Extensive Form Games

Game Theory

MohammadAmin Fazli

TOC

- Perfect Information Extensive Form Games
- Backward Induction and MinMax Algorithms
- Imperfect Information Extensive Form Games
- The Sequence Form

- Reading:
  - Chapter 5 of the MAS book

Extensive Form Games

- The normal form game representation does not incorporate any notion of sequence, or time, of the actions of the player
- The extensive form is an alternative representation that makes the temporal structure explicit.
- Two variants:
  - perfect information extensive-form games
  - imperfect-information extensive-form games

Perfect-Information Games

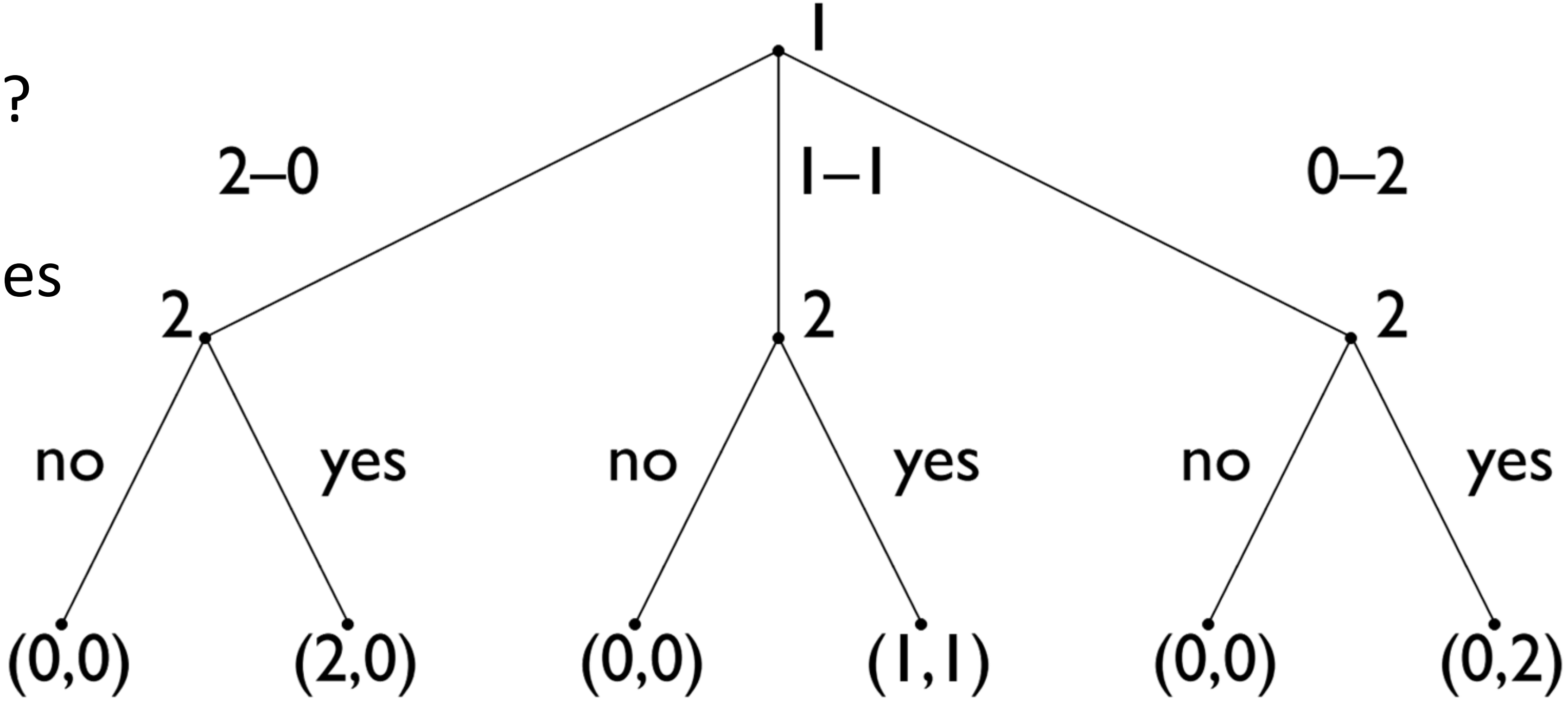
- A (finite) perfect-information game (in extensive form) is defined by the tuple (*N, A, H, Z, χ, ρ, σ, u*), where:
  - Players: *N* is a set of *n* players
  - Actions: *A* is a (single) set of actions
  - Choice nodes and labels for these nodes:
    - Choice nodes: *H* is a set of non-terminal choice nodes
    - Action function: $\chi: H \rightarrow 2^A$ assigns to each choice node a set of possible actions
    - Player function: $\rho: H \rightarrow N$ assigns to each non-terminal node *h* a player $i \in N$ who chooses an action at *h*

Perfect-Information Games

- A (finite) perfect-information game (in extensive form) is defined by the tuple (*N, A, H, Z, χ, ρ, σ, u*), where:
  - Terminal nodes: *Z* is a set of terminal nodes, disjoint from *H*
  - Successor function: $\sigma: H \times A \rightarrow H \cup Z$ maps a choice node and an action to a new choice node or terminal node such that for all $h_1, h_2 \in H$ and $a_1, a_2 \in A$, if $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ then $h_1 = h_2$ and $a_1 = a_2$
    - Choice nodes form a tree: nodes encode history
  - Utility function: $u = (u_1, u_2, \ldots, u_n)$, $u_i: Z \rightarrow R$ is a utility function for player *i* on the terminal nodes *Z*

Example

- What are the sharing game's formal definition elements?
- How many pure strategies player does each player has?
  - Player 1: 3
  - Player 2: 8



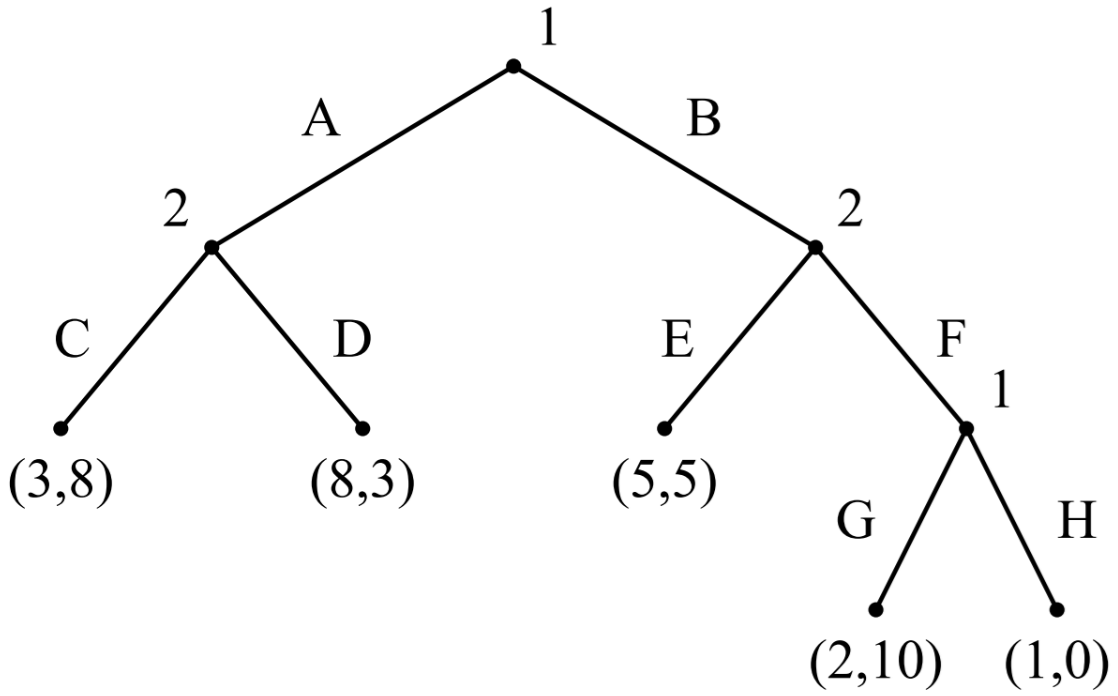
Pure Strategies

- A pure strategy for a player in a perfect-information game is a complete specification of which action to take at each node belonging to that player.
- **Pure Strategies:** Let $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ be a perfect-information extensive-form game. Then the pure strategies of player *i* consist of the cross product

$$\prod_{h \in H, \rho(h)=i} \chi(h)$$

Pure Strategies Example

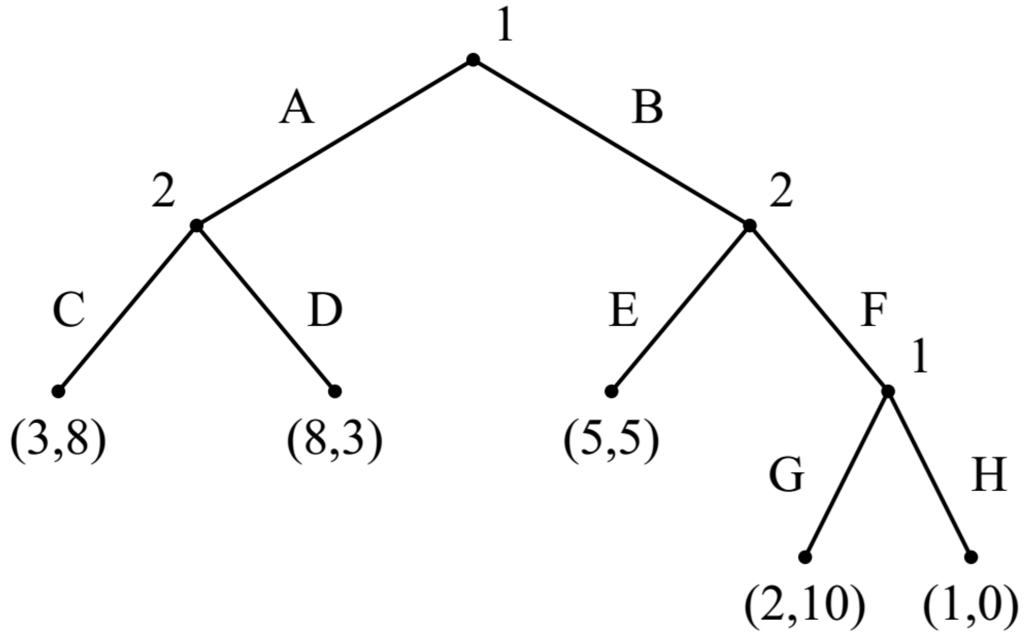
- Pure strategies for player 2:
  - $S_2 = \{(C, E), (C, F), (D, E), (D, F)\}$
- Pure strategies for player 1:
  - $S_1 = \{(B, G), (B, H), (A, G), (A, H)\}$



Nash Equilibria

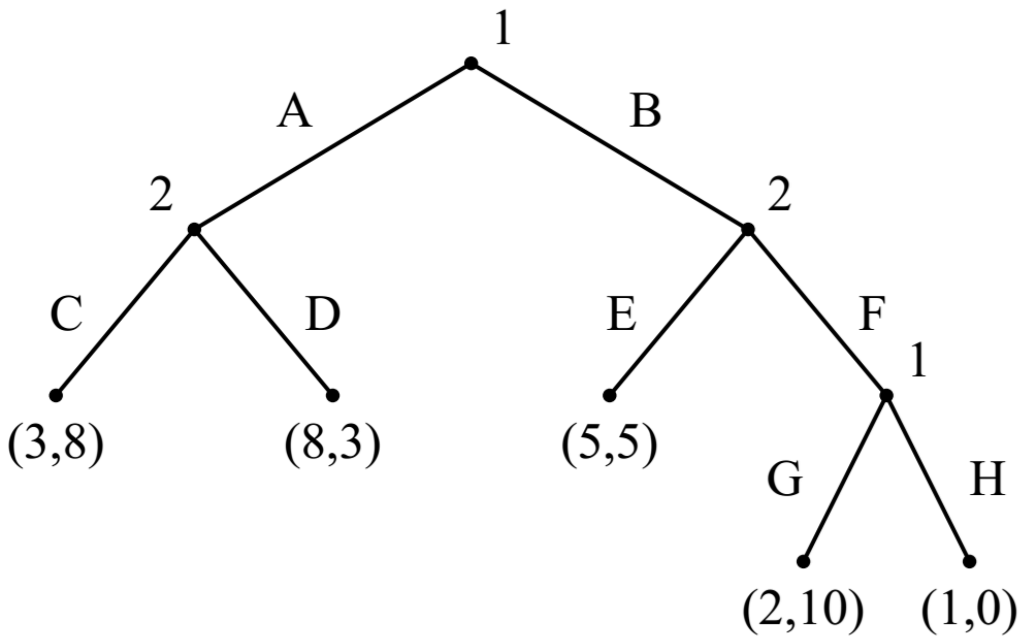
- Given our new definition of pure strategy, we are able to reuse our old definitions of:
  - Mixed strategies
  - Best response
  - Nash equilibrium

	CE	CF	DE	DF
AG	$3,8$	$3,8$	$8,3$	$8,3$
AH	$3,8$	$3,8$	$8,3$	$8,3$
BG	$5,5$	$2,10$	$5,5$	$2,10$
BH	$5,5$	$1,0$	$5,5$	$1,0$



Nash Equilibria

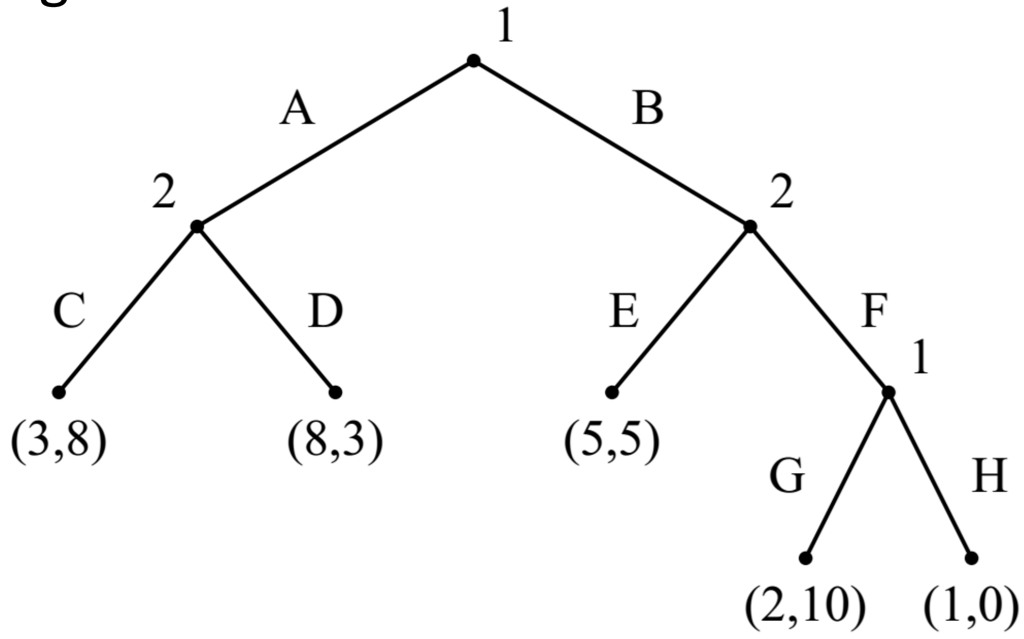
	CE	CF	DE	DF
AG	$3,8$	$3,8$	$8,3$	$8,3$
AH	$3,8$	$3,8$	$8,3$	$8,3$
BG	$5,5$	$2,10$	$5,5$	$2,10$
BH	$5,5$	$1,0$	$5,5$	$1,0$



- Theorem: Every perfect information game in extensive form has a PSNE.
  - Proof: This is easy to see, since the players move sequentially.
  - We will see the constructive proof by backward induction.
- Pure-strategy equilibria:
  - (*A, G*), (*C, F*)
  - (*A, H*), (*C, F*)
  - (*B, H*), (*C, E*)

Subgame Perfect Equilibrium

- There's something intuitively wrong with the equilibrium (*B, H*),(*C, E*)
  - Why would player 1 ever choose to play H if he got to the second choice node?
    - After all, G dominates H for him

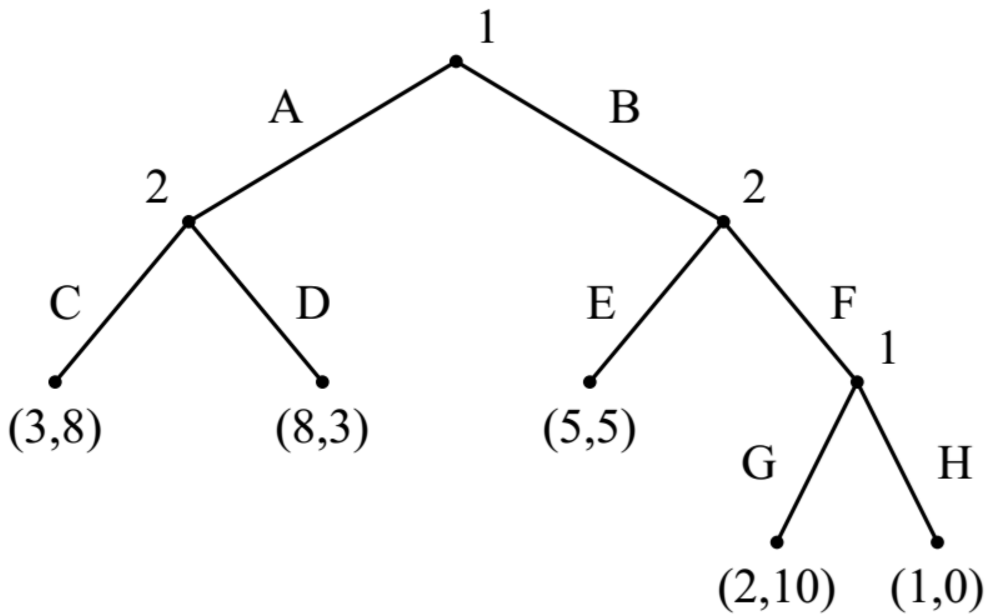


- He does it to threaten player 2, to prevent him from choosing *F* , and so gets 5
  - However, this seems like a non-credible threat
  - If player 1 reached his second decision node, would he really follow through and play *H*?

Subgame Perfect Equilibrium

- **Subgame of G rooted at h:** The subgame of *G* rooted at *h* is the restriction of *G* to the descendents of *h*.
- **Subgame of G:** The set of subgames of *G* is defined by the subgames of *G* rooted at each of the nodes in *G*.
- **Subgame Perfect Equilibrium:** *s* is a subgame perfect equilibrium of *G* iff for any subgame *G′* of *G*, the restriction of *s* to *G′* is a Nash equilibrium of *G′*. Since *G* is its own subgame, every SPE is a NE.

Subgame Perfect Equilibrium



- Which equilibria from the example are subgame perfect?
  - (*A, G*),(*C, F*): is subgame perfect
  - (*B, H*),(*C, E*): (*B, H*) is non-credible
  - (*A, H*),(*C, F*): (*A, H*) is non-credible

Computing Subgame Perfect Equilibria

- Backward Induction Algorithm:

function BACKWARDINDUCTION (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$ // h is a terminal node
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util\_at\_child \leftarrow$ BACKWARDINDUCTION$(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

Computing the Subgame Perfect Equilibria

- In zero-sum setting, the algorithm is called the MinMax Algorithm

- It's possible to speed things up by pruning nodes that will never be reached in play: "alpha-beta pruning".

function AlphaBetaPruning (node h, real α, real β) **returns** $u_1(h)$
if $h \in Z$ **then**
  return $u_1(h)$ // h is a terminal node
$best\_util \leftarrow (2\rho(h) - 3) \times \infty$ // $-\infty$ for player 1; ∞ for player 2
forall $a \in \chi(h)$ **do**
  if $\rho(h) = 1$ **then**
    $best\_util \leftarrow \max(best\_util, \text{AlphaBetaPruning}(\sigma(h, a), \alpha, \beta))$
    if $best\_util \geq \beta$ **then**
      return $best\_util$
    $\alpha \leftarrow \max(\alpha, best\_util)$
  else
    $best\_util \leftarrow \min(best\_util, \text{AlphaBetaPruning}(\sigma(h, a), \alpha, \beta))$
    if $best\_util \leq \alpha$ **then**
      return $best\_util$
    $\beta \leftarrow \min(\beta, best\_util)$
return $best\_util$

Computing Subgame Perfect Equlibria

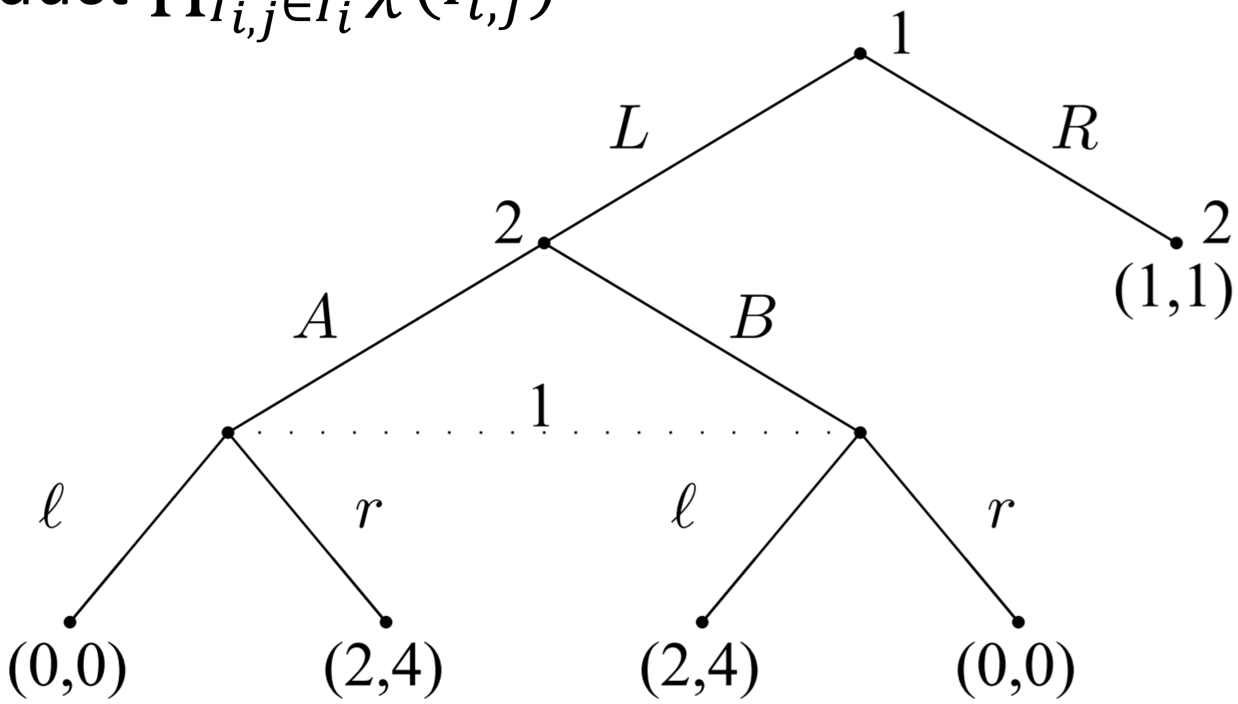
- Theorem: Given a two-player perfect-information extensive-form game with L leaves, the set of subgame-perfect equilibrium payoffs can be computed in time $O(L^3)$

Imperfect Information Extensive Games

- Imperfect information extensive-form games:
  - Each player's choice nodes partitioned into information sets.
  - Agents cannot distinguish between choice nodes in the same information set.
- An imperfect-information game (in extensive form) is a tuple $(N, A, H, Z, \chi, \rho, \sigma, u, I)$, where
  - $(N, A, H, Z, \chi, \rho, \sigma, u)$ is a perfect-information extensive-form game, and
  - $I = (I_1, \dots, I_n)$, where $I_i = (I_{i,1}, \dots, I_{i,k_i})$ is an equivalence relation on (that is, a partition of) $\{h \in H, \rho(h) = i\}$ with the property that $\chi(h) = \chi(h')$ and $\rho(h) = \rho(h')$ whenever there exists a j for which $h \in I_{i,j}$ and $h' \in I_{i,j}$.

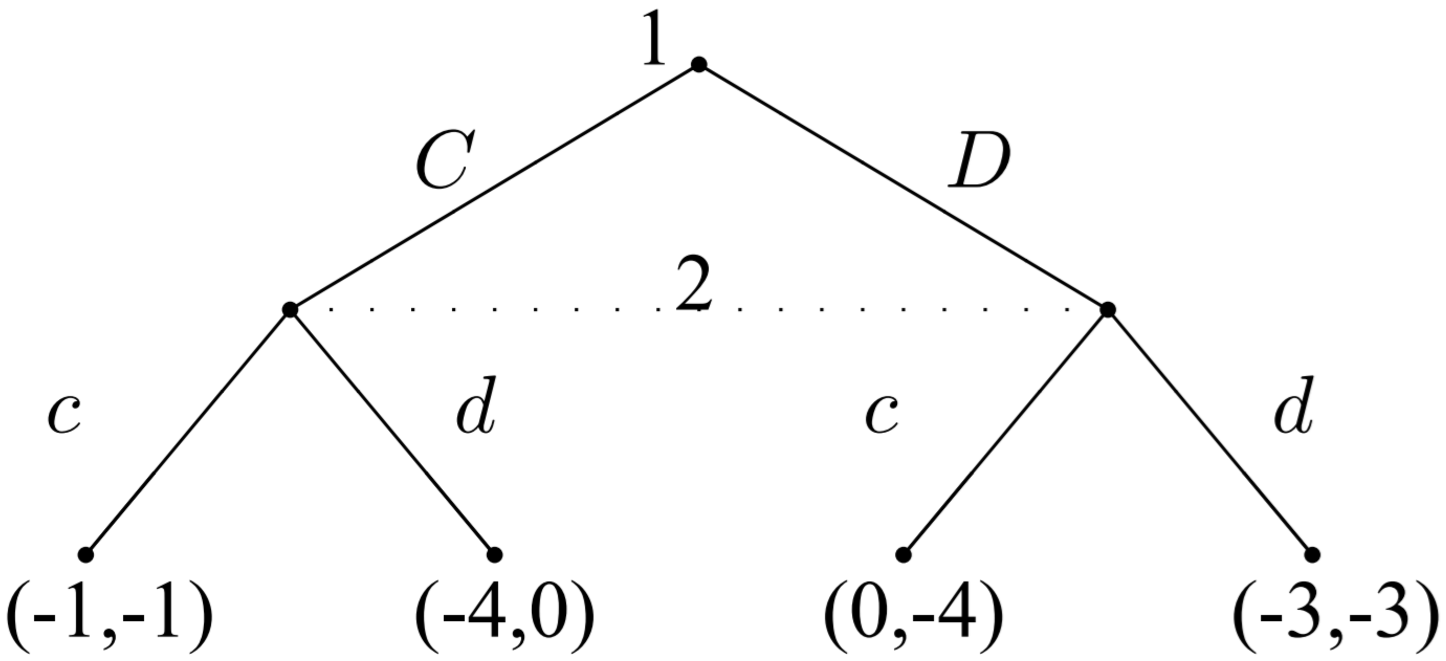
Strategies in IIEGs

- **Pure strategies:** Let G = (N,A,H,Z,χ,ρ,σ,u,I) be an imperfect information extensive-form game. Then the pure strategies of player i consist of the cartesian product $\prod_{I_{i,j} \in I_i} \chi(I_{i,j})$



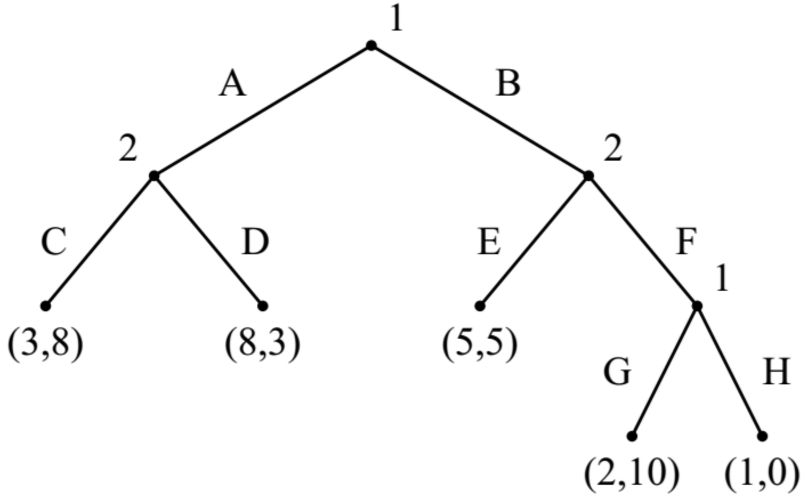
Normal-Form Games with IIEGs

- We can represent any normal form game.



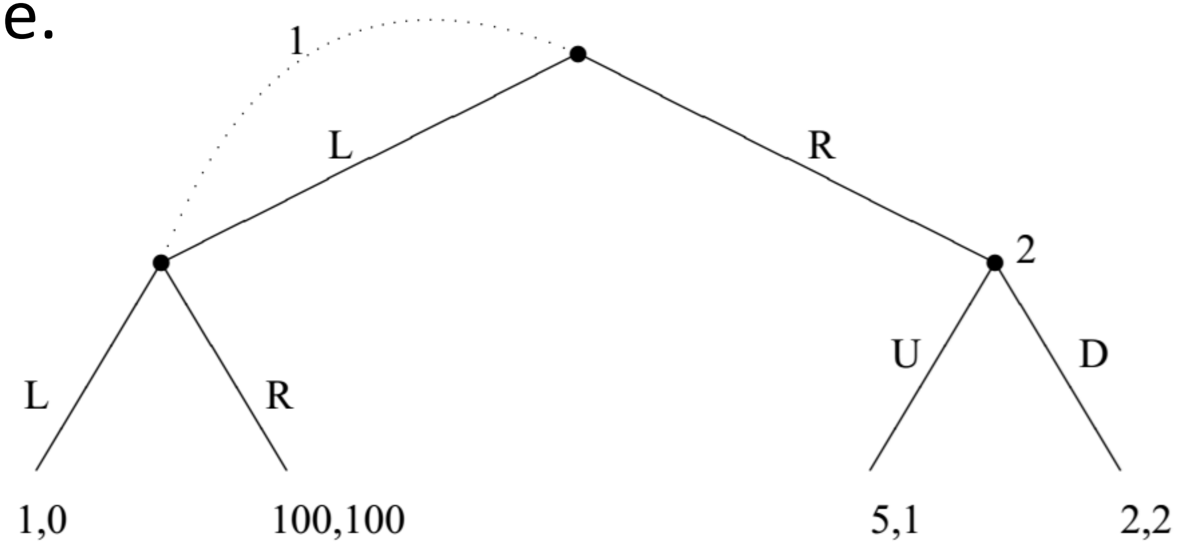
Randomized Strategies

- There are two meaningfully different kinds of randomized strategies in imperfect information extensive form games
  - mixed strategies
  - behavioral strategies
- **Behavioral Strategy:** independent coin toss every time an information set is encountered
  - *A* with probability *0.*5 and *G* with probability *0.*3
- **Mixed Strategy:** randomize over pure strategies
  - A mixed strategy that is not a behavioral strategy (*0.*6(*A, G*)*, 0.4*(*B, H*))



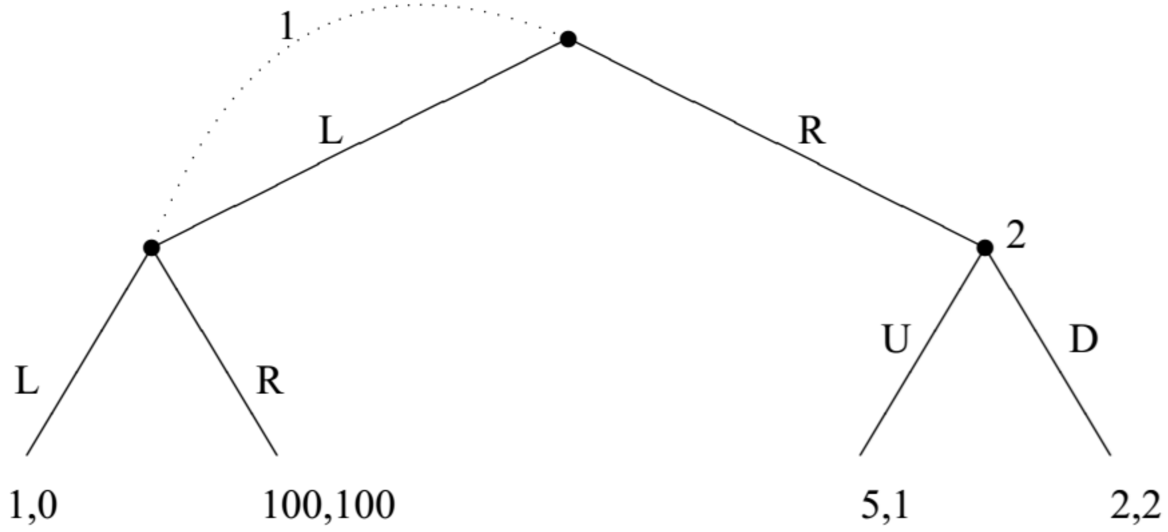
Games of Imperfect Recall

- The expressive power of behavioral and mixed strategies are not equivalent
- Imagine that player 1 sends two proxies to the game with the same strategies. When one arrives, he doesn't know if the other has arrived before him, or if he's the first one.
- Pure strategies: (L,R), (U,D)
- Mixed equilibrium:
  - *D* is dominant for 2.
  - *R,D* is better for 1 than *L,D*
  - *R, D* is an equilibrium



Games of Imperfect Recall

- Equilibrium with behavioral strategies:
  - Again, D strongly dominant for 2
  - If 1 uses the behavioral strategy (*p,*1 - *p*), his expected utility is $p^2 + 100p(1$



Games with Perfect Recall

- **Perfect recall:** Player i has perfect recall in an imperfect-information game G if for any two nodes h, h′ that are in the same information set for player i, for any path $h_0, a_0, h_1, a_1, \ldots, h_m, a_m, h$ from the root of the game to h (where the h_j are decision nodes and the a_j are actions) and for any path $h_0, a'_0, h'_1, a'_1, \ldots, h'_m, a'_m, h'$ from the root to h′ it must be the case that:
  - m = m’
  - For all $0 \leq j \leq m$, if $\rho(h_j) = i$ then h_j and h'_j are in the same equivalence class
  - For all $0 \leq j \leq m$, if $\rho(h_j) = i$ then $a_j = a'_j$

Games with Perfect Recall

- Theorem (Kuhn): In a game of perfect recall, any mixed strategy of a given agent can be replaced by an equivalent behavioral strategy, and any behavioral strategy can be replaced by an equivalent mixed strategy. Here two strategies are equivalent in the sense that they induce the same probabilities on outcomes, for any fixed strategy profile (mixed or behavioral) of the remaining agents.

Sequence Form Representation

- **Sequence Form Representation:** Let G be an imperfect-information sequence form game of perfect recall. The sequence-form representation of G is a tuple(N,Σ,g,C):
  - N is a set of agents
  - $\Sigma = (\Sigma_1, \ldots, \Sigma_n)$, where Σ_i is the set of sequences available to agent i;
  - $g = (g_1, \ldots, g_n)$, where $g_i : \Sigma \rightarrow R$ is the payoff function for agent i;
  - $C = (C_1, \ldots, C_n)$, where C_i is a set of linear constraints on the realization probabilities of agent i.

Sequence Form Representation

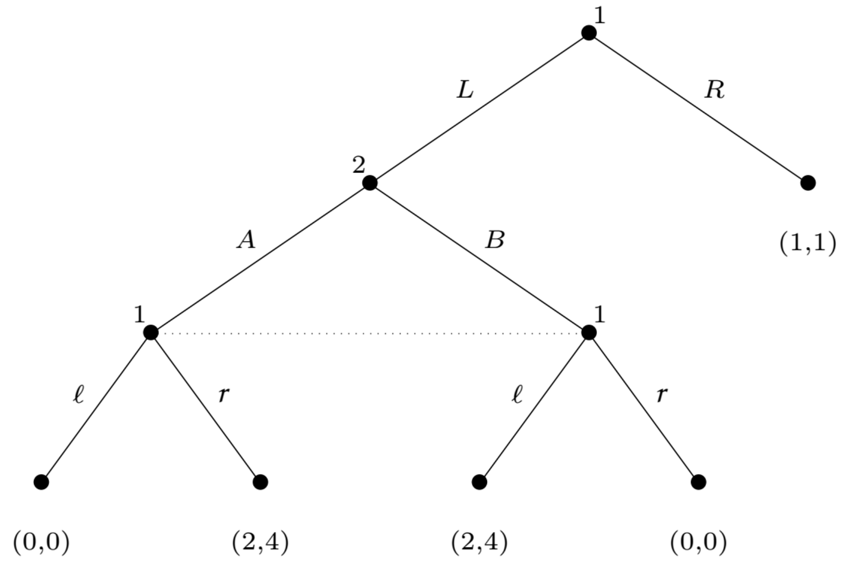
- **Sequence:** A sequence of actions of player $i \in N$, defined by a node $h \in H \cup Z$ of the game tree, is the ordered set of player i's actions that lie on the path from the root to h. Let Ø denote the sequence corresponding to the root node. The set of sequences of player i is denoted Σ_i, and $\Sigma = \Sigma_1 \times \cdots \times \Sigma_n$ is the set of all sequences.

- **Payoff function:** The payoff function $g_i : \Sigma_i \rightarrow R$ for agent i is payoff function given by $g(\sigma) = u(z)$ if a leaf node $z \in Z$ would be reached when each player played his sequence $\sigma_i \in \sigma$, and by $g(\sigma) = 0$ otherwise.

Sequence Form Representation

- $\Sigma_1 = \{\emptyset, L, R, Ll, Lr\}$
- $\Sigma_2 = \{\emptyset, A, B\}$

	$\emptyset$	A	B
$\emptyset$	$0,0$	$0,0$	$0,0$
L	$0,0$	$0,0$	$0,0$
R	$1,1$	$0,0$	$0,0$
$L\ell$	$0,0$	$0,0$	$2,4$
Lr	$0,0$	$2,4$	$0,0$



Sequence Form Representation

- Consider an agent i following a behavioral strategy that assigned probability $\beta_i(h,a_i)$ to taking action a_i at a given decision node h.

- **Realization plan of** β_i**:** The realization plan of β_i for player $i \in N$ is a mapping $r_i : \Sigma_i \rightarrow [0, 1]$ defined as $r_i(\sigma_i) = \prod_{c \in \sigma_i} \beta_i(c)$. Each value $r_i(\sigma_i)$ is called a realization probability.

Sequence Form Representation

- G is a game of perfect recall. This entails that, given an information set $I \in I_i$, there must be one single sequence that player i can play to reach all of his nonterminal choice nodes $h \in I$. We denote this mapping as $seq_i : I_i \rightarrow \Sigma_i$, and call $seq_i(I)$ the sequence leading to information set I.
- As long as the new sequence still belongs to Σ_i, we say that the sequence $\sigma_i a_i$ extends the sequence σ_i. We denote by $Ext_i : \Sigma_i \rightarrow 2^{\Sigma_i}$ a function mapping from sequences to sets of sequences, where $Ext_i(\sigma_i)$ denotes the set of sequences that extend the sequence σ_i.
- We introduce the $Ext_i(I)$ = shorthand $Ext_i(I) = Ext_i(seq_i(I))$
- **Realization Plan:** A realization plan for player $i \in N$ is a function $r_i : \Sigma_i \rightarrow [0, 1]$ satisfying the following constraints:

$$r_i(\emptyset) = 1$$

$$\sum_{\sigma'_i \in \mathrm{Ext}_i(I)} r_i(\sigma'_i) = r_i(\mathrm{seq}_i(I)) \qquad \forall I \in I_i$$

$$r_i(\sigma_i) \geq 0 \qquad \forall \sigma_i \in \Sigma_i$$

Sequence Form Representation Best Response Computation

- An LP for best response computation in sequence form representation:

$$
\begin{aligned}
\text{maximize} \quad & \sum_{\sigma_1 \in \Sigma_1} \left(\sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2) \right) r_1(\sigma_1) \\
\text{subject to} \quad & r_1(\emptyset) = 1 \\
& \sum_{\sigma_1' \in \mathrm{Ext}_1(I)} r_1(\sigma_1') = r_1(\mathrm{seq}_1(I)) && \forall I \in I_1 \\
& r_1(\sigma_1) \geq 0 && \forall \sigma_1 \in \Sigma_1
\end{aligned}
$$

- In an equilibrium, player 1 and player 2 best respond simultaneously. However, if we treated both r_1 and r_2 as variables then the objective function would no longer be linear.

Sequence Form Representation Best Response Computation

- The dual form has not this problem:

$$\begin{array}{ll} \text{minimize} & v_0 \\ \text{subject to} & v_{\mathcal{I}_1(\sigma_1)} - \displaystyle\sum_{I' \in \mathcal{I}_1(\text{Ext}_1(\sigma_1))} v_{I'} \geq \sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2) \quad \forall \sigma_1 \in \Sigma_1 \end{array}$$

- Description:
  - Denote the variables of our dual LP as v; there will be one v_I for every information set $I \in I_1$ and one additional variable v_0 (corresponding to the first constraint)
  - $\mathcal{I}_i(\sigma_i): \Sigma_i \to I_i \cup \{0\}$: It is defined to be 0 iff $\sigma_i = \emptyset$, and to be the information set $I \in I_i$ in which the final action in σ_i was taken otherwise.

Sequence Form Representation Equilibria Computation

- Zero-sum games:

$$\begin{aligned}
\text{minimize} \quad & v_0 \\
\text{subject to} \quad & v_{\mathcal{I}_1(\sigma_1)} - \sum_{I' \in \mathcal{I}_1(\text{Ext}_1(\sigma_1))} v_{I'} \geq \sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2) \qquad \forall \sigma_1 \in \Sigma_1 \\
& r_2(\emptyset) = 1 \\
& \sum_{\sigma'_2 \in \text{Ext}_2(I)} r_2(\sigma'_2) = r_2(\text{seq}_2(I)) \qquad \forall I \in I_2 \\
& r_2(\sigma_2) \geq 0 \qquad \forall \sigma_2 \in \Sigma_2
\end{aligned}$$

- LCP form for general-sum games:

$$r_1(\sigma_1)\left[\left(v^1_{\mathcal{I}_1(\sigma_1)} - \sum_{I' \in \mathcal{I}_1(\text{Ext}_1(\sigma_1))} v^1_{I'}\right) - \left(\sum_{\sigma_2 \in \Sigma_2} g_1(\sigma_1, \sigma_2) r_2(\sigma_2)\right)\right] = 0 \qquad \forall \sigma_1 \in \Sigma_1$$

$$r_2(\sigma_2)\left[\left(v^2_{\mathcal{I}_2(\sigma_2)} - \sum_{I' \in \mathcal{I}_2(\text{Ext}_2(\sigma_2))} v^2_{I'}\right) - \left(\sum_{\sigma_1 \in \Sigma_1} g_2(\sigma_1, \sigma_2) r_1(\sigma_1)\right)\right] = 0 \qquad \forall \sigma_2 \in \Sigma_2$$