Game Theory

Lecture Notes By

Y. Narahari

Department of Computer Science and Automation

Indian Institute of Science

Bangalore, India

March 2007

Chapter 2: Extensive Form Games

Note: *This is a only a draft version, so there could be flaws. If you find any errors, please do send email to* hari@csa.iisc.ernet.in. *A more thorough version would be available soon in this space.*

Extensive Form of a Game

- The extensive form of a game is a richly structured way to describe game situations.
  - It was first proposed by von Neumann and Morgenstern (1944).
  - The version developed by Kuhn (1953) is now a de facto standard.
- The extensive form captures
  - who moves when
  - what actions each player can take
  - what players know when they move
  - what the outcomes are as a function of the actions
  - payoffs of players from each outcome.

Example: Matching Pennies with observation (Version B of Matching Pennies). Here, player 1 first puts down his rupee coin heads up or tails up. Player 2 sees the outcome and then puts down his rupee coin heads up or tails up. The payoffs are as before. The extensive form of this is represented by the game tree in Figure 2.

- The nodes are classified into
  - root node (initial decision node)
  - decision nodes
  - terminal nodes

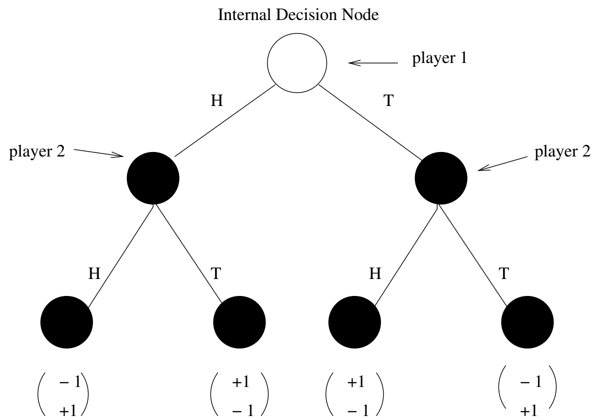

Figure 1: Game tree for matching pennies with observation

- Each possible sequence of events that could occur in the game is represented by a path of branches from the root to one of the terminal nodes.
- When the game actually happens, the path that represents the actual sequence of events is called the *path of play*.
- The goals of game theoretic analysis is to predict the path of play.
- Each decision node is labeled with the player who takes a decision at that node. The branches outgoing at the decision node are labeled with the possible actions the player may choose.
- The terminal nodes are labeled with the payoffs for the players.
- When a given position can be reached through different sequences, each of these sequences is depicted separately in the tree.
- Nodes represent not only the current position but also how it was reached.

Example: Matching Pennies without observation (Version C)

Here player 1 moves first and puts down his rupee coin heads up or tails up. Player 2 does not observe the outcome. He then moves putting down his rupee coin heads up or tails up. The game tree is as follows.

- The game has the same structure as for version B, however the two decision nodes corresponding to player 2 have been encircled.
- A set of encircled decision nodes is called an *information set*. When play reaches one of the decision nodes in an information set and it is the turn of that player to move, the player does not know which of these nodes he is actually at. The reason for this is that the player does not observe something about what has previously happened in the game.
- Here, since player 2 has not observed the outcome of player 1's move, he cannot say at which node he is actually in.

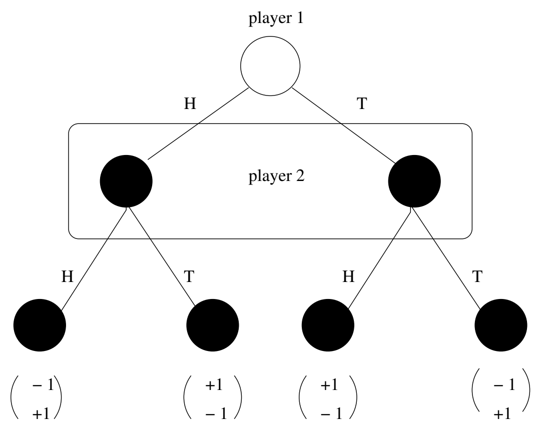

Figure 2: Game tree for matching pennies without observation

- An information set of a player thus gives a set of his decision nodes which are indistinguishable for him.
- A listing of all the information sets of a player gives a listing of all the possible distinguishable events or circumstances in which he might be called upon to move.
- At every node within an information set, a player must have the same set of possible actions.

Definition: *Games with Perfect Information* are those in which all the information sets are singleton decision nodes.

- This implies that every player is able to observe *all the previous moves* made in the game. He precisely knows where he is and how he has reached this node.

  If at least one information set of at least one player has two or more elements, the game is said to be of *imperfect* information.

Example: Version B of Matching Pennies (with observation case) is a game of perfect information. Version C (without observation case) is a game of imperfect information.

Example: Version A of Matching Pennies (simultaneous moves):

- The game tree of version C provides a representation of the game
- A game tree, similar to that of version C, but with player 1 and player 2 swapped also represents the game
- The order of play is immaterial

Example: **Matching Pennies with chance Node (version D)**: Here the two players first toss a coin to decide who moves first. Then it proceeds as version C. The game tree is shown below.

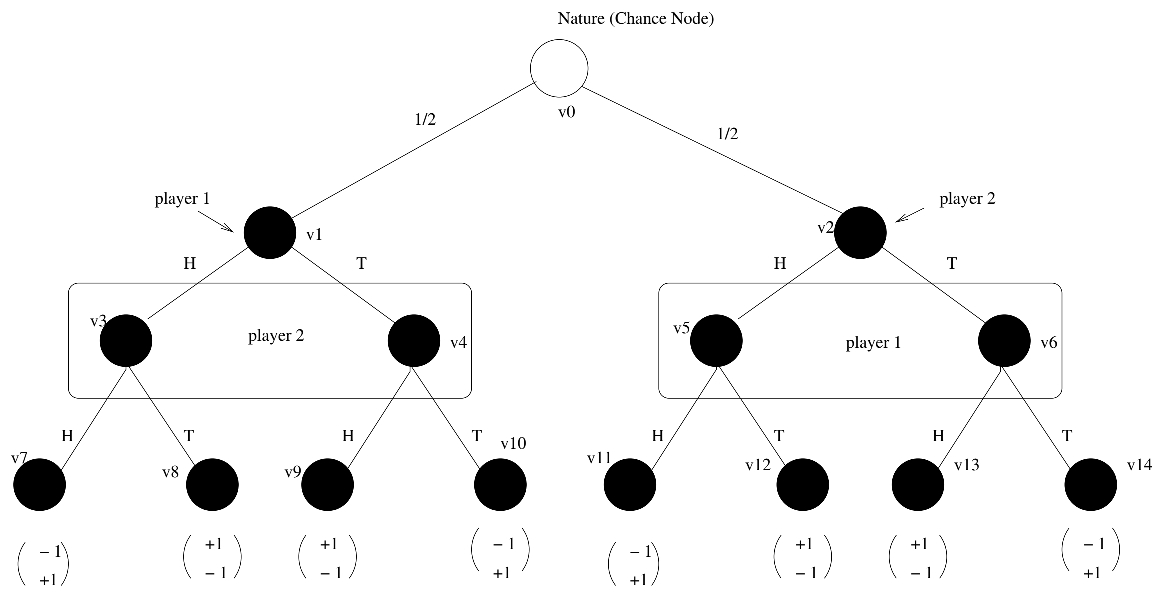

Figure 3: Game tree for matching pennies with a chance node

- The initial node is called a *chance node* or a move of *nature*. In this case, it has two branches each with probability $\frac{1}{2}$.
- The nature is regarded as an additional player; nature is assumed to play its actions with fixed probabilities. This introduces an element of chance into the game.
- We are now ready to define a game in extensive form.
- An extensive form game is the tuple $G = (N, V, A, p(\cdot), \alpha(\cdot),\ I, I(\cdot),\ \rho(\cdot))$

  A game represented in extensive form is the above together with utility functions.

$$G_E = (N, V, A, p(\cdot), \alpha(\cdot), IS, I(\cdot), \delta(\cdot), \rho(\cdot), (u_i))$$

- $N = \{1, 2, \ldots, n\}$ set of players; $N^+ = \{0, 1, 2, \ldots, n\}$ set of players including the nature
- V = set of nodes or vertices; x_0 = is the initial node or the root node
- A = set of actions
- p is called the immediate predecessor (or parent) function; p: $V \backslash \{x_0\} \rightarrow V$. $p(x)$ = parent of x or immediate predecessor of x; $s(x)$ = set of immediate successors of x = set of children of x = $\bar{p}^1(x)$
- T = is called the set of terminal nodes. $T = \{x \in V : s(x) = \phi\}$. $x \in V \backslash T$ are called *decision nodes*.
- α, the *action mapping* is defined by $\alpha : V \backslash \{x_0\} \rightarrow A$. $\alpha(x)$ = action that leads from $p(x)$ to x. Note that $x_1, x_2 \in s(x), x_1 \neq x_2 \Rightarrow \alpha(x_1) \neq \alpha(x_2)$.

  Choices for node x are given by $c(x) = \{a \in A : a = \alpha(x') \text{ for some } x' \in s(x)\}$.

- IS: Collection of information sets; we associate an information set to every node through the function I: $V \rightarrow IS$. IS is a partition of V. $I(x)$ gives the information set associated with x. All nodes in a single information set have the same action choices. That is, $I(x) = I(x') \Rightarrow c(x) = c(x') \quad \forall x, x' \in V$. Set of all actions associated with an information set $J \in IS$ is defined as:

$$\begin{aligned} C(J) &= \{a \in A : a \in c(x) \text{ for some } x \in J \\ &= \bigcup_{x \in J} c(x) \end{aligned}$$

- $\delta : IS \rightarrow \{0, 1, \ldots, n\}$ assigns an information set to a player. $\delta(J) =$ represents the player to whom the information set J belongs. $IS_i =$ set of information sets of player i, $i = 0, 1, \ldots, n$. This is equal to $\{J \in IS : \delta(J) = i\}$

- The function ρ assigns probabilities to actions at the chance node.

$$\rho : IS_0 \times A \rightarrow [0, 1]$$

$$\rho(J, a) = 0 \qquad \text{for } J \in IS_0 \text{ and } a \in A, \ a \notin C(J)$$

$$\sum_{a \in C(J)} \rho(J, a) = 1 \quad \forall J \in IS_0$$

- $u_1, u_2, \ldots, u_n : T \rightarrow R$ are utility functions.

Games with Perfect Recall

- It is generally assumed that extensive form games satisfy a condition called *perfect recall*.
- This condition asserts that whenever a player moves, he remembers all the information that he knew earlier in the game, including all of his own past moves.
- That is if decision nodes y and z of player i are indistinguishable to him (that y and z are in the same information set of player i), then for any earlier decision node and move that player i recalls at y, there must be an indistinguishable decision mode and move that he recalls at z.

Technically, an extensive game is said to be of *perfect recall* if the following two conditions hold:

1. For any two decision nodes y and z,

   $I(y) = I(z) \Rightarrow y$ is neither a predecessor of z nor a successor of z

2. If y and z are two decision nodes of player i which are in the same information set and if x is a predecessor of y (not necessarily an immediate one) that is also in one of the information sets of player i with $a \in A$ being the action $I(x)$ on the path to y, then there must be a predecessor node to z that is an element of $I(x)$, say x', and the action at x' that is on the path to z must also be a.

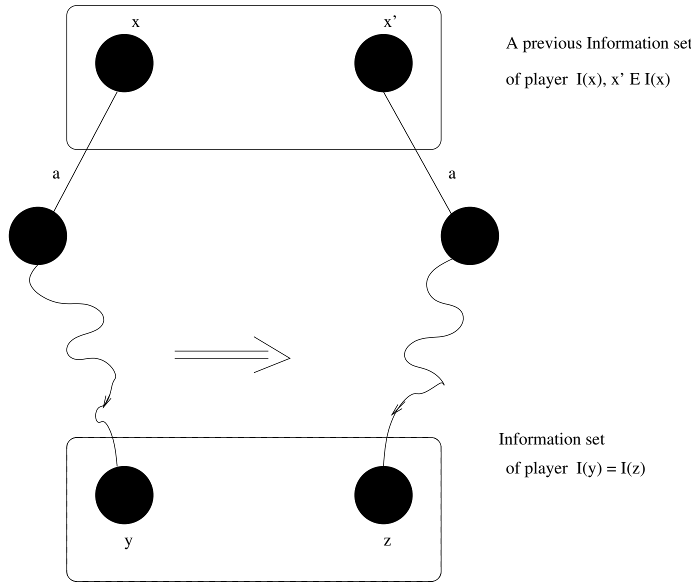

Figure 4: A game tree to illustrate games with perfect recall

- All the examples discussed so far are games of perfect recall the same thing at z.
- Note that a game of perfect recall may be a game with perfect information or a game with imperfect information.

Example 1 of a game without perfect recall: As the game progresses, player 2 forgets a move by player 1 that he once knew.

- At node 2, player 2 knew that player 1's action is l, whereas at node 9, he does not remember whether player 1 played l or r at node 1.

Example 2 of a game without perfect recall: In this case, player 1 at nodes 4,5,6,7 (same information set) has forgotten what he himself has played at node 1.

Note: It is clear that games with perfect information are always games with perfect recall but not vice-versa.

Note: A finite game is one in which the number of nodes is finite. This will immediately imply that

1. The number of players is finite
2. Each player has a finite number of possible actions for each information set
3. The game ends after a finite number of moves

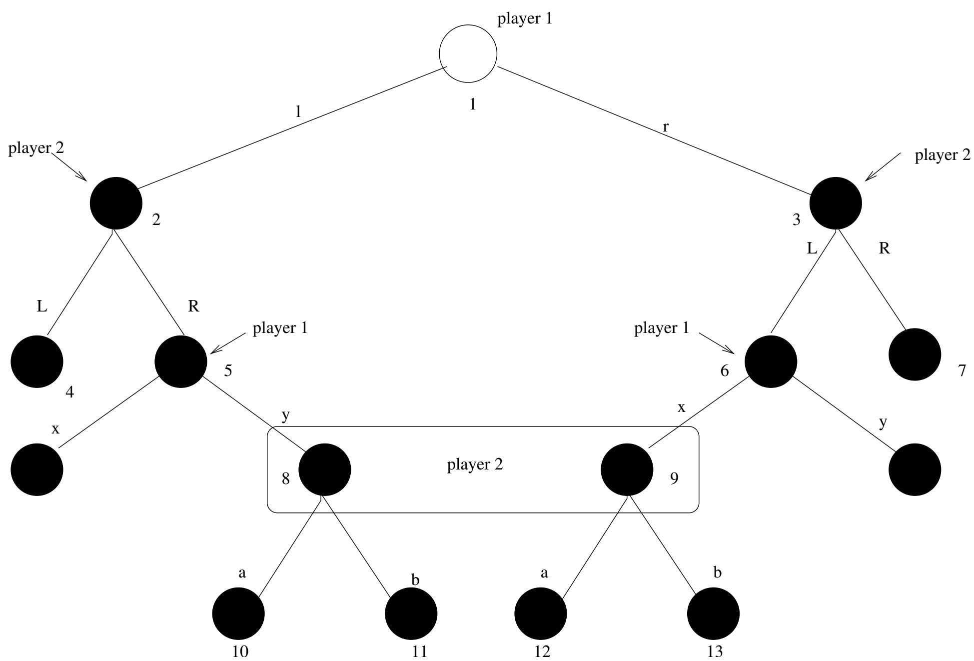

Figure 5: An Example of a game without perfect recall

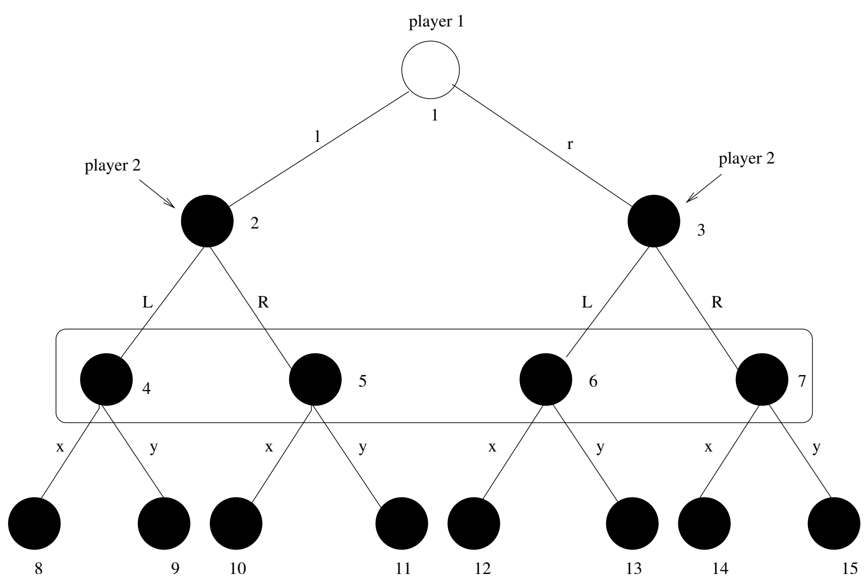

Figure 6: An example of a game without perfect recall