Lecture 5

Ashish Rastogi, Keshav Kunal

August 30, 2002

In this lecture, we will discuss games in extensive form, define the notion of a tree and subgame perfect equilibria, and understand some underlying properties of the Nash equilibrium in such games. Next, we will broach the theory of utility.

Contents

1 Games In Extensive Form

An *extensive game* is an explicit description of the sequential structure of decision problems encountered by the players in a strategic situation. The model allows us to study solutions in which each player can consider her plan of action not only at the beginning of the game, but also at any point of time at which he has to make a decision.

A general model of an extensive game allows each player, when making his choices, to be imperfectly informed about what has happened in the past. However, in this lecture, we limit our attention to the simpler model where each player is perfectly informed about the players' previous actions at each point in the game.

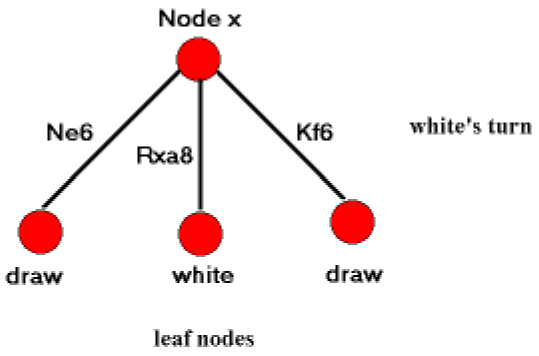

Figure 1: A node one level above the leaf nodes in the tree for the game of chess. It is white's turn to move.

Definition An extensive game is a detailed description of the sequential structure of the decision problems encountered by the players in a strategic situation. There is perfect information in such a game if each player, when making any decision, is perfectly informed of all the events that have previously occurred and of the pay-offs associated when the game ends. Further, players take turns in making moves, and the number of moves in a game is finite.

1.1 Chess

Consider the game of chess. There are two players *white* and *black* who move alternatively. At any stage in the game, depending upon the board position, each player has a finite number of moves to choose from. The board position depends upon the previous moves of both *white* and *black*. When the game ends, there are three possible outcomes: *white wins*, *black wins*, *draw*. In modeling this game, let us assume that a player remembers (is perfectly informed) of all previous moves. Further, let us assume that the total number of moves is bounded by a maximum, after which, if no result is obtained, a *draw* would be forced.

Such a game of chess may be modeled as a tree. The root node corresponds to the initial state of the game. At the beginning of the game, it is *white*'s turn to move. At any time, the state of the chess board is determined by the sequence of moves that have been made from the starting of the game. The different edges emanating from the root correspond to possible first moves by *white*, and these edges are incident on nodes that correspond to the board position (state) achieved by the move. From each of these nodes, then there are edges that emanate and correspond to a possible move by *black*, and these edges are incident on nodes that then correspond to the board position after a move by *white* and a move by *black*. The game tree is constructed in such a manner, and the leaves correspond to a situation where either the game has ended (it has resulted in a *white win* or a *black win*) or the maximum number of moves have been player (in which case, we declare it a *draw*).

Hence, each leaf can be labeled with one of *white*, *black* or *draw* corresponding to the outcome the leaf represents. Consider a node x that has as its children, only leaf nodes (Figure 1). Suppose it is *white*'s turn when the game is in the state corresponding to node x. It is clear that once the game reaches this state, then *white* can play *Rxa8* (move the rook to the square $a8$) and win the game. Hence, we can label node x as *white*, because we know that once the game reaches node x, *white* is sure to win. If at node x, there are no children with label *white*, then white would look to *draw*, and in such a case node x can be labeled *draw*. Finally, if *white* can neither win nor draw from the board position at x, then *white* is sure to lose and we may label node x as *black*. In this manner, we can fold up the finite game-tree and label each node with one of the tree labels depending on the labels of its children and the player who has her move.

This procedure is more formally described as *Zermelo's Algorithm* for solutions of games in extensive form later in these scribes. Note that if the root node is labeled with *white*, then *white* can force a win. Similarly, if it is labeled *draw* or *black*, *white* would *draw* or *lose* respectively.

The above analysis of the game of chess implies that chess is a trivial game because the outcome may be determined even before the first move has been made. If such is the case, then why is chess so interesting? Should there not, then, be a computer that should *always* be able to defeat a human?

In our entire analysis, we assumed that the game tree was easily constructed and available to us. We assumed that it was finite, and ignored its size. Now, let us consider what would be the size of a typical game tree of chess. Let us assume that the maximum number of moves is 64. The number of possible board positions is roughly $\binom{64}{32} \times 32!$ assuming that no pieces are killed. Hence, the number of nodes in the game tree is extremely large for it to be completely stored in any practical computer memory or human mind, and so the assumption that players have complete information about the sequence of moves that have been played, and the pay-offs at the leaf nodes turns out to be infeasible for the game of chess, which is why it is still interesting.

1.2 Simplified Two Player PickStick

Consider the following two player game. Suppose that there are 10 sticks on the board. Two players (player 1 and 2) take turns alternately, and in each turn, a player may either pick 1 stick or pick 2 sticks. The player who picks the last stick wins. What would be the strategy of each player in this game?

Suppose player 1 adopts the following strategy: at any stage, after player 1's move, the number of sticks left on board are a multiple of 3. This strategy implies that starting with 10 sticks, player 1 picks up 1 stick (thus 9 sticks remain, which is a multiple of 3). Then, if player 2 picks up 1 stick, then player 1 picks up 2, and if player 2 picks up 2 sticks, then player 1 picks up 1 (thus picking up a total of 3 sticks in one round of player 1 followed by player 2) which will ensure that after every move of player 1, the number of sticks is divisible by three. What does such a strategy ensure? Consider the last but one round, player 1 would have played so that 3 sticks would be left on the house. In such a case, no matter what player 2 players, player 1 always has a winning move.

Simplified *PickStick* can also be modeled using a game tree. Each node in the tree corresponds to a particular number of sticks left on board. The root node then corresponds to 10 sticks. The leaf nodes correspond to no sticks left on the board and the player who arrives at a leaf node wins. Two edges emanate from each internal node, corresponding to picking 1 or 2 sticks. In such a case, consider a node where 3 sticks are left, and it is player 2's turn. It is clear that if she picks 1 stick, then player 1 can pick 2 and win, and in the other case, if she picks 2 sticks, then player 1 can pick 1 and win. Therefore, we can label this node as one where player 1 can definitely win. Building up the tree like this, we get the strategy mentioned in earlier paragraph.

1.3 Generalized Two Player PickStick

In the generalized two player *PickStick*, each of the two players can pick 1 through k sticks, and initially there are m sticks on the board. As described earlier, a winning strategy corresponds to picking sticks such that the number of boards left on the board are a multiple of $k+1$. Once again, a player who can ensure this strategy will necessarily win the game. In the simplified *PickStick* mentioned earlier, we have $m = 10$ and $k = 2$. Mathematically, an *extensive game with perfect information* has the following components.

- A set N (the set of *players*).
- A set H of sequences (finite or infinite) that satisfies the following three properties.
  - The empty sequence ϕ is a member of H.
  - If $(a^k)_{k=1,\ldots,K} \in H$ (where K may be infinite) and $L < K$ then $(a^k)_{k=1,\ldots,L} \in H$.
  - If an infinite sequence $(a^k)_{k=1}^{\infty}$ satisfies $(a^k)_{k=1,\ldots,L} \in H$ for every positive integer L then $(a^k)_{k=1}^{\infty} \in H$.

  (Each member of H is a *history*; each component of a history is an **action** taken by a player.) A history $(a^k)_{k=1,\ldots,K} \in H$ is **terminal** if it is infinite or if there is no a^{K+1} such that $(a^k)_{k=1,\ldots,K+1} \in H$. The set of terminal histories is denoted Z.
- A function P that assigns to each nonterminal history (each member of $H \setminus Z$) a member of N. (P is the **player function**, $P(h)$ being the player who takes an action after history h.)
- For each player $i \in N$ a preference relation $\succsim_i$ on Z (the **preference relation** of player i.)

Sometimes it is convenient to specify the structure of an extensive game without specifying the player's preferences. We refer to a triple $< N, H, P >$ whose components satisfy the first three conditions in definition as an **extensive game form with perfect information**.

If the set H of possible histories is finite then the game is $finite$. If the longest history is finite then the game has a *finite horizon*. Let h be a history of length k; we denote by (h, a) the history of length $k+1$ consisting of h followed by a.

Throughout these scribes we refer to an extensive game with perfect information simply as an "extensive game". Further, we assume that the game is *finite*. We interpret such a game as follows. After any nonterminal history h, player $P(h)$ chooses an action from the set

$$A(h) = \{a : (h, a) \in H\}$$

The empty history is the starting point of the game; we sometimes refer to it as the *initial history*. At this point, player $P(\phi)$ chooses a member of $A(\phi)$. For each possible choice a^0 from this set player $P(a^0)$ subsequently chooses a member of the set $A(a^0)$; this choice determines the next player to move, and so on. A history after which no more choices have to be made is terminal.

1.4 Example: Three Player PickStick

Consider the following variation of PickStick game described earlier. There are m sticks on the board initially. Three players A, B and C are playing the game and take turns. Each player can pick up i sticks ($i \in \{1, 2, \ldots, k\}$) in one turn. The player that picks last wins the game. How would the players play this game?

This game can be modeled in extensive form as:

- The set of players, $N = \{A, B, C\}$;
- The set of histories $H = \{(a^i)_{i=1,\ldots K}$ where a^j is the number of sticks picked ($0 < a^j \leq k$) by the player who has the jth turn (A if $j \equiv 1 \pmod 3$, B if $j \equiv 2 \pmod 3$, C if $j \equiv 3 \pmod 3$).
- For a terminal history $h \in Z$, let the length $l(h)$ denote the total number of actions in h. Player A wins if she is the last to pick sticks, that is $l(h) \equiv 1 \pmod 3$. Define $H_p = \{h \in Z | l(h) \equiv p(mod\ 3)\}$ for $p \in \{1, 2, 3\}$. Therefore we have H_1, H_2 and H_3 as the set of histories for which player A, B and C win respectively.
- The preference relation for player A, $\succ_A$ is given as follows:
  1. For $h_1, h_2 \in H_i$, $h_1 \sim_A h_2$
  2. For $h_1 \in H_1$ and $h_2 \in H_2$ or $h_2 \in H_3$, $h_1 \succ_A h_2$
- Preference relation for player B and player C ($\succ_B$ and $\succ_C$ is defined similarly.

1.5 Example

Two persons use the following procedure to share two desirable identical indivisible objects. One of them proposes an allocation, which the other either accepts or rejects. In the event of rejection, neither person receives either of the objects. Each person cares only about the number of objects he obtains.

An extensive game that models the individuals predicament is $< N, H, P, (\succsim_i) >$ where

- $N = \{1, 2\}$;

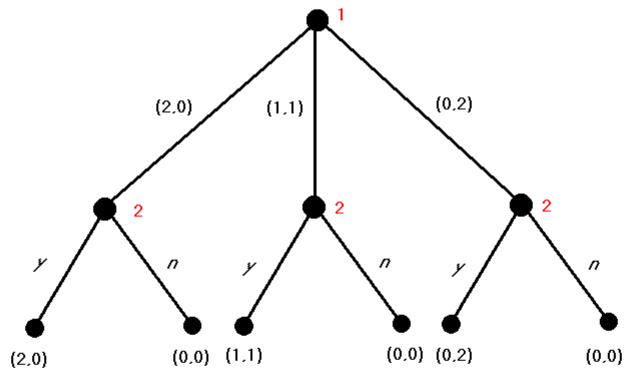

Figure 2: An extensive game that models the procedure for allocating two identical indivisible objects between two people.

- H consists of the ten histories ϕ, (2,0), (1,1), (0,2), ((2,0), y), ((2,0), n), ((1,1), y), ((1,1),n), ((0,2),y), ((0,2),n);
- $P(\phi) = 1$ and $P(h) = 2$ for every nonterminal history $h \neq \phi$.
- $((2,0), y) \succ_1 ((1,1), y) \succ_1 ((0,2), y) \sim_1 ((2,0), n) \sim_1 ((1,1), n) \sim_1 ((0,2), n)$
- $((0,2), y) \succ_2 ((1,1), y) \succ_2 ((2,0), y) \sim_2 ((0,2), n) \sim_2 ((1,1), n) \sim_2 ((2,0), n)$

1.6 Game Tree

A convenient representation of a game in extensive form is by the use of a tree. Each node in this tree is marked by two labels: one, a history $h = (a^k)_{k=1,\ldots,m}$ to which that node correponds, and two, the player $P(h)$ who makes the move if the game has a history h. Edges emanating from a node with history h are incident on nodes that correspond to histories $h' = (a^k)_{k=1,\ldots,m+1}$ where a^{m+1} is a move of the player $P(h)$. Finally, no edges emanate from the leaf nodes and they are labeled with a pay-off vector $\vec{P} = (p^1, \ldots, p^{|N|})$.

Note that in this tree, a path from the root node to any other node represents a history. In particular, the game will reach the state corresponding to a node x if players follow edges corresponding to the unique path from the root node to x. Further, observe that the progress of the game can be traced out as a unique path from the root node to one of the leaf nodes. Every move of a player translates into following an arc from one node to another node.

Figure 2 represents the game tree for the example given above.

2 Solving a Game in Extensive Form

Given a game in extensive form along with player pay-offs, assuming that the players are rational and aware of each other's rationality, how will the game end? Which player will win in such a case? In this section, we are interested in developing solution concepts for games in extensive form. We begin by defining strategy in such a game, and then define *Nash equilibria* and *Subgame Perfect equilibrium* in this context.

2.1 Strategy

A strategy of a player in an extensive game is a plan that specifies the action chosen by the player for *every* history after which it is his turn to move. Hence, the strategy function f_i for player i is a function of the following form:

$$f : H_i \to A_i \qquad H_i = \{h \in H | P(h) = i\}$$

where H_i is the set of histories where it is player i's move, and A_i is the action space for player i. An important point here is that a strategy specifies the action chosen by a player for *every* history after which it is his turn to move, *even for histories that, if the strategy is followed, are never reached.* In other words, for each node in which it is player i's move, his strategy should choose one of the edges emanating from that node.
So, for this example, a strategy of player 1 has to choose one of the edges from $\{(2, 0), (1, 1), (0, 2)\}$.
A strategy of player 2 has to choose y or n , corresponding to each of the 3 possible nodes in which it is his move and can be represented by a triple like (y, n, y) which implies that he chooses y,n and y if he is at the left,middle or right node respectively.
A *strategy profile* is a tuple consisting of a strategy of each player. In game tree terms, a strategy profile associates every node with an edge emanating from it. A possible strategy profile can be

- $f_1(\phi) = (1, 1)$
- $f_2((2, 0)) = y, f_2((1, 1)) = y, f_2((0, 2)) = n$

We can use $\{(1, 1), yyn\}$ to conveniently represent this strategy profile.

2.2 Nash Equilibrium

Definition A **Nash equilibrium of an extensive game with perfect information** $< N, H, P, (\geq_i) >$ is a strategy profile s^* such that for every player $i \in N$ we have

$$O(s*_{-i}, s_i*) \geq_i O(s*_{-i}, s_i)\text{for every strategy}s_i \text{ of player}i.$$

Using this definition, the game in the example above has 9 Nash equilibria - $\{(1, 1), nyy\}$ and $\{(1, 1), nyn\}$ are two of them.Can you figure out the rest? Note that the total number of strategy profiles is $2 \times 8 = 16$.
One of the reasons why this game has so many Nash equilibria is that the players are

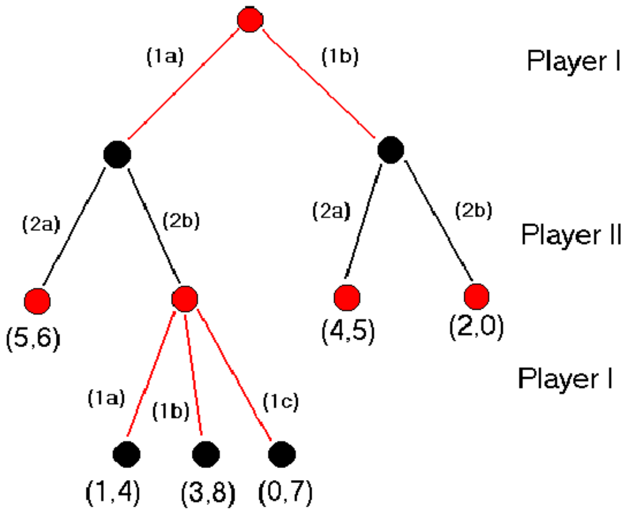

Figure 3: Another extensive game

indifferent between strategies at some stage. For instance for the node $(2, 0)$, player 2 does not prefer the move y over n as they both lead to the same payoff 0 for him. Games like *PickStick* are more *interesting* in the sense that a certain strategy is always preferable over (or not preferable) over the other and we will deal with such games only(Payoffs which ensure such a property are called *generic payoffs*). Consider another extensive game as modelled in Figure 3. This game has four Nash equilibria. One of them is given by

- $f_1(\phi) = 1a,\ f_1((1a, 2b)) = 1b$
- $f_2((1a)) = 2b,\ f_2((1b)) = 2b$

Can you figure out the others? We will discuss the other Nash equilibria in the remaining sections.

2.3 Zermelo's Algorithm

We will now describe Zermelo's Algorithm that will help us develop the intuitive basis for *subgame perfect equilibria*. Suppose x is a node in the game tree. Further, let it be i's turn when the game has reached node x. Zermelo's Algorithm associates with the node x, a pay-off vector $\vec{p_x}$ (where $p_x[j]$ is the pay-off of player j, $j \in [n]$) with the property that player i (who has to move when the game is at node x) can ensure a minimum pay-off of $p_x[i]$.

The labeling of the nodes with these pay-off vectors occurs in the following manner:

1. All leaf nodes correspond to outcomes of the game. Hence, pay-off vectors for leaf nodes are part of the specification of the game.

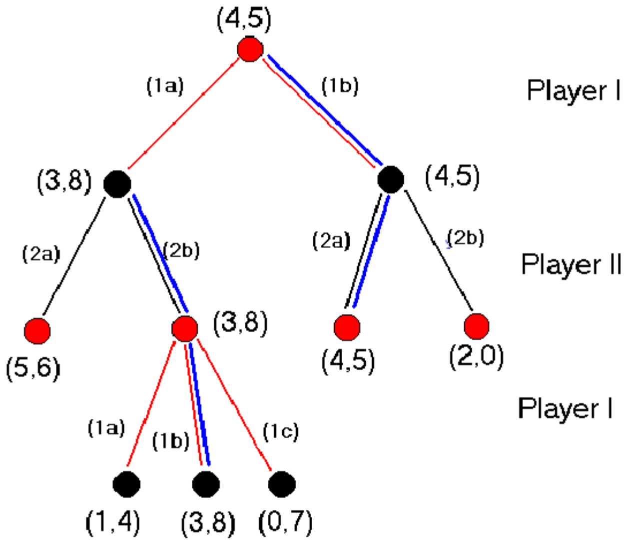

Figure 4: Execution of the Zermelo's Algorithm.

2. An internal node x, with player i to move, is marked with a pay-off vector $\vec{p}_x$ that is the pay-off vector of that child which maximizes player i's own payoff. Formally, $\vec{p}_x = \vec{p}_y$ such that y is a child of x and $p_y[i] = \max\{p_z[i] | z$ is a child of $x\}$.

3. Call the edges between nodes that share the same pay-off labels as *blue edges*.

For example, consider the game tree in Figure 3. The red nodes imply player 1's turn and the blue nodes signify player 2's turn. Red arcs correspond to moves made by player 1 and black arcs correspond to moves made by player 2. For each of the leaves, a pair of values is given that is the pay-off to player 1 followed by player 2.

In this game, the Zermelo's tree assigns labels to internal nodes as shown in Figure 4. The root node is labeled with a pay-off vector $(4, 5)$. This payoff can be achieved simply if the players follow the strategies corresponding to the unique path of blue edges from the root to a leaf node.

2.4 Subgame Perfect Equilibrium

After running Zermelo's Algorithm, corresponding to each node, we obtain a *pay-off vector*, and also, all internal nodes have exactly one *blue edge* that emanates from them. These edges constitute a strategy profile. Note that this strategy profile will constitute a Nash equilibrium. This Nash equilibrium is called the *Subgame Perfect equilibrium.*It can be thought of as a more credible Nash equilibrium. The payoff vector at the root node is the payoff associated with this equilibrium.

We need to define a subgame before we can formally define a Subgame Perfect equilibrium. A subgame of an extensive game with game tree $\mathcal{T}$ is a game modelled by the subtree under a particular node x, ie. a subtree with x as the root and all its *descendants* as the other nodes. So the number of subgames induced by a game is equal to the number of nontermminal nodes.

Definition A strategy profile s^* is a *Subgame Perfect Equilibrium* of a game if it is a Nash equilibrium of every subgame of the game.

Does Subgame Perfect Equilibrium, guarantee the best payoff for both players? Consider the example in Figure 4. Note that there is an outcome of $(5, 6)$ that would be better for both the players as compared to the Subgame Perfect Equilibrium. One can find a strategy profile which is at Nash equilibrium and has a payoff of $(5, 6)$. Consider

- $f_1(\phi) = 1a,\ f_1((1a, 2b)) = 1a$
- $f_2((1a)) = 2a,\ f_2((1b)) = 2a$

Neither player can improve his uitlity and the payoff is $(5, 6)$. Infact you get another Nash equilibrium with the same payoff if you replace $f_2((1b)) = 2a$ with $f_2((1b)) = 2b$.
However, in a zero sum game, since there cannot be an outcome where both players may improve their pay-offs (if one increases his pay-off, then the pay-off of the other player, by definition, must necessarily decrease) as compared to the Subgame Perfect Equilibrium. It is left as an exercise to the reader to show that Subgame Perfect equilibrium is also the only Nash equilibrium in a zero sum game.

3 Theory of Utility

In the remainder of these scribes, we introduce the theory of utility that will continue to be the subject of the next couple of lectures. The model we have studied assumes that each decision-maker is "rational" in the sense that he is aware of his alternatives, forms expectations about any unknowns, has clear preferences and chooses his action deliberately after some process of optimization.

Sometimes the decision-maker's preferences are specified by giving a *utility function* $U : \Omega \to \mathcal{R}$ that maps the set of outcomes to real numbers conserving the *preference relation* (a complete transitive reflexive binary relation) $\succsim$ on the set of outcomes. Here, we must have $U(x) \geq U(y)$ if and only if $x \succsim y$.

Consider the famous *gold and water* paradox. The utility of gold is much lower than water, however, the value of gold is much higher than that of water. Why is it the case that a commodity of such low utility is valued so much higher? Informally, one gets the feeling that since the relative abundance of water is much higher than that of gold, the factor of availability must play in somewhere in value-determination. The theory of utility seeks to answer such questions in a formal model.

Another paradox is the following. Suppose I ask you to choose from one of the following options: 1. I give you Rs. 10 with probability 1, or 2. I give you Rs. 70 with probability of 0.5. Which option are you likely to choose? Now consider a scaled variation of this puzzle. I now offer you Rs. 10 crore with probability 1, or on the other hand Rs. 70 crore with probability of 0.5. Which option are you likely to choose now?