CS 886: Multiagent Systems

Extensive Form Games

Kate Larson

Computer Science
University of Waterloo

Outline

Extensive Form Games

aka Dynamic Games, aka Tree-Form Games

- Extensive form games allows us to model situations where agents take actions over time
- Simplest type is the perfect information game

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \to 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \to N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \to H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

- $u = (u_1, \ldots, u_n)$ where $u_i : Z \to \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \to 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \to N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \to H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

- $u = (u_1, \ldots, u_n)$ where $u_i : Z \to \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \rightarrow 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \rightarrow N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \rightarrow H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

- $u = (u_1, \ldots, u_n)$ where $u_i : Z \rightarrow \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \rightarrow 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \rightarrow N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \rightarrow H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$
- $u = (u_1, \ldots, u_n)$ where $u_i : Z \rightarrow \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \rightarrow 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \rightarrow N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \rightarrow H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

$$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

- $u = (u_1, \ldots, u_n)$ where $u_i : Z \rightarrow \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \to 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \to N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \to H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

- $u = (u_1, \ldots, u_n)$ where $u_i : Z \to \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \rightarrow 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \rightarrow N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \rightarrow H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$
- $u = (u_1, \ldots, u_n)$ where $u_i : Z \rightarrow \mathbb{R}$ is utility function for player i over Z

Perfect Information Game

Perfect Information Game: $G = (N, A, H, Z, \alpha, \rho, \sigma, u)$

- N is the player set $|N| = n$
- $A = A_1 \times \ldots \times A_n$ is the action space
- H is the set of non-terminal choice nodes
- Z is the set of terminal nodes
- $\alpha : H \rightarrow 2^A$ action function, assigns to a choice node a set of possible actions
- $\rho : H \rightarrow N$ player function, assigns a player to each non-terminal node (player who gets to take an action)
- $\sigma : H \times A \rightarrow H \cup Z$, successor function that maps choice nodes and an action to a new choice node or terminal node where

  $$\forall h_1, h_2 \in H \text{ and } a_1, a_2 \in A \text{ if } h_1 \neq h_2 \text{ then } \sigma(h_1, a_1) \neq \sigma(h_2, a_2)$$

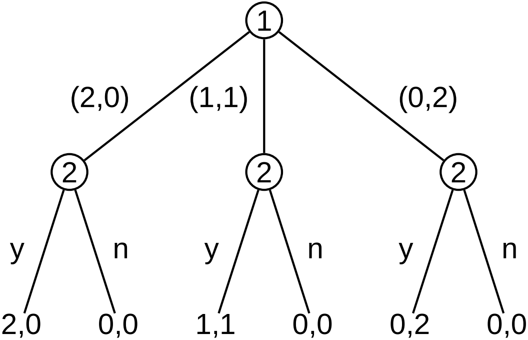
- $u = (u_1, \ldots, u_n)$ where $u_i : Z \rightarrow \mathbb{R}$ is utility function for player i over Z

Tree Representation

- The definition is really a tree description
- Each node is defined by its history (sequence of nodes leading from root to it)
- The descendents of a node are all choice and terminal nodes in the subtree rooted at the node.

Example

Sharing two items



Strategies

- A strategy, s_i of player i is a function that assigns an action to each non-terminal history, at which the agent can move.
- Outcome: $o(s)$ of strategy profile s is the terminal history that results when agents play s
- **Important:** The strategy definition requires a decision at each choice node, regardless of whether or not it is possible to reach that node given earlier moves

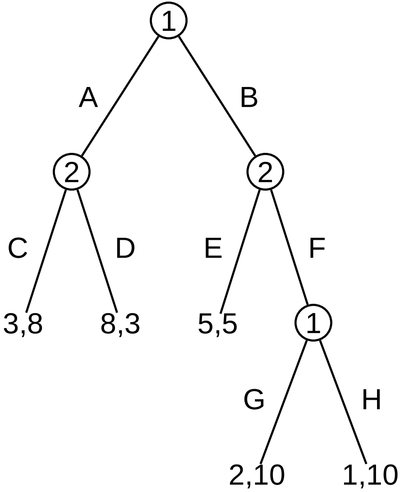
Strategies

- A strategy, s_i of player i is a function that assigns an action to each non-terminal history, at which the agent can move.
- Outcome: $o(s)$ of strategy profile s is the terminal history that results when agents play s
- **Important:** The strategy definition requires a decision at each choice node, regardless of whether or not it is possible to reach that node given earlier moves

Strategies

- A strategy, s_i of player i is a function that assigns an action to each non-terminal history, at which the agent can move.
- Outcome: $o(s)$ of strategy profile s is the terminal history that results when agents play s
- **Important:** The strategy definition requires a decision at each choice node, regardless of whether or not it is possible to reach that node given earlier moves

Example

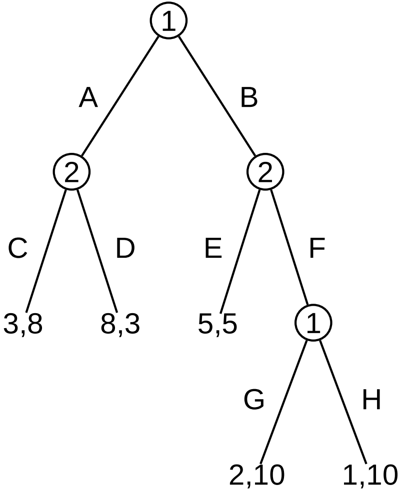


Strategy sets for the agents

$$S_1 = \{(A, G), (A, H), (B, G), (B, H)\}$$

$$S_2 = \{(C, E), (C, F), (D, E), (D, F)\}$$

Example



Strategy sets for the agents

$$S_1 = \{(A,G),(A,H),(B,G),(B,H)\}$$

$$S_2 = \{(C,E),(C,F),(D,E),(D,F)\}$$

Example

We can transform an extensive form game into a normal form game.

	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3,8	3,,8	8,3	8,3
(A,H)	3,8	3,,8	8,3	8,3
(B,G)	5,5	2,10	5,5	2, 10
(B,H)	5,5	1,0	5,5	1,0

Nash Equilibria

Definition (Nash Equilibrium)

Strategy profile s^* is a Nash Equilibrium in a perfect information, extensive form game if for all i

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*) \forall s_i'$$

Theorem

Any perfect information game in extensive form has a pure strategy Nash equilibrium.

Intuition: Since players take turns, and everyone sees each move there is no reason to randomize.

Nash Equilibria

Definition (Nash Equilibrium)

Strategy profile s^* is a Nash Equilibrium in a perfect information, extensive form game if for all i

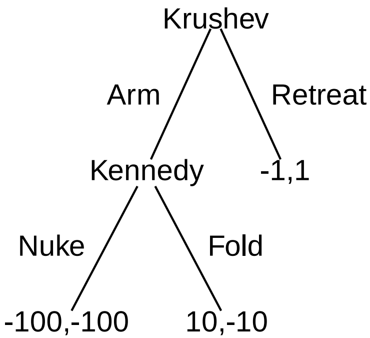
$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*) \forall s_i'$$

Theorem

Any perfect information game in extensive form has a pure strategy Nash equilibrium.

Intuition: Since players take turns, and everyone sees each move there is no reason to randomize.

Example: Bay of Pigs



What are the NE?

Subgame Perfect Equilibrium

Nash Equilibrium can sometimes be too weak a solution concept.

Definition (Subgame)

Given a game G, the subgame of G rooted at node j is the restriction of G to its descendents of h.

Definition (Subgame perfect equilibrium)

A strategy profile s^* is a subgame perfect equilibrium if for all $i \in N$, and for all subgames of G, the restriction of s^* to G' (G' is a subgame of G) is a Nash equilibrium in G'. That is

$$\forall i, \forall G', u_i(s_i^*|_{G'}, s_{-i}^*|_{G'}) \geq u_i(s_i'|_{G'}, s_{-i}^*|_{G'}) \forall s_i'$$

Subgame Perfect Equilibrium

Nash Equilibrium can sometimes be too weak a solution concept.

Definition (Subgame)

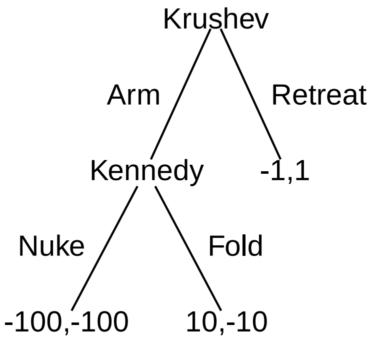
Given a game G, the subgame of G rooted at node j is the restriction of G to its descendents of h.

Definition (Subgame perfect equilibrium)

A strategy profile s^* is a subgame perfect equilibrium if for all $i \in N$, and for all subgames of G, the restriction of s^* to G' (G' is a subgame of G) is a Nash equilibrium in G'. That is

$$\forall i, \forall G', u_i(s_i^*|_{G'}, s_{-i}^*|_{G'}) \geq u_i(s_i'|_{G'}, s_{-i}^*|_{G'}) \forall s_i'$$

Example: Bay of Pigs



What are the SPE?

Existence of SPE

Theorem (Kuhn's Thm)

Every finite extensive form game with perfect information has a SPE.

You can find the SPE by backward induction.

- Identify equilibria in the bottom-most trees
- Work upwards

Existence of SPE

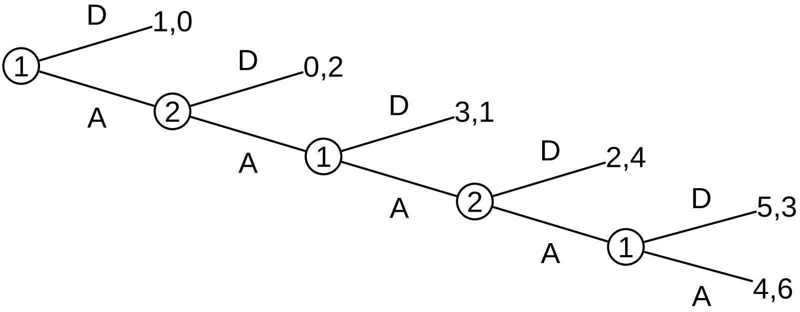
Theorem (Kuhn's Thm)

Every finite extensive form game with perfect information has a SPE.

You can find the SPE by backward induction.

- Identify equilibria in the bottom-most trees
- Work upwards

Centipede Game



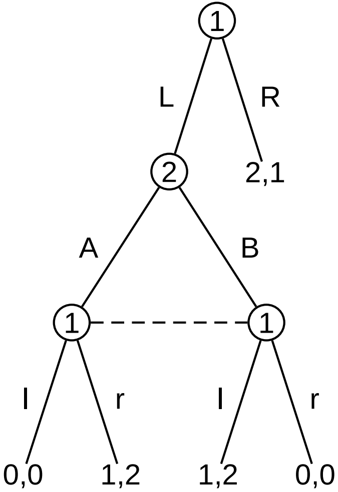
Imperfect Information Games

- Sometimes agents have not observed everything, or else can not remember what they have observed

Imperfect information games: Choice nodes H are partitioned into *information sets*.

- If two choice nodes are in the same information set, then the agent can not distinguish between them.
- Actions available to an agent must be the same for all nodes in the same information set

Example

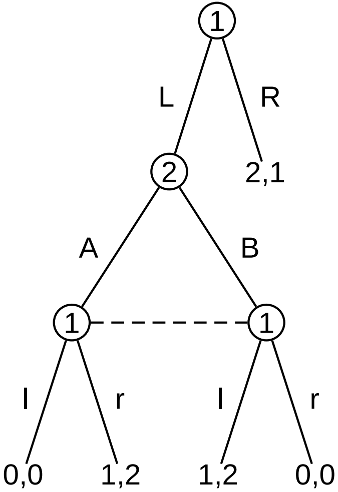


Information sets for agent 1

$$I_1 = \{\{\emptyset\}, \{(L, A), (L, B)\}\}$$

$$I_2 = \{\{L\}\}$$

Example

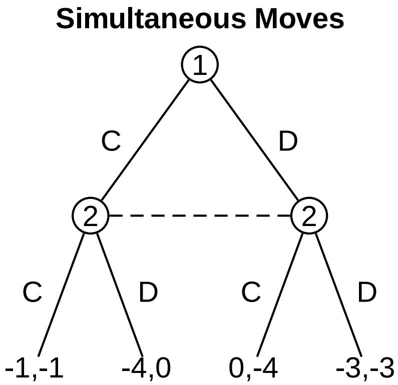


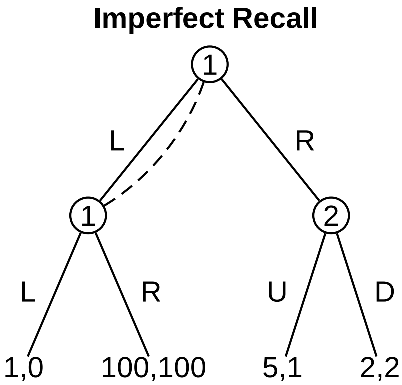
Information sets for agent 1

$$I_1 = \{\{\emptyset\}, \{(L, A), (L, B)\}\}$$

$$I_2 = \{\{L\}\}$$

More Examples





Strategies

- **Pure strategy:** a function that assigns an action in $A_i(I_i)$ to each information set $I_i \in \mathcal{I}_i$
- **Mixed strategy:** probability distribution over pure strategies
- **Behavorial strategy:** probability distribution over actions available to agent i at each of its information sets (independent distributions)

Strategies

- **Pure strategy:** a function that assigns an action in $A_i(I_i)$ to each information set $I_i \in \mathcal{I}_i$
- **Mixed strategy:** probability distribution over pure strategies
- **Behavorial strategy:** probability distribution over actions available to agent i at each of its information sets (independent distributions)

Strategies

- **Pure strategy:** a function that assigns an action in $A_i(I_i)$ to each information set $I_i \in \mathcal{I}_i$
- **Mixed strategy:** probability distribution over pure strategies
- **Behavorial strategy:** probability distribution over actions available to agent i at each of its information sets (independent distributions)

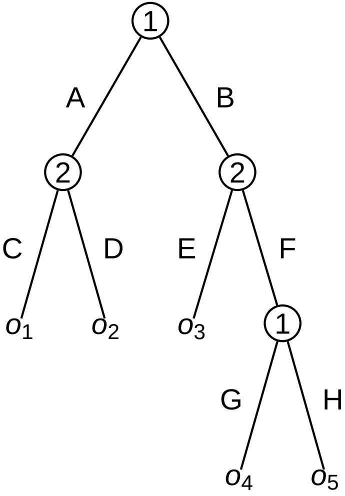
Behavioral Strategies

Definition

Given extensive game G, a behavioral strategy for player i specifies, for every $I_i \in \mathcal{I}_i$ *and action* $a_i \in A_i(I_i)$*, a probability* $\lambda_i(a_i, I_i) \geq 0$ *with*

$$\sum_{a_i \in A_i(I_i)} \lambda(a_i, I_i) = 1$$

Example



Mixed Strategy:
(0.4(A,G), 0.6(B,H))

Behavorial Strategy:

- Play A with probability 0.5
- Play G with probability 0.3

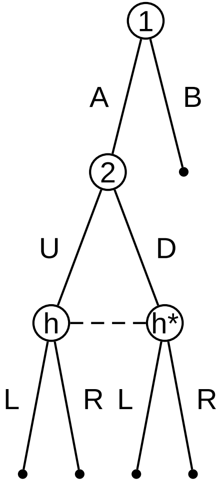
Mixed and Behavorial Strategies

In general you can not compare the two types of strategies.

But for games with perfect recall

- Any mixed strategy can be replaced with a behavorial strategy
- Any behavorial strategy can be replaced with a mixed strategy

Example



Mixed Strategy:
(<0.3(A,L)>,<0.2(A,R)>,
<0.5(B,L)>)

Behavorial Strategy:

- At I_1: (0.5, 0.5)
- At I_2: (0.6, 0.4)

Outline

Bayesian Games

So far we have assumed that all players know what game they are playing

- Number of players
- Actions available to each player
- Payoffs associated with strategy profiles

	L	R
U	3,?	-2, ?
D	0, ?	6, ?

Bayesian games (games of incomplete information) are used to represent uncertainties about the game being played

Bayesian Games

So far we have assumed that all players know what game they are playing

- Number of players
- Actions available to each player
- Payoffs associated with strategy profiles

	L	R
U	3,?	-2, ?
D	0, ?	6, ?

Bayesian games (games of incomplete information) are used to represent uncertainties about the game being played

Bayesian Games

There are different possible representations.

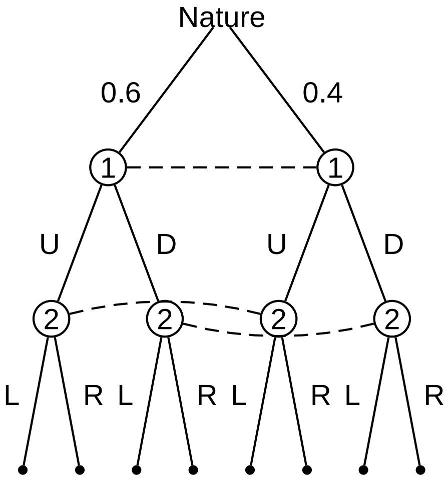
Information Sets

- N set of agents
- G set of games
  - Same strategy sets for each game and agent
- $\Pi(G)$ is the set of all probability distributions over G
  - $P(G) \in \Pi(G)$ common prior
- $I = (I_1, \ldots, I_n)$ are information sets (partitions over games)

Example

Extensive Form With Chance Moves

A special player, Nature, makes probabilistic moves.



Epistemic Types

Epistemic types captures uncertainty directly over a game's utility functions.

- N set of agents
- $A = (A_1, \ldots, A_n)$ actions for each agent
- $\Theta = \Theta_1 \times \ldots \times \Theta_n$ where Θ_i is *type space* of each agent
- $p : \Theta \rightarrow [0, 1]$ is common prior over types
- Each agent has utility function $u_i : A \times \Theta \rightarrow \mathbb{R}$

Example

BoS

- 2 agents
- $A_1 = A_2 = \{\text{soccer, hockey}\}$
- $\Theta = (\Theta_1, \Theta_2)$ where $\Theta_1 = \{\text{H, S}\}$, $\Theta_2 = \{\text{H, S}\}$
- Prior: $p_1(H) = 1$, $p_2(H) = \frac{2}{3}$, $p_2(S) = \frac{1}{3}$

Utilities can be captured by matrix-form

$\theta_2 = H$

	H	S
H	2,2	0,0
S	0,0	1,1

$\theta_2 = S$

	H	S
H	2,1	0,0
S	0,0	1,2

Strategies and Utility

- A strategy $s_i(\theta_i)$ is a mapping from Θ_i to A_i. It specifies what action (or what distribution of actions) to take for each type.

Utility: $u_i(s|\theta_i)$

- *ex-ante* EU (know nothing about types)

$$EU = \sum_{\theta_i \in \Theta_i} p(\theta_i) EU_i(s_i|\theta_i)$$

- *interim* EU (know own type)

$$EU = EU_i(s|\theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i}|\theta_i) \sum_{a \in A} \Pi_{j \in N} s_j(a_j, \theta_j)) u_i(a, \theta_{-i}, \theta_i)$$

- *ex-post* EU (know everyones type)

Strategies and Utility

- A strategy $s_i(\theta_i)$ is a mapping from Θ_i to A_i. It specifies what action (or what distribution of actions) to take for each type.

Utility: $u_i(s|\theta_i)$

- *ex-ante* EU (know nothing about types)

$$EU = \sum_{\theta_i \in \Theta_i} p(\theta_i) EU_i(s_i|\theta_i)$$

- *interim* EU (know own type)

$$EU = EU_i(s|\theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i}|\theta_i) \sum_{a \in A} \Pi_{j \in N} s_j(a_j, \theta_j)) u_i(a, \theta_{-i}, \theta_i)$$

- *ex-post* EU (know everyones type)

Strategies and Utility

- A strategy $s_i(\theta_i)$ is a mapping from Θ_i to A_i. It specifies what action (or what distribution of actions) to take for each type.

Utility: $u_i(s|\theta_i)$

- *ex-ante* EU (know nothing about types)

$$EU = \sum_{\theta_i \in \Theta_i} p(\theta_i) EU_i(s_i|\theta_i)$$

- *interim* EU (know own type)

$$EU = EU_i(s|\theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i}|\theta_i) \sum_{a \in A} \Pi_{j \in N} s_j(a_j, \theta_j)) u_i(a, \theta_{-i}, \theta_i)$$

- *ex-post* EU (know everyones type)

Strategies and Utility

- A strategy $s_i(\theta_i)$ is a mapping from Θ_i to A_i. It specifies what action (or what distribution of actions) to take for each type.

Utility: $u_i(s|\theta_i)$

- *ex-ante* EU (know nothing about types)

$$EU = \sum_{\theta_i \in \Theta_i} p(\theta_i) EU_i(s_i|\theta_i)$$

- *interim* EU (know own type)

$$EU = EU_i(s|\theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i}|\theta_i) \sum_{a \in A} \Pi_{j \in N} s_j(a_j, \theta_j)) u_i(a, \theta_{-i}, \theta_i)$$

- *ex-post* EU (know everyones type)

Example

- 2 firms, 1 and 2, competing to create some product.
- If one makes the product then it has to share with the other.
- Product development cost is $c \in (0, 1)$
- Benefit of having the product is known only to each firm
  - Type θ_i drawn uniformly from $[0, 1]$
  - Benefit of having product is θ_i^2

Bayes Nash Equilibrium

Definition (BNE)

Strategy profile s^* *is a Bayes Nash equilibrium if* $\forall i, \forall \theta_i$

$$EU(s_i^*, s_{-i}^* | \theta_i) \geq EU(s_i', s_{-i}^* | \theta_i) \forall s_i' \neq s_i^*$$

Example Continued

- Let $s_i(\theta_i) = 1$ if i develops product, and 0 otherwise.
- If i develops product

$$u_i = \theta_i^2 - c$$

If it does not then

$$u_i = \theta_i^2 Pr(s_j(\theta_j) = 1)$$

- Thus, develop product if and only if

$$\theta_i^2 - c \geq \theta_i^2 Pr(s_j(\theta_j) = 1) \Rightarrow \theta_i \geq \sqrt{\frac{c}{1 - Pr(s_j(\theta_j) = 1)}}$$

Example Continued

- Let $s_i(\theta_i) = 1$ if i develops product, and 0 otherwise.
- If i develops product

$$u_i = \theta_i^2 - c$$

If it does not then

$$u_i = \theta_i^2 Pr(s_j(\theta_j) = 1)$$

- Thus, develop product if and only if

$$\theta_i^2 - c \geq \theta_i^2 Pr(s_j(\theta_j) = 1) \Rightarrow \theta_i \geq \sqrt{\frac{c}{1 - Pr(s_j(\theta_j) = 1)}}$$

Example Continued

- Let $s_i(\theta_i) = 1$ if i develops product, and 0 otherwise.
- If i develops product

$$u_i = \theta_i^2 - c$$

If it does not then

$$u_i = \theta_i^2 Pr(s_j(\theta_j) = 1)$$

- Thus, develop product if and only if

$$\theta_i^2 - c \geq \theta_i^2 Pr(s_j(\theta_j) = 1) \Rightarrow \theta_i \geq \sqrt{\frac{c}{1 - Pr(s_j(\theta_j) = 1)}}$$

Example Continued

Suppose $\hat{\theta}_1$, $\hat{\theta}_2 \in (0, 1)$ are cutoff values in BNE.

- If so, then $Pr(s_j(\theta_j) = 1) = 1 - \hat{\theta}_j$
- We must have

$$\hat{\theta}_i \geq \sqrt{\frac{c}{\hat{\theta}_j}} \Rightarrow \hat{\theta}_i^2 \hat{\theta}_j = c$$

and

$$\hat{\theta}_j^2 \hat{\theta}_i = c$$

- Therefore

$$\hat{\theta}_i^2 \hat{\theta}_j = \hat{\theta}_j^2 \hat{\theta}_i$$

and so

$$\hat{\theta}_i = \hat{\theta}_j = \theta^* = c^{\frac{1}{3}}$$

Example Continued

Suppose $\hat{\theta}_1, \hat{\theta}_2 \in (0,1)$ are cutoff values in BNE.

- If so, then $Pr(s_j(\theta_j) = 1) = 1 - \hat{\theta}_j$
- We must have

$$\hat{\theta}_i \geq \sqrt{\frac{c}{\hat{\theta}_j}} \Rightarrow \hat{\theta}_i^2 \hat{\theta}_j = c$$

  and

$$\hat{\theta}_j^2 \hat{\theta}_i = c$$

- Therefore

$$\hat{\theta}_i^2 \hat{\theta}_j = \hat{\theta}_j^2 \hat{\theta}_i$$

  and so

$$\hat{\theta}_i = \hat{\theta}_j = \theta^* = c^{\frac{1}{3}}$$

Example Continued

Suppose $\hat{\theta}_1$, $\hat{\theta}_2 \in (0,1)$ are cutoff values in BNE.

- If so, then $Pr(s_j(\theta_j) = 1) = 1 - \hat{\theta}_j$
- We must have

$$\hat{\theta}_i \geq \sqrt{\frac{c}{\hat{\theta}_j}} \Rightarrow \hat{\theta}_i^2 \hat{\theta}_j = c$$

and

$$\hat{\theta}_j^2 \hat{\theta}_i = c$$

- Therefore

$$\hat{\theta}_i^2 \hat{\theta}_j = \hat{\theta}_j^2 \hat{\theta}_i$$

and so

$$\hat{\theta}_i = \hat{\theta}_j = \theta^* = c^{\frac{1}{3}}$$

Example Continued

Suppose $\hat{\theta}_1$, $\hat{\theta}_2 \in (0, 1)$ are cutoff values in BNE.

- If so, then $Pr(s_j(\theta_j) = 1) = 1 - \hat{\theta}_j$
- We must have

$$\hat{\theta}_i \geq \sqrt{\frac{c}{\hat{\theta}_j}} \Rightarrow \hat{\theta}_i^2 \hat{\theta}_j = c$$

  and

$$\hat{\theta}_j^2 \hat{\theta}_i = c$$

- Therefore

$$\hat{\theta}_i^2 \hat{\theta}_j = \hat{\theta}_j^2 \hat{\theta}_i$$

  and so

$$\hat{\theta}_i = \hat{\theta}_j = \theta^* = c^{\frac{1}{3}}$$