Dynamic games of complete information

IE 5545: Decision Analysis
Prof. Krishnamurthy Iyer*

May 2, 2026

Having developed a framework to describe and analyze static games of complete information, we are interested in seeing how one can extend the tools and concepts to *dynamic games with complete information*. These are games where the timing of players' actions is important — either players move sequentially while observing the actions of other players, or the game has multiple stages where a player might be required multiple times during the course of the interaction. For example, consider games such as Chess, or two firms that are engaged in repeated Cournot competition over many time periods. Most interesting games of practical importance typically fall into this category, and hence we would like to understand whether we can meaningfully talk about concepts similar to Nash equilibria for such games.

We start our analysis by extending our framework to represent dynamic games. Note that to describe a dynamic game, in addition to describing the set of players, their payoff functions over the set of outcomes, and their knowledge about rationality and the structure of the game (as in the static game), we also need to describe the (1) timing of the moves, (2) the set of actions available to each player at the time of their move, and (3) the information the players have about the past play at the time of their move. To represent these additional components of a dynamic game, we use the concept of a *game tree*, which we describe below.

1 Game trees

Before we describe game trees, we will differentiate between two types of dynamic games of complete information.

In games such as tic-tac-toe, or chess, when it is time for a player to move, she knows all the past actions that have been taken in the game until then. In other words, everything that has happened in the game until her move that is relevant to her decision is known to the player. We refer to such games as *games of perfect information*. Most turn-based games fall into this category.

On the other hand, consider rock-paper-scissors. While we have viewed it as a static game of complete information, another description of the game is one where player I chooses first among rock, paper & scissors, and then player II chooses *without observing what player I has chosen*. With this description, rock-paper-scissors can be viewed as a dynamic game, where player II does not know everything that has happened in the game until her move. Such games are referred to as *games of imperfect information*. The above argument in fact shows that all static games of complete information are also dynamic games of imperfect information — since the players have to make their choice simultaneously. But so are any *repetitions* of such static games (for example, the repeated Cournot competition alluded to earlier or the best-of-three rock-paper-scissors).

We will first begin by describing the game tree for games of perfect information, and then we will extend the notion to games of imperfect information.

1.1 Game trees for perfect information games

Restricting our attention to games of perfect information, we define a game tree as a rooted-tree with certain additional information regarding the players, their moves and the payoffs. We define these in detail next.

Recall that a tree is a connected graph with no cycles. A *rooted-tree* is a tree with a distinguished vertex (or node), referred to as its root.

A game tree is a rooted tree with the following additional information:

1. **Leaves:** Each leaf of the tree is associated with an outcome, and the corresponding payoffs for each player. (Recall that a leaf of a tree is a vertex of degree one). We refer to the leaf vertices as *terminal nodes*. The interpretation is that if the game "reaches" a terminal node, the game ends, and the outcome of the game is the outcome associated with that terminal node.

   We denote the set of terminal nodes by $\mathbb{Z}$. For each $x \in \mathbb{Z}$, let $\pi_i(x)$ denote the payoff received by player i in the associated outcome.

2. **Non-terminal nodes:** Each non-terminal node is of two types:

   (a) a decision node, in which case the node is associated to a unique player;

   (b) or a chance node, in which case the node is associated with (a fictional player called) nature.

   The interpretation here is that if the game "reaches" a decision node, the player associated with that decision node makes a move (i.e. plays an action). Similarly, if the game "reaches" a chance node, then the game proceeds randomly according to a prespeficied distribution (for example, through a roll of a die, or a toss of a coin).

   For any decision node x, we let $i(x)$ denote the player associated with x. Similarly, if x is a chance node, we let $i(x) = \mathsf{nature}$.

3. **Branches:** For any non-terminal node x, let $A(x)$ denote the set of all branches at x that are directed away from the root. Each branch in $A(x)$ corresponds to one possible way in which the game can proceed from x.

   In particular, for any decision node x, each branch in $A(x)$ corresponds to an action available to the player $i(x)$ at x. Thus, for a decision node, the set $A(x)$ denotes the set of all actions available to the player $i(x)$ at x.

   Similarly, at each chance node x, each branch in $A(x)$ corresponds to one possible way in which the randomness (due to nature) can get resolved. In this case, we also associate with each branch in $A(x)$ the probability with which that branch will be choosen for the game to proceed along. Let $p(x)$ denote the distribution over $A(x)$ with which nature resolves the randomness — for any $a \in A(x)$, $p(x, a)$ denotes the probability with which a is chosen at node x. Note that $\sum_{a \in A(x)} p(x, a) = 1$.

4. **Timing:** The timing of the game is as follows. The game begins at the node $x = \mathsf{root}$. Whenever the game reaches a decision node x, the player $i(x)$ chooses an action $a \in A(x)$, and the game proceeds along the branch a to x's neighboring node. Whenever the game reaches a chance node x, nature chooses a branch $a \in A(x)$ *independently* according to distribution

$p(x)$, and the game proceeds along a to x's neighboring node. Finally, when the game reaches a terminal node $x \in \mathbb{Z}$, the game ends with outcome associated with x, and each player i receives a payoff $\pi_i(x)$.

As an aside, the reason we use trees to represent a dynamic game of perfect information is that in such a game, a player (or the play) can never return to the same point in the game. This is because, since all the information about the history of the game is known before each move, if the game reaches the same setting for the second time, the player about to move knows that the game was at that setting before. However, this was not true the first time the game reached that point. This extra information might be useful for that player in deciding her action, and hence we must not represent the two instances with the same decision node, but instead by two different decision nodes. This implies that no decision node can be reached twice, and hence there can be no cycles in the graph representing the game. Thus, the graph representing the game must be a tree.

Example (Ultimatum game)**.** Two players (I and II) engage in a negotiation game. Player I is offered \$10 that she has to split with player II. The negotiation is as follows: player I offers player II an amount \$$x$, where $x \in \{0, 1, \cdots, 10\}$. If player II accepts the offer, then the game ends, player II receives \$$x$, and player I gets to keep the remaining \$$(10 - x)$. If player II rejects the offer, then neither player receives any amount. (In particular, player I gets \$0 if player II rejects the offer.) We assume that both players only care about how much money they receive, and prefer to maximize this quantity.

In this case, the set of players is $\{\mathrm{I}, \mathrm{II}\}$. Any outcome of the game can be represented as (x, y), where x is the amount player I receives and y is the amount player II receives in that outcome. The set of outcomes can be represented as $\{(x, 10 - x) : x = 0, 1, 2 \ldots, 10.\} \cup \{(0, 0)\}$. The payoff functions of the two players can be taken as $\pi_{\mathrm{I}}(x, y) = x$ and $\pi_{\mathrm{II}}(x, y) = y$. The game tree is shown in Figure 1. One possible way the game could proceed is player I makes an offer of \$2, and player

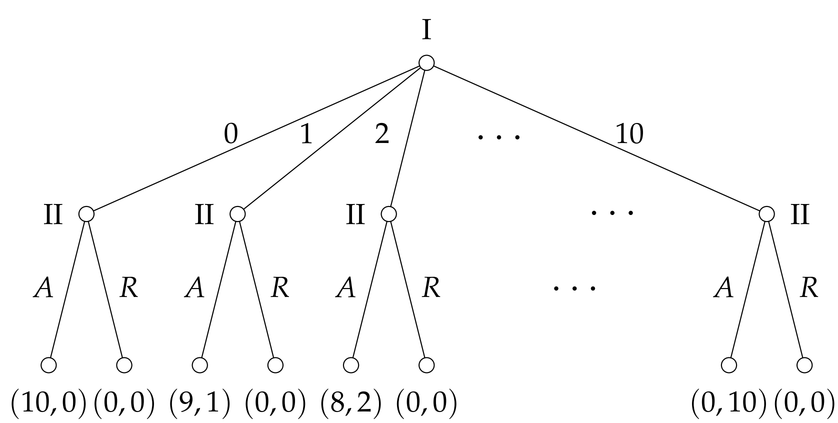

Figure 1: Ultimatum game.

II accepts it. In this case, the outcome is $(8, 2)$, with corresponding payoffs to the two players.

Note that, unlike static games, here there is a difference between actions and strategies for a player. In particular, to strategize, the player II has to decide whether to accept or reject the offer for each possible offer player I can make. Namely, player II has to make a decision at each of her decision node. Thus, while the set of strategies for player I is same as the set of actions available at the root (e.g., make an offer of \$5), a *strategy* for player II needs to specify what she will do at each of her decision node (e.g., accept only those offers that are strictly greater than \$2). □

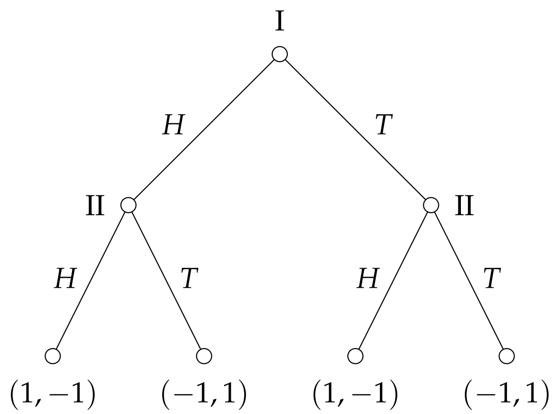

Figure 2: Trivial matching pennies.

Generalizing the above observation, we have the following definition of a pure strategy for games with perfect information.

Definition 1. *In any game of perfect information, a* pure strategy *for player i is a function that takes as input a decision node associated with player i, and gives as output an action available at that node. Formally, a pure strategy s_i specifies an action $s_i(x) \in A(x)$ for each decision node x such that $i(x) = i$.*

Example (Ultimatum game (contd.))**.** For example, for the ultimatum game, a pure strategy for player II where she accepts any offer strictly greater than \$4, and reject any other offer, can be represented as a function $s(x) = A$ if $x > 4$, and $s(x) = R$ otherwise. Note that we can also represent it as a vector, namely $[R, R, R, R, R, A, A, A, A, A, A]$.

Also, observe that in the ultimatum game, player I has 11 pure strategies, whereas player II has $2^{11} = 2048$ pure strategies. This suggests that a game tree can compactly represent games where players have lot of pure strategies (unlike a strategic form we studied before).

1.2 Games of imperfect information

Consider a modified version of matching pennies, called trivial matching pennies, where player 1 moves first and chooses between Heads (H) or Tails (T), and subsequently, after seeing player 1's choice, player 2 moves second and also chooses between Heads (H) or Tails (T). The payoffs are same as in the matching pennies game: a match yields a payoff of 1 to player 1 and a payoff of -1 to player 2, whereas a non-match yields a payoff of -1 to player 1 and a payoff of 1 to player 2. The game tree for trivial matching pennies is shown in Figure 2.

In this game, player 1 has two strategies, namely H or T. Player 2 on the other hand has four strategies, as follows:

1. (H, H) — play H if player 1 plays H, and play H if player 1 plays T;
2. (T, H) — play T if player 1 plays H, and play H if player 1 plays T;
3. (H, T) — play H if player 1 plays H, and play T if player 1 plays T;
4. (T, T) — play T if player 1 plays H, and play T if player 1 plays T.

How does matching pennies differ from trivial matching pennies? In matching pennies, player 2 does not observe player 1's actions. Due to this, player 2 cannot play different actions in her

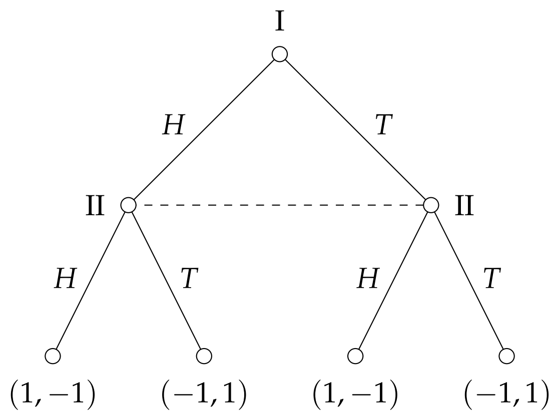

Figure 3: Matching pennies.

two decision nodes. In other words, strategies (H, T) and (T, H) are not available for player 2 in matching pennies, and she is forced to play the same action in each of her two decision nodes, choosing between strategies (H, H) and (T, T).

We can represent this restriction of player 2's choice by saying that the two decision nodes of player 2 belong to the same *information set*, meaning player 2 has the same information in the two decision nodes and has to play the same action. We represent this by drawing a blob around the two decision nodes, or joining the two decision nodes by a dashed line as in Figure 3.

Formally, we define the information sets as follows:

Definition 2 (Information sets)**.** *An information set I_i of player i is a subset of decision nodes, all associated with player i, such that player i has the same information at each of the decision nodes in the set. Consequently, the player is restricted to playing the same action at each decision node $x \in I_i$.*

The concept of information set captures the fact that a player cannot differentiate between her decision nodes belonging to the same information set. What properties must all the decision nodes belonging to the same information set satisfy?

1. First, we must have that all decision nodes belonging to the same information set must be associated with the same player. That is, if $x, x' \in I$, then $i(x) = i(x')$. This is necessary, because if two nodes are associated with different players, they are free to choose different actions at these nodes, and our requirement that the same action is chosen at each decision node in an information set cannot hold.

2. Second, at each decision node belonging to the same information set, the set of available actions must be the same. In other words, we must have that if $x, x' \in I$, then $A(x) = A(x')$. Otherwise, the player can just check which actions are available to her, and differentiate between the nodes. Thus, not only do we require that same actions be played at different decision nodes belonging to the same information set, we require the set of available actions to be the same.

Note that if every information sets contains only one decision node, then the players are not restricted in their choice of actions, and they have the same set of pure strategies as in the game of perfect information. Thus, we can define a game of perfect information as one in which all information sets are singleton sets.

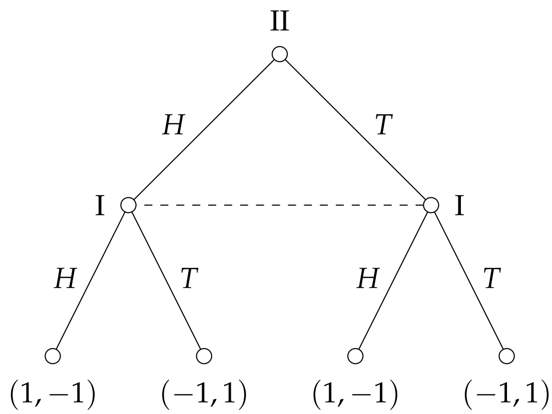

Figure 4: Matching pennies (Different game tree).

2 Extensive form

Taken together, we can define the *extensive form* of a dynamic game of complete information as follows:

Definition 3 (Extensive Form)**.** *An extensive form of a dynamic game of complete information specifies the following:*

1. *Set of Players* $\{1, \cdots, N\}$*;*
2. *Payoff functions* π_i *for each player* i*;*
3. *Game tree, with all the information sets.*

Note that a game can have multiple extensive forms. For example, matching pennies can be depicted by second game tree as in Figure 4.

3 Strategies

Given the definition of the extensive form, we can formally define the notion of a pure strategy.

Definition 4. *A pure strategy* s_i *for a player* i *is a rule that specifies an action (from the set of available actions) at each of her information sets.*

Note that in a game of perfect information, since each decision node is in a singleton information set, this is same as saying that a pure strategy is a rule that specifies an action at each decision node of the player. So, this definition is consistent with our earlier definition of a pure strategy.

Analogous to static games, we can define a mixed strategy as follows.

Definition 5. *A* mixed strategy *for player* i *is a distribution over the set of pure strategies for player* i*.*

For example, one mixed strategy for player 2 in trivial matching pennies is $\sigma_2 = (0.5(H, H), 0.5(H, T))$, where player 2 plays (H, H) with probability 0.5, and plays (H, T) with probability 0.5.

In addition to this, in dynamic games, we have another notion of a strategy where a player chooses a random action.

Definition 6. *A* behavioral strategy *for player i is a rule that specifies a distribution over available actions at each of her information sets.*

In contrast to a pure strategy, which specifies a pure action at each information set, a behavioral strategy specifies a random action at each information set. For example, one behavioral strategy for player 2 in trivial matching is given by $\hat{\sigma}_2 = ((0.5H, 0.5T), (0.3H, 0.7T))$, where player 2 plays the mixed action $(0.5H, 0.5T)$ if player 1 plays H, and plays $(0.3H, 0.7T)$ if player 1 plays T.

Note that in a mixed strategy, a player chooses a pure strategy randomly at the *start* of the game, and follows that pure strategy throughout the rest of the game. In contrast, in a behavioral strategy, a player chooses an action randomly at *each* information set. In other words, in a mixed strategy, the player randomizes ("tosses the coin/rolls a die") once at the beginning of the game, whereas in a behavioral strategy, the player randomizes at every information set.

It turns out that in any game where a player has *perfect recall* about her past (she does not forget the past actions taken), for every mixed strategy there is an equivalent behavioral strategy, and for every behavioral strategy, there is an equivalent mixed strategy. Since we will only focus on such games, we will henceforth use the terms mixed strategy and behavioral strategy interchangeably.