプレイ

割引因子 (discount factor)

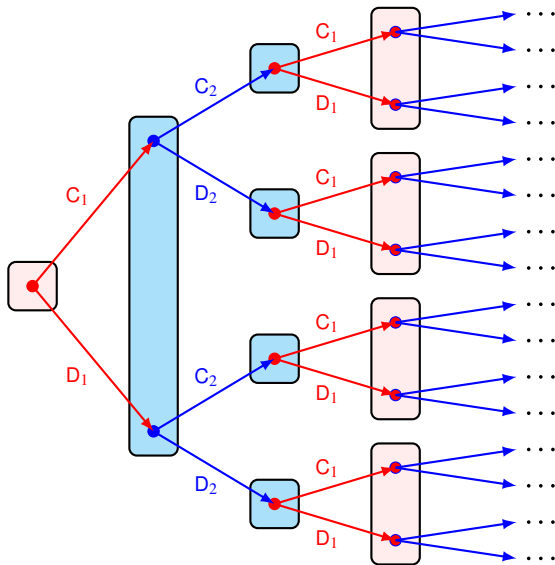
- 将来得られるであろう利得は、すぐに得られる利得より価値が低い
⇒ 利得を割り引いて考える
- 現在から t 回先に得られる利得は δ^t 倍
- δ ($0 < \delta < 1$) : **割引因子** (discount factor)

繰り返し囚人のジレンマ

以下の囚人のジレンマの一般形を無限回プレイ．黙秘は協力 (cooperation; C), 自白は裏切り (defection; D) とする．ただし, $P > Q > R > S$.

		利得行列	
1 \ 2		C ₂	D ₂
C ₁		(Q, Q)	(S, P)
D ₁		(P, S)	(R, R)

繰り返し囚人のジレンマの展開形



ゲームの木が無限に大きくなるので、別の方法で解析

繰り返し囚人のジレンマにおける戦略

繰り返し囚人のジレンマにおける戦略

以下の 4 通りを考える

常に協力する戦略 (all-C) : 常に協力

常に裏切る戦略 (all-D) : 常に裏切り

トリガー戦略 (trigger) : 初回は協力. 2 回目以降は, 相手が過去一度でも裏切っていたら裏切り, そうでなければ協力

しっぺ返し戦略 (tit-for-tat) : 初回は協力. 2 回目以降は相手の直前の行動を真似る

(all-C, all-C) における各プレイヤーの割引総利得 (1 回の利得 (Q, Q))

$$Q + \delta Q + \delta^2 Q + \cdots = \sum_{t=0}^{\infty} \delta^t Q = \frac{Q}{1 - \delta}$$

(all-C, trigger), (all-C, tit-for-tat), (trigger, trigger), (tit-for-tat) も同じ

繰り返し囚人のジレンマにおける各戦略の割引総利得

利得行列

1 \ 2	C ₂	D ₂
C ₁	(Q, Q)	(S, P)
D ₁	(P, S)	(R, R)

(all-D, trigger), (all-D, tit-for-tat) における各プレイヤーの割引総利得

1 回目は (P, S), 2 回目以降は (R, R) なので, プレイヤー 1 は

$$P + \frac{\delta R}{1 - \delta} = \frac{R}{1 - \delta} + (P - R)$$

プレイヤー 2 は

$$S + \frac{\delta R}{1 - \delta} = \frac{R}{1 - \delta} - (R - S)$$

各戦略に対する利得行列

1 \ 2	all-C	all-D	trigger	tit-for-tat
all-C	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{S}{1-\delta}, \frac{P}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$
all-D	$\left(\frac{P}{1-\delta}, \frac{S}{1-\delta}\right)$	$\left(\frac{R}{1-\delta}, \frac{R}{1-\delta}\right)$	$\left(P + \frac{\delta R}{1-\delta}, S + \frac{\delta R}{1-\delta}\right)$	$\left(P + \frac{\delta R}{1-\delta}, S + \frac{\delta R}{1-\delta}\right)$
trigger	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(S + \frac{\delta R}{1-\delta}, P + \frac{\delta R}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$
tit-for-tat	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(S + \frac{\delta R}{1-\delta}, P + \frac{\delta R}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$

繰り返し囚人のジレンマにおけるナッシュ均衡

各戦略に対する利得行列

1 \ 2	all-C	all-D	trigger	tit-for-tat
all-C	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{S}{1-\delta}, \frac{P}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$
all-D	$\left(\frac{P}{1-\delta}, \frac{S}{1-\delta}\right)$	$\left(\frac{R}{1-\delta}, \frac{R}{1-\delta}\right)$	$\left(P + \frac{\delta R}{1-\delta}, S + \frac{\delta R}{1-\delta}\right)$	$\left(P + \frac{\delta R}{1-\delta}, S + \frac{\delta R}{1-\delta}\right)$
trigger	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(S + \frac{\delta R}{1-\delta}, P + \frac{\delta R}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$
tit-for-tat	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(S + \frac{\delta R}{1-\delta}, P + \frac{\delta R}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$	$\left(\frac{Q}{1-\delta}, \frac{Q}{1-\delta}\right)$

ナッシュ均衡

- (all-D, all-D) (ワンショット囚人のジレンマと同様)
- (trigger, trigger), (trigger, tit-for-tat), (tit-for-tat, trigger), (tit-for-tat, tit-for-tat)

$$P + \frac{\delta R}{1-\delta} = (P - R) + \frac{R}{1-\delta} \leq \frac{Q}{1-\delta}$$

のときナッシュ均衡. 整理して, $\delta \geq \frac{P-Q}{P-R}$ のときナッシュ均衡

- 割引因子 δ が大きければ, プレイヤー間の協調が生まれる
- 割引因子 $\delta = 0$ のときは, 将来の利得を無視するため協調は生まれない
- **任意の戦略を許しても**, (δ が大きければ) (trigger, trigger) や (tit-for-tat, tit-for-tat) などはナッシュ均衡となることを示せる

アクセルロッドの実験

アクセルロッド (Robert Axelrod) の実験 (1980)

- 繰り返し囚人のジレンマに対するコンピュータプログラムの競技大会
- 反復回数 200 回
- 14 のプログラム + 行動をランダムに選択するプログラムの総当たり戦
- 優勝は**しっぺ返し戦略**, 最下位は (さすがに) ランダム
- しっぺ返し戦略のプログラム (FORTRAN) はたった 4 行!
- 一番長い 77 行のプログラムは 14 位
- 2 位もしっぺ返し戦略 (41 行) だが, 相手が 2 回目に裏切り始めると, 相手より 1 回多く裏切り返す戦略. そして, 3 回目は 2 回, 4 回目は 3 回と増やしていく. さらに, ある条件を満たすとこの回数をリセット.

フォーク定理

標準形ゲームにおけるミニマックス利得 (minmax payoff)

$$v_i = \min_{q_{-i} \in Q_{-i}} \max_{q_i \in Q_i} f(q_i, q_{-i})$$

プレイヤー i が (混合戦略も含めて) 最適行動を取るなら、最低でも v_i の利得が得られる

割引平均利得 (discounted average payoff)

割引総利得 V に対して $(1 - \delta)V$

毎回一定の利得 c が得られるとすると、割引総利得は $\frac{c}{1 - \delta} \cdot \frac{c}{1 - \delta} = V$ とおいて、
 $c = (1 - \delta)V$ が平均を表すと考え

フォーク定理 (folk theorem)

無限回繰り返しゲームにおいて、割引因子 δ を十分大きくとれば、各プレイヤー i の **割引平均利得** が **ミニマックス利得** v_i より大きいナッシュ均衡を (実行可能な範囲で) 達成可能

「folk」は民間伝承 (folklore) から来ている。研究者の間では知られた定理だったが、誰も証明を出版しようとしなかったため

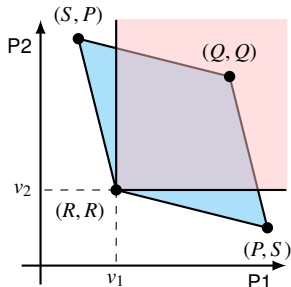
繰り返し囚人のジレンマに対するフォーク定理

利得行列

$1 \setminus 2$	C_2	D_2
C_1	(Q, Q)	(S, P)
D_1	(P, S)	(R, R)

ミニマックス利得 $v_1 = v_2$

- プレイヤー 1 の D_1 は C_1 を支配 \Rightarrow プレイヤー 2 の行動にかかわらず最適応答は D_1
- プレイヤー 1 の利得を最小化するプレイヤー 2 の行動は D_2
- ミニマックス利得 $v_1 = v_2 = R$



- 実行可能な範囲は純粋戦略の組 (Q, Q) , (S, P) , (R, R) , (P, S) で囲まれた領域
- $x > v_1$ かつ $y > v_2$ と重なる領域が, ナッシュ均衡で達成可能な割引平均利得
- (trigger, trigger) の割引平均利得

$$\left((1 - \delta) \frac{Q}{1 - \delta}, (1 - \delta) \frac{Q}{1 - \delta} \right) = (Q, Q)$$