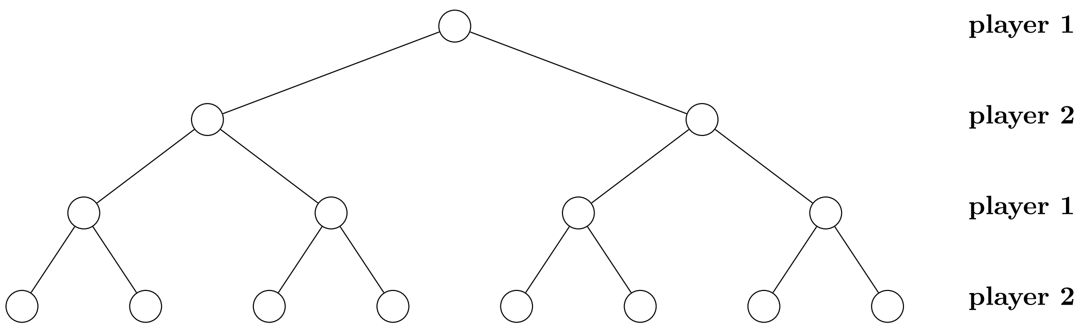
CS 6840 Algorithmic Game Theory November 22, 2024

Lecture 36: Extensive Form Games

Instructor: Eva Tardos *Scribe: Henry Rogers*

Extensive Form Games

Extensive form games have a Game Tree, which represents all possible sequences of moves by the players, and so all possible states that are possible in the game. Players each make moves in an alternating fashion, and each move sends the game into a new state, from which the next player can make a move. Below is a simple game where each player has 2 moves at each turn. We can imagine that at each leaf node there is a payoff vector that determines how much each player wins in that state when the game ends.



A popular game that follows this format is Chess. At every move, there is a known fixed set of possible moves, and each move sends the game into a particular state, from which the next player makes their move.

Perfect Information Games

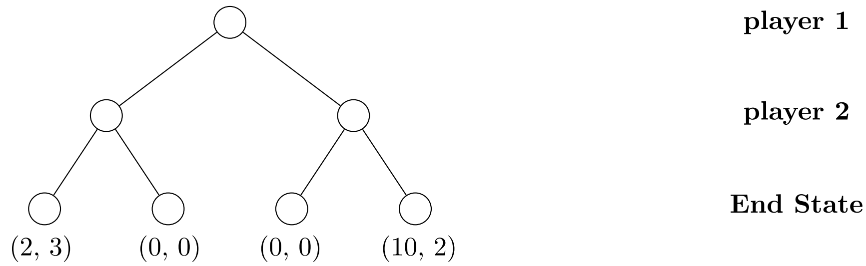
It is important to note that in these games, both players know all the possible moves, and once a player makes a move, they know all possible moves their opponent could make after. This makes these games **Perfect Information Games**. Chess is a perfect information game, and as such it is solvable. This means we could in theory map the entire state space of the game, and know the optimal move for any state. We would do this by working backwards from the leaf nodes in which we win, since we can know what each player should do at each step. However the state space of chess is so large that this is not feasible within the current constraints of computation.

It is also worth noting that even if there is randomness in the tree, as long as we know the distribution of this randomness, we can still find our optimal sequence in expectation.

Sub-Game Perfect Nash

The ability to fully solve games of this type creates a Nash Equilibrium, and in fact creates a Nash at every point in the sub tree for each choice. This is called a **Sub-Game Perfect Nash**: After any prefix, you are still at a Nash for the remaining sub-game.

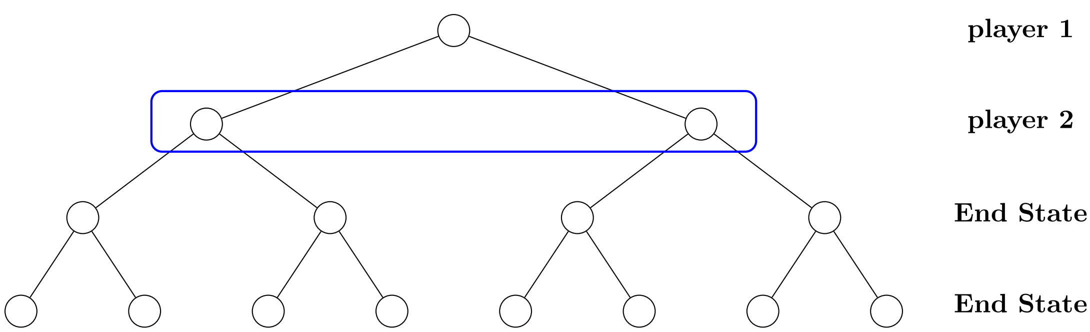
Non Sub-game Perfect Example:



Consider the above game tree, where player 1 and player 2 each make one move, and end in one of the leaf states, receiving value according to the pictured payoff vector. So in the far right node, player 1 and player 2 receive value 10 and 2 respectively. Clearly this far right nodes is best for player 1, and far left is best for player 2. The middle two nodes are the worst case for both. The obvious Nash in this game if everyone is rational is that player 1 goes Right, and player 2 then goes Right because 2 is better than 0 for them.

But now consider if player 2 commits to going Left no matter what. Then if player 1 goes right, both get 0. Player 1 is aware of this fact, they will be forced to go Left, since they would rather have 2 than 0. So we end up at the far left node that is best for player 2, and only ok for player 1. This is still a Nash, but it is not Sub-Game perfect.

Imperfect Information Games



The blue box is the **Information Set** for player 2 at this turn. This depends on the previous moves, and you can assume that each player perfectly remembers everything that has happened so far. But in an Imperfect Information Game, it is not the case that each player knows everything the other player might have at every turn. Another difference here is that there may be randomness that is not observed by all players. For example a player can draw a random card into their hand, and the other player does not know the outcome of this draw.

These differences mean these games are not solvable in the simple bottom-up approach that works for perfect information games, since when a player makes a move, they don't know all the options of the other player up to that point, and so it is harder to map out all possible states of the game.

We can convert these games to simultaneous move games that are no longer extensive. What this means is at every node, you decide in advance which way you are going to go. This is a pure strategy that is described by a specific choice for each information set. It is also worth noting that you could randomize these choices, but that doesn't change the conversion to simultaneity.

We then have for our number of pure strategies:

- Decision at every n, info sets for P.
- k options at each set.
- So total k^n pure strategies since k choices at each decision point.

So this will became of classical game with two players but with k^n strategies each, which is not polynomial in the description of the game (which we take to be the grade tree).

We will see that for 2 person zero sum games this can still be solved, and a Coarse Correlated Equilibrium can be solved for any game of this sort.

Here is a different way of describing our strategy for each information set.

We have for each info set v, there is a probability x_{vi} that the player chooses option i from info set v. To ensure this is a valid probability distribution we just need to make sure

$$x_{vi} \geq 0$$
$$\sum_i x_{vi} = 1$$

We then have

- kn x variables for player 1
- a similar set of y variables for player 2

Probability of reaching a leaf can be represented by $x \cdot y \cdot x \cdot \ldots$

For the necessary sequence of events to reach that leaf.

We would like to be able to define a matrix A such that our payoff is defined by

$$x^T A y$$

but we cannot do this immediately with our current variables. This is because with the variables x and y, to express the probability that we reach a leaf with some payoff we must take a product of all the variables on the path from the root to the leaf, so not of this form.

To get our A matrix, we define new variables that will let us define strategies in a way that allows us to express payoffs in the $x^T A y$ form.

$$z_{vi} = \prod x_{wj} \quad \text{for path from root to } v_i$$
$$z_w = \prod x_{wi} \quad \text{for path from root to } w.$$

This will be continued next lecture.