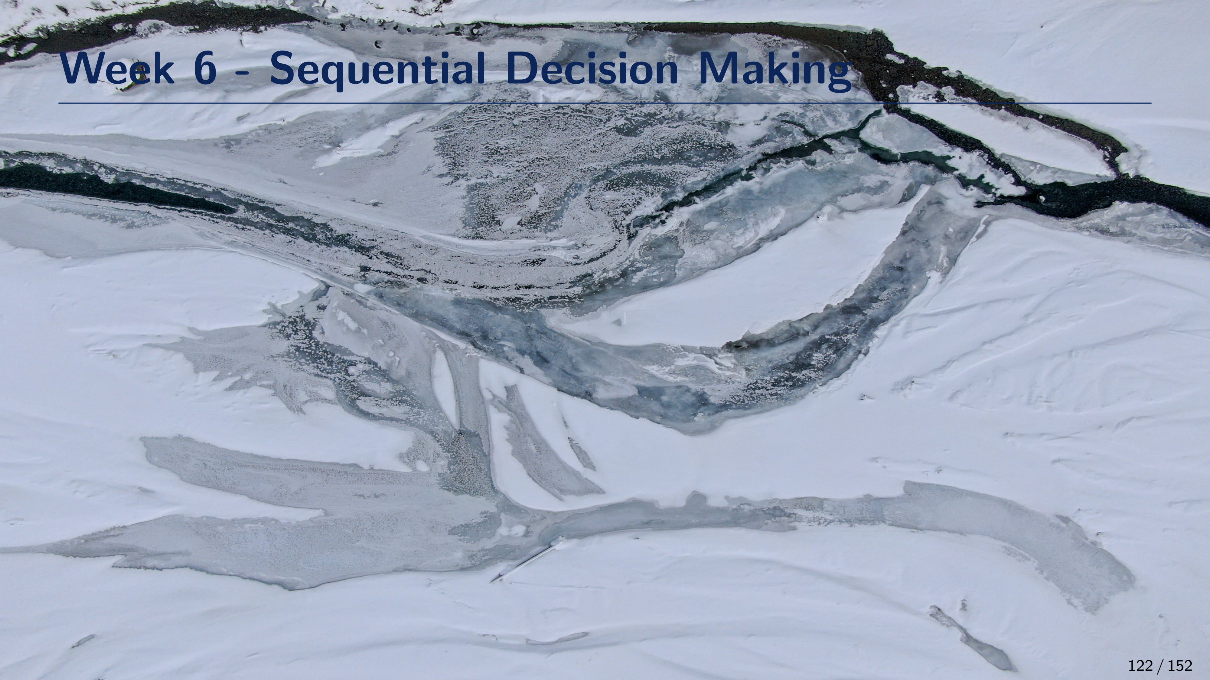
Week 6 - Sequential Decision Making



Extensive Form Games

Sequential moves

- Let's use a tree-like structure, similarly to Chess

Extensive Form Games

Imperfect information

- Player can't see opponent's cards
- Consider these two situations
  $(A\spadesuit 8\diamondsuit) - (K\heartsuit K\diamondsuit)$ and $(A\spadesuit 8\diamondsuit) - (2\heartsuit 7\heartsuit)$
- Even though that these situations are different, player 1 can't distinguish them
- Let's make some game states indistinguishable (from the player's) point of view, so that he must use the same strategy in all the nodes he can't tell apart

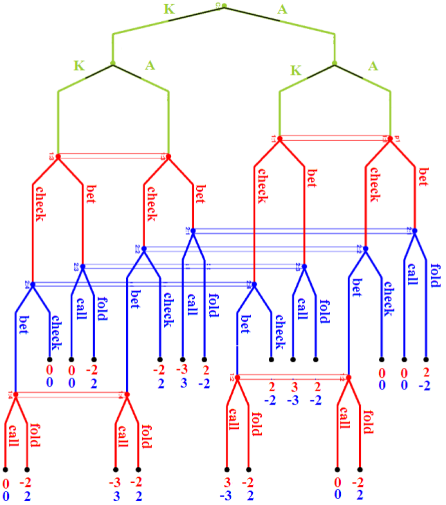
Extensive Form Games

Chance

- Let's add another player, the chance player (typically denoted as the player 0 or the player c)
- The chance does plays according to some fixed probability distribution

Extensive Form Games Tree Example

Simple poker-like game

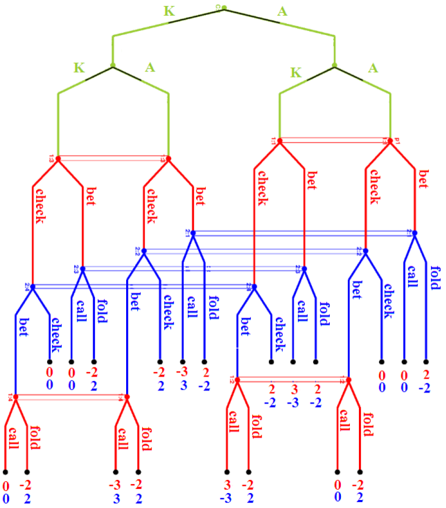


Extensive Form Games Formalization

An extensive form game consists of

- A finite set $N = \{1, 2...n\}$ (the set of **players**).
- A finite set H of sequences. Each member of H is a **history**, each component of history is an **action**. The empty sequence is in H, and every prefix of a history is also history $((h, a) \in H \implies (h \in H))$. $h \sqsubseteq h^{'}$ denotes that h is a prefix of $h^{'}$. $Z \subseteq H$ are the terminal histories (they are not a prefix of any other history).
- The set of actions available after every non-terminal history $A(h) = \{a : (h, a) \in H\}$.
- A function p that assigns to each non-terminal history an **acting player** (member of $N \cup c$, where c stands for chance).
- A function f_c that associates with every history for which $p(h) = c$ a probability measure on $A(h)$. Each such probability measure is independent of every other such measure.
- For each player $i \in N$, a partition $\mathcal{I}_i$ of $h \in H : p(h) = i$. $\mathcal{I}_i$ is the **information partition** of player i. A set $I_i \in \mathcal{I}_i$ is an **information set** of player i.
- For each player $i \in N$ an **utility function** $u_i : Z \to \mathbb{R}$.

Extensive Form Games Tree Example



Extensive Form Games Formalization

- $N = \{1, 2\}$
- $H = \{(\varnothing), (K), (K, A), (K, A, bet), (K, A, bet, call), \cdots\}$
- $Z \subseteq H$, $Z = \{(K, A, bet, call), (K, A, bet, fold), \cdots\}$
- $A(K, A) = \{bet, fold\}$
  $A(K, A, bet) = \{call, fold\}$
- $p(\varnothing) = c, P(K) = c, P(K, A) = 1, P(K, A, check) = 2$
- $f_c(\varnothing) = 0.5A, 0.5K$
  $f_c(A) = 0.5A, 0.5K$
- $\mathcal{I}_1 = \{\{(K, A), (K, K)\}, \{(K, A, check), (K, K, check)\}\ldots\}$
  $\mathcal{I}_2 = \{\{(A, K), (A, K)\}, \{(A, K, check), (K, K, check)\}\ldots\}$
- $u_1(K, A, bet, call) = -3$
  $u_2(K, A, bet, fold) = -1$

Strategies

- Now the player does not choose a row/column, instead an edge in the game tree
- Since we need that the player can't distinguish the states merged into information sets, we allow the player to choose an action in information sets in contrast to histories/nodes
- This way, the player must play the same strategy in all histories grouped in that information set.

Definition: Behavior Strategy

Behavior Strategy of player i, $sigma_i$, is a collection $(\sigma_i(I_i))$ of independent probability measures, where $\sigma_i(I_i)$ is the probability measure over $A(I_i)$. $\sigma_i(h, a)$ denotes the probability assigned by $\sigma_i(I_i)$ to the action a. Strategy profile of the game $\sigma = (\sigma_0, \sigma_1, ...\sigma_n)$ is a collection of strategies for all players in the game.

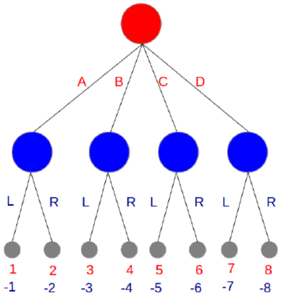
Strategies

- Extensive form games is a powerful model, and it allows us to capture some non-realistic properties
- Real life players do not forget information that they already knew
- This property is called **perfect recall**. We say that an extensive form games satisfies perfect recall, if all players can recall their previous actions and the corresponding information sets
  - We will see shortly to formalize this important property

Extensive Form Games To Normal Form Games

- If the game satisfies perfect recall, we can convert an extensive form game to an equivalent normal form game
- A pure strategy in normal form game corresponds to all combinations of pure strategies in information sets of that player

Extensive Form Games To Normal Form Games



	A	B	C	D
(A-L, B-L, C-L, D-L)	1;-1	3;-3	5;-5	7;-7
(A-L, B-L, C-L, D-R)	1;-1	3;-3	5;-5	8;-8
(A-L, B-L, C-R, D-L)	1;-1	3;-3	6;-6	7;-7
(A-L, B-L, C-R, D-R)	1;-1	3;-3	6;-6	8;-8
(A-L, B-R, C-L, D-L)	1;-1	4;-4	5;-5	7;-7
(A-L, B-R, C-L, D-R)	1;-1	4;-4	5;-5	8;-8
(A-L, B-R, C-R, D-L)	1;-1	4;-4	6;-6	7;-7
(A-L, B-R, C-R, D-R)	1;-1	4;-4	6;-6	8;-8
(A-R, B-L, C-L, D-L)	2;-2	3;-3	5;-5	7;-7
(A-R, B-L, C-L, D-R)	2;-2	3;-3	5;-5	8;-8
(A-R, B-L, C-R, D-L)	2;-2	3;-3	6;-6	7;-7
(A-R, B-L, C-R, D-R)	2;-2	3;-3	6;-6	8;-8
(A-R, B-R, C-L, D-L)	2;-2	4;-4	5;-5	7;-7
(A-R, B-R, C-L, D-R)	2;-2	4;-4	5;-5	8;-8
(A-R, B-R, C-R, D-L)	2;-2	4;-4	6;-6	7;-7
(A-R, B-R, C-R, D-R)	2;-2	4;-4	6;-6	8;-8

Extensive Form Games To Normal Form Games

Lemma: Extensive Form Games to Normal Form Games

Given any two-player extensive form game with perfect recall, it's possible to create an equivalent normal form game

- Therefore, all properties that we showed for the normal-from games, do also hold for the extensive games.
- Existence of the equilibrium.
- There is always some pure best response.
- Nice properties of equilibrium for two players zero sum games.
- Not-so-nice properties of other games ...
- It is also easy to represent any normal form game as an extensive game.

Sequence Form

- The conversion to normal form game allows us to solve the extensive form game using the techniques we already know (linear programming for two playres zero-sum case)
- Unfortunately, as we have seen, the constructed game can be exponentially large
- Can we somehow fix the fact that the constructed game can be exponential?
- The idea is to represent all paths - **sequences** for the players

Definition: Sequence

A sequence of moves of a player i is the sequence of his actions on the path from the root (history $(\varnothing)$) to the node/history h, and is denoted $s_i(h)$.

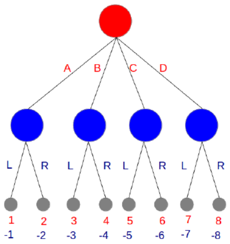
Sequence Form

- $s_1(A, K, check, bet, call) = (check, call)$
- $s_2(A, K, check, bet, call) = (bet)$
- Let S_i be the set of all sequence for a player
- Clearly, the size of S_i is linear in the size of the game tree
- Using the sequences, we can now conveniently formalize the perfect recall

Definition: Perfect Recall

The game satisfies perfect recall, iff for all players, $s_i(h_1) = s_i(h_2)$ for any two histories $h_1, h_2 \in I_i$.

Sequence Form

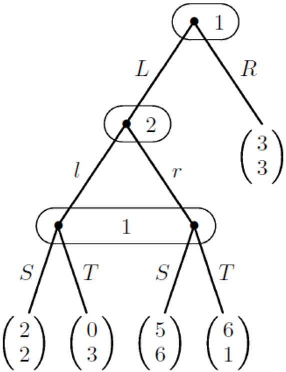


	A	B	C	D
(A-L, B-L, C-L, D-L)	1;-1	3;-3	5;-5	7;-7
(A-L, B-L, C-L, D-R)	1;-1	3;-3	5;-5	8;-8
(A-L, B-L, C-R, D-L)	1;-1	3;-3	6;-6	7;-7
(A-L, B-L, C-R, D-R)	1;-1	3;-3	6;-6	8;-8
(A-L, B-R, C-L, D-L)	1;-1	4;-4	5;-5	7;-7
(A-L, B-R, C-L, D-R)	1;-1	4;-4	5;-5	8;-8
(A-L, B-R, C-R, D-L)	1;-1	4;-4	6;-6	7;-7
(A-L, B-R, C-R, D-R)	1;-1	4;-4	6;-6	8;-8
(A-R, B-L, C-L, D-L)	2;-2	3;-3	5;-5	7;-7
(A-R, B-L, C-L, D-R)	2;-2	3;-3	5;-5	8;-8
(A-R, B-L, C-R, D-L)	2;-2	3;-3	6;-6	7;-7
(A-R, B-L, C-R, D-R)	2;-2	3;-3	6;-6	8;-8
(A-R, B-R, C-L, D-L)	2;-2	4;-4	5;-5	7;-7
(A-R, B-R, C-L, D-R)	2;-2	4;-4	5;-5	8;-8
(A-R, B-R, C-R, D-L)	2;-2	4;-4	6;-6	7;-7
(A-R, B-R, C-R, D-R)	2;-2	4;-4	6;-6	8;-8

	0	A	B	C	D
0	0;0	0;0	0;0	0;0	0;0
L_a	0;0	1;-1	0;0	0;0	0;0
R_a	0;0	2;-2	0;0	0;0	0;0
L_b	0;0	0;0	3;-3	0;0	0;0
R_b	0;0	0;0	4;-4	0;0	0;0
L_c	0;0	0;0	0;0	5;-5	0;0
R_c	0;0	0;0	0;0	6;-6	0;0
L_d	0;0	0;0	0;0	0;0	7;-7
R_d	0;0	0;0	0;0	0;0	8;-8

Figure: Normal form vs. sequence form

Extensive Form Games To Normal Form Games



$$A = \begin{array}{cc} l & r \\ \boxed{\begin{array}{cc} 2 & 5 \\ 0 & 6 \\ 3 & 3 \\ 3 & 3 \end{array}} \end{array} \begin{array}{l} \langle L, S\rangle \\ \langle L, T\rangle \\ \langle R, S\rangle \\ \langle R, T\rangle \end{array} \qquad B = \begin{array}{cc} l & r \\ \boxed{\begin{array}{cc} 2 & 6 \\ 3 & 1 \\ 3 & 3 \\ 3 & 3 \end{array}} \end{array} \begin{array}{l} \langle L, S\rangle \\ \langle L, T\rangle \\ \langle R, S\rangle \\ \langle R, T\rangle \end{array}$$

$$A = \begin{array}{ccc} \emptyset & l & r \\ \boxed{\begin{array}{ccc} & & \\ & & \\ 3 & & \\ & 2 & 5 \\ & 0 & 6 \end{array}} \end{array} \begin{array}{l} \emptyset \\ L \\ R \\ LS \\ LT \end{array} \qquad B = \begin{array}{ccc} \emptyset & l & r \\ \boxed{\begin{array}{ccc} & & \\ & & \\ 3 & & \\ & 2 & 6 \\ & 3 & 1 \end{array}} \end{array} \begin{array}{l} \emptyset \\ L \\ R \\ LS \\ LT \end{array}$$

Figure: Normal form, sequence form, the payoff matrices

Sequence Form

- We can see that the sequence form is a valid, lossless representation
- Can we just use it directly to create another linear program to solve the game?
- But we can't choose the rows/columns of the payoff matrix A/B arbitrarily as in normal form game!
- Denote the probability the players assigns to his sequence $s_i(h)$ as $x(s_i(h))$.
- See that x now might not be a probability distribution to form a correct strategy! (find an example)
- Let's add some constraints for the sequence's probabilities, so that these probabilities form a valid strategy

$$\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h))$$

$$\sigma(\varnothing) = 1$$

Sequence Form

$$\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h))$$

$$x(\varnothing) = 1$$

- Note that these restrictions are linear
- To compute the expected payoff given the sequence strategies, we just need to go through all the terminal nodes, and compute the reach probability of that node

$$\sum_{t \in Z} \sigma_c(t)\sigma_1(t)\sigma_2(t)u_i(t)$$

- Now we can formalize the the utility matrix A (for player 1)

$$A_{x,y} = \sum_{t \in Z \mid s_1(t)=x, s_2(t)=y} \sigma_c(t)\sigma_1(t)\sigma_2(t)u_1(t)$$

Sequence Form

- We can write the $\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h))$, $x(\varnothing) = 1$ as $Ex = e$
- Similarly, we can use $Fy = f$ for the second player

$$\begin{bmatrix} 1 & & & & \\ -1 & 1 & 1 & & \\ & -1 & & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Figure: $Ex = e$, $Fy = f$

- To compute a best response (given a fixed strategy y)

$$\begin{aligned} \max_x \quad & x^T A \\ \text{subject to} \quad & Ex = e, \; x \geq 0 \end{aligned}$$

- Now consider the dual problem

$$\begin{aligned} \min_x \quad & e^T u \\ \text{subject to} \quad & E^T u \geq Ay \end{aligned}$$

Sequence Form

- The dual problem

$$\min_{x} \quad e^T u$$
$$\text{subject to} \quad E^T u \geq Ay$$

- This dual finds the best response (for the player 1).
- In the case of zero-sum, player 2 wants to minimize this value.
- The strategy of player 2 is the y
  - We need to make sure that the y forms a valid strategy: $Fy = f$
  - Now the y won't be fixed, but the player 2 chooses the strategy he wants to play

Sequence Form

$$\min_{u} \quad e^T u$$
$$\text{subject to} \quad E^T u \geq Ay$$

- We need to make sure that the y forms a valid strategy: $Fy = f$
- Now the y won't be fixed, but the player 2 chooses the strategy he wants to play

Final Linear Program

$$\min_{u,y} \quad e^T u$$
$$\text{subject to} \quad E^T u \geq Ay,\; Fy = f, y \geq 0$$

Theorem - Sequence Form LP

The solution to the "Final Linear Program" corresponds to a Nash equilibrium (sequence form) for two players, zero-sum game

Corolllary

There's a polynomial algorithm to compute a Nash equilibrium for two players zero-sum extensive form games