Non-cooperative games in extensive form (game trees)

In non-cooperative game theory, a second basic model (besides the strategic form) is the game tree, also called the extensive-form game.

A game tree assumes that players choose their strategies sequentially, rather than simultaneously, with the goal of maximizing their payoff (or expected payoff, if the game includes chance). Game trees can describe players with perfect information, where players always know the game state and therefore the full history of the game, or imperfect information, where they do not.

Formally, a tree is a directed graph - that is, a finite set of nodes u and edges (u, v) with a distinguished root node r so that there is a unique path $u_0u_1...u_n$ from the root node $r = u_0$ to any other node $u = u_n$. A game tree then is a tree whose nodes represent game states, with terminal nodes (leafs) representing payoffs to every player.

Definition 4.2. A *game tree* is a tree with the following additional information:

- A finite set of N players.
- Each non-terminal node of the tree is either a *decision node* and belongs to one of the N players, or is a *chance node*.
- For a decision node u, every edge uv to a child v of u is assigned a different *move*. These moves are called the *moves at* u.
- For a chance node u, the edges uv to the children v of u are assigned *probabilities* (that is, non-negative reals that sum to 1).
- Every leaf of the game tree has an N-tuple of real-valued payoffs, one for each player. □

Note that a game tree also allows for chance, and a chance node can be replaced by a leaf of payoffs representing expected values.

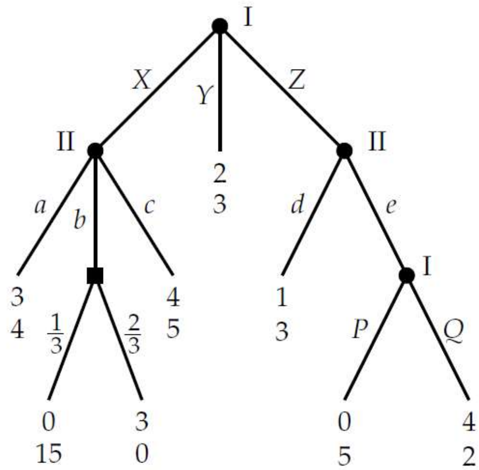


Figure 4.1 Example of a game tree. The square node indicates a chance move. At a leaf of the tree, the top payoff is to player I, the bottom payoff to player II.

Strategies and reduced strategies for game trees

In a game tree, a strategy is a derived concept, where as in a game in strategic form the strategies are assumed as given. A strategy in a game tree can be considered as a plan of action for the player, specifying a move for each game state that the player may encounter.

Definition 4.3. In a game tree, a *strategy* of a player specifies a move for every decision node of that player. □

Hence, if a player has *k* decision nodes, their strategies are in the form of a *k*-tuple of moves. For simplicity, we write this as an ordered list, for example, ACD instead of (A, C, D).

A strategy profile in an N-player game tree is thus an N-tuple of strategies, with one strategy for each player. (Note that this is formally a tuple of tuples, but strategies can be represented as character strings instead of tuples.)

Note that a strategy must specify a move even at the decision nodes that are unreachable because of a player's earlier moves. For simplicity, we can define a reduced strategy as a strategy in which these unreachable nodes are replaced by a placeholder.

Definition 4.5. In a game tree, a *reduced strategy* of a player specifies a move for every decision node of that player, except for those decision nodes that are unreachable due to an earlier own move, where the move is replaced by " $*$ ". A *reduced strategy profile* is a tuple of reduced strategies, one for each player of the game.

Importantly, reduced strategies only disregard decision nodes made unreachable because of a player's own earlier moves.

A reduced strategy profile in an N-player game tree is then an N-tuple of reduced strategies. (Sometimes, reduced strategies are defined differently - elimination of dominated strategies or elimination of strategies with identical payoffs, for example.)

Best responses and equilibrium for game trees with perfect information

Since players aim to maximize their expected payoffs, their best response moves at any game state can be solved using backwards induction. To perform backwards induction, mark the payoff-maximizing move(s) at each decision node for the relevant player, proceeding from the leaves upwards. (If there is always one optimal move for each decision node, then backwards induction gives a unique recommendation.)

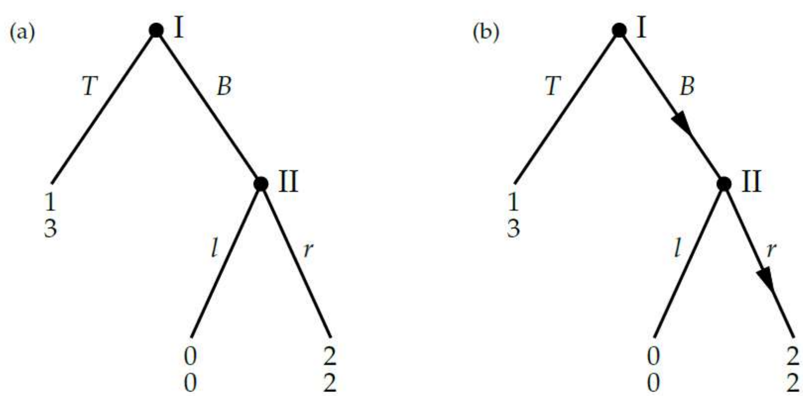


Figure 4.3 A two-player game tree (a) with backward induction moves (b).

Backward induction defines a best response move for every decision node of every player, which in turn gives a best response strategy for each player.

Therefore, the result of backward induction is at least one strategy profile of mutual best responses in every subgame of the game, and we call this equilibrium a subgame perfect equilibrium (SPE).

Theorem 4.6. *Backward induction defines an SPE.*

The proof of this proceeds by induction:

- Base case: For a decision node one move removed from leaves, this is trivially true. (An optimal move is an equilibrium.)

- Inductive step: Consider a decision node u for player X, where each possible subtree $T_1 \ldots T_n$ has a specified strategy profile defining an SPE. Then player X's optimal move(s) to T_i along with T_i's strategy profile defines an equilibrium, because if this was not the case, some player could improve her expected payoff by making some move in T_i's strategy profile. But this would imply that the moves selected so far in T_i do not define an equilibrium, contradicting the inductive assumption.

Note that an SPE must be an unreduced strategy profile, because moves at unreachable nodes can affect equilibrium. If the game has no chance moves, the equilibrium path is just a certain path in the game tree. In contrast, an SPE defines a move in every part of the game tree.

Two consequences of this theorem is that since backwards induction can be performed on any game tree with perfect information, all such games have a pure-strategy (non-randomized) equilibrium, which is the SPE.

Corollary 4.7. *Every game tree with perfect information has a pure-strategy equilibrium.*

Corollary 4.8. *Every game tree with perfect information has an SPE.*

Strategic form and reduced strategic form for game trees with perfect information

The strategic form of a game tree is defined by the strategy profile (set of strategies for each player), and the expected payoff to each player resulting from each strategy profile.

Definition 4.4. The *strategic form* of a game tree is defined by the set of strategies for each player according to Definition 4.3, and the expected payoff to each player resulting from each strategy profile. □

However, the game tree and the strategic form are played differently: game tree play is sequential, and strategic form play is simultaneous. Every game tree can be converted to strategic form, but a game in strategic form can only be represented by a tree with extra components representing imperfect information. When the players simultaneously select their strategies in the strategic form, they are assumed to follow these strategies as the play in the game tree unfolds. The strategic form therefore disregards the sequential representation of moves in the game tree.

By constructing the strategic form of a game tree, we can see that a game in strategic form may have equilibria that are not subgame perfect.

For the following game tree (a), we construct the corresponding game in strategic form (b). Note that as the tree only has three leafs, there is a repeated cell in the strategic form: (T, l) and (T, r) are identical because the "move" of player II is trivial.

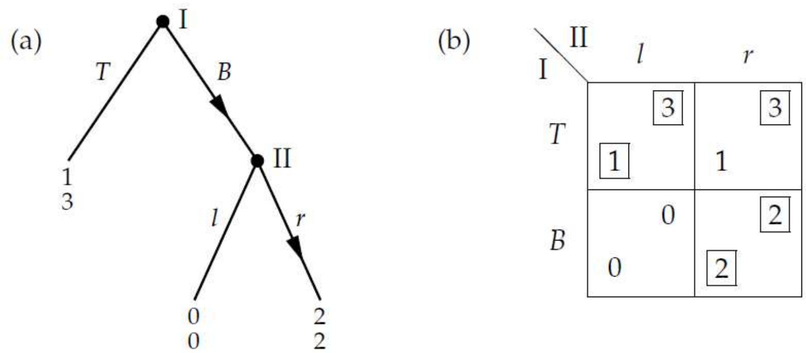


Figure 4.5 Game tree (a) of the Threat game with backward induction moves, and its strategic form (b) with best responses. Note: In the game tree, the payoffs to player I are shown at the *top*, but in the strategic form they appear at the *bottom* left of each cell.

While backwards induction in (a) specifies the unique SPE (B, r), the game has a second equilibrium (T, l) as seen in (b).

- If Player II threatens to choose the suboptimal l, then Player I will choose T, giving Player II a higher payoff than the SPE. If Player II knows Player I will choose T, then it does not matter whether she chooses l or r.
- Note that l is weakly dominated by r, but it is nevertheless part of an equilibrium.

We can also construct the reduced strategic form of a game tree by listing all reduced strategies for each player and the expected payoffs for each reduced strategy profile. Specifically:

- For each player, choose a fixed order of the decision nodes, where nodes that come earlier in the tree should come first. If the player has k decision nodes, a (reduced) strategy has k "slots" to define a move, one for each decision node.

- Generate all (reduced) strategies. A reduced strategy defines a move for each slot, or "$*$" if the respective decision node is unreachable due to an earlier move. For two players, list the (reduced) strategies of player I as rows and of player II as columns.
- For each *leaf* of the game tree, consider the path from the root to the leaf. Assume first that there are no chance moves on that path. For each player, consider the moves of that player on that path. For example, the leftmost leaf in Figure 4.1 with payoffs $(3, 4)$ has move X for player I and move a for player II.
- For each player, identify *all strategies* that agree with the moves on the path to the leaf. In the example, this is the reduced strategy $X*$ for player I, or the unreduced strategies XP and XQ, and the strategies ad and ae for player II. For all these strategy pairs, enter the payoff pair, here $(3, 4)$, in the respective cells. Repeat this for each leaf.

  As another example, the leaf with payoffs $(2, 3)$ that directly follows move Y is reached by move Y of player I (which defines row $Y*$ in the reduced strategic form), and *no moves* of player II, so *all strategies* of player II agree with these moves; that is, the entire row $Y*$ is filled with the payoff pair $(2, 3)$.
- If there are chance moves on the path from the root to the leaf, let p be the product of chance probabilities on that path. Multiply each payoff at the leaf with p, and *add* the result to the current payoffs in the respective cells that are determined as before (assuming the payoffs in each cell are initially zero).

As illustration, for the following game tree, we construct the reduced strategic form by following these steps.

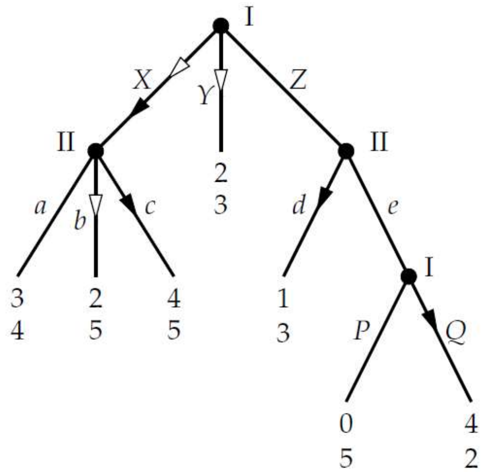
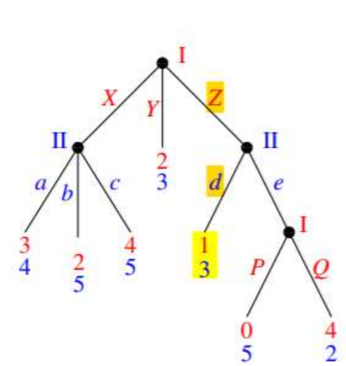


Figure 4.4 The game tree in Figure 4.1 with the chance move replaced by its expected payoffs, and backward induction moves that depend on player II's choice between b (white arrow, two possible choices X or Y by player I) and c (black arrow, unique choice X by player I).

First, generate a list of all reduced strategies for each player. For Player I, these are X*, Y*, ZP, and ZQ. Player II has no reduced strategies because she has no decision nodes that are made unreachable by her own earlier moves.

Next, place these reduced strategies into a table and fill in the entries by examining each leaf and, in turn, filling in its payoff for all strategies that end up here. For example, here we examine the node with payoff (1, 3), and fill in this for all strategies wherein Player I chooses Z while Player II chooses d.



I \ II	ad	ae	bd	be	cd	ce
X*	3, 4	3, 4	2, 5	2, 5	4, 5	4, 5
Y*	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3
ZP	1, 3		1, 3		1, 3	
ZQ	1, 3		1, 3		1, 3	

The final reduced strategic form demonstrates that the (non-SPE) equilibria are (X*, bd), (X*, cd), (X*, ce), and (Y*, bd).

I \ II	ad	ae	bd	be	cd	ce
$X*$	$\boxed{3}$, 4	3, 4	$\boxed{2}$, $\boxed{5}$	2, $\boxed{5}$	$\boxed{4}$, $\boxed{5}$	$\boxed{4}$, $\boxed{5}$
$Y*$	2, $\boxed{3}$	2, $\boxed{3}$	$\boxed{2}$, $\boxed{3}$	2, $\boxed{3}$	2, $\boxed{3}$	2, $\boxed{3}$
ZP	1, 3	0, $\boxed{5}$	1, 3	0, $\boxed{5}$	1, 3	0, $\boxed{5}$
ZQ	1, $\boxed{3}$	$\boxed{4}$, 2	1, $\boxed{3}$	$\boxed{4}$, 2	1, $\boxed{3}$	$\boxed{4}$, 2

Figure 4.7 Reduced strategic form of the extensive game in Figure 4.1. The star $*$ stands for an arbitrary move at the second decision node of player I, which is not reachable after move X or Y.

Note that the reduced strategy form can still be very large, as the number of move combinations grows exponentially with the number of decision nodes.

Commitment games

A commitment game is a new sequential game obtained from a simultaneous two-player game in strategic form. In the commitment game, Player I moves before Player II; thus, Player II chooses a strategy depending on the move of Player I.

In the commitment version of an *m x n* game, Player I retains *m* strategies, while Player II now has n^m strategies: for each possible move by Player I, Player II can react with one of *n* moves.

This means that the number of strategies for Player II is much larger than in strategic form, but the game is also simpler because Player II plays deterministically, and Player I can simply solve an optimization problem.

In commitment games, we are only concerned with finding SPE, rather than equilibria that are not subgame perfect.

For example, in the simultaneous Quality game, equilibrium is at (B, r). But in the sequential Quality game, where Player I (restaurateur) chooses first:

- SPE is at (T, $l_T r_B$)
    - second layer: if player I chooses T, player II will choose l, giving payoff 2 to both players
    - if player I chooses B, player II will choose r, giving payoff 1 to both players
    - because player I anticipates this, he will choose T
- the original equilibrium (B, r) is a non-subgame perfect equilibrium, as these are mutual best responses

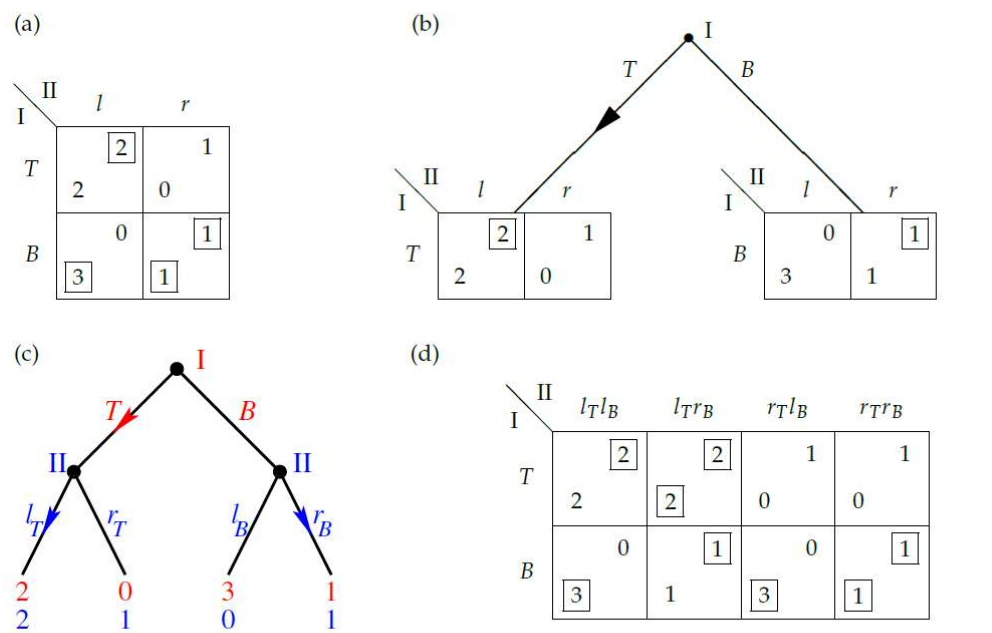


Figure 4.9 The simultaneous Quality game (a) from Figure 3.2(b), converted to its sequential *commitment* game (b) where player I moves first and player II can *react* to player I's move. The conventional representation of (b) is the extensive game (c). The arrows in (b) and (c) show the backward induction outcome. The strategic form of (c) with best-response payoffs is shown in (d).

Note that while the first-mover advantage ensures that the player who moves first is not worse off, the choice of which player to move first is not trivial. In fact, if Player II moves first, then equilibrium will be at (r, $B_l B_r$) with payoff 1 to both.

On the equilibrium path, the move T of player I is followed by move l_T of player II. However, it is not sufficient to simply call this equilibrium (T, l_T) or (T, l), because player II's move r_B must be specified to know that T is player I's best response.

Another example of a commitment game is the Stackelberg duopoly in extensive form. To form a game tree with the setup from the previous chapter, we split the moves of player II by the prior move of player I, and use backwards induction to solve for player II's optimal move at each of the four decision nodes, which are 6, 4, 4, 3 to player I's moves 0, 3, 4, 6 respectively. Knowing player II's best responses to his possible moves, player I optimizes his profits by producing at 6 for payoff 18. Thus, the SPE is (6, 6-(x/2)).

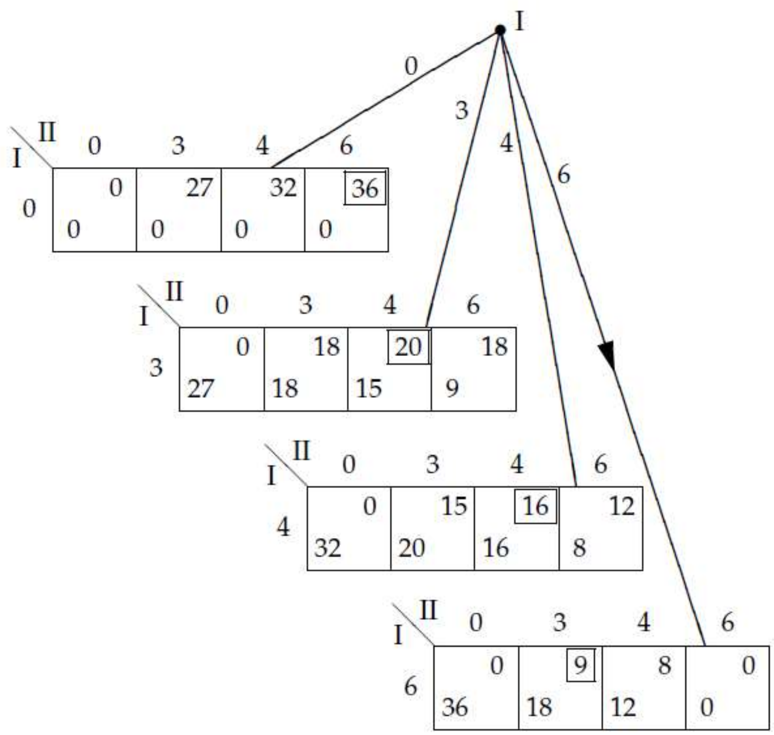


Figure 4.10 Commitment game for the duopoly game in Figure 3.8.

Note that in the commitment game, Player I's strategy is a quantity x, while Player II's strategy is a function y(x) that defines a best response for each quantity of x. The SPE must specify not only the optimal commitment x_0 and optimal response $y(x_0)$, but rather a complete specification of player II's strategy y(x).