A Nash equilibrium is a pair of strategies that are best replies to each other. If the players share a common conjecture that each will play its Nash equilibrium strategy, then that Nash equilibrium is self-enforcing. Neither player has an incentive to change its strategy unilaterally.

Some games do not have Nash equilibria in pure strategies. But they do have Nash equilibria in mixed strategies. Mixed strategies represent the players' uncertainty about what pure strategy the other player will play. In a mixed strategy Nash equilibrium, each player is indifferent among the pure strategies that can be played in its mixed strategy.

Further Reading

My presentation in this chapter broadly matches the standard texts on noncooperative game theory. Brandenburger 1992 is an excellent first source on common conjectures and Nash equilibrium.

The literature on the spatial theory of elections is massive, and much of it begins with Downs 1957. Formal spatial theory of elections begins in the mid-1960s. Ordeshook 1986 and Enelow and Hinich 1984 provide good surveys of the topic. Enelow and Hinich 1990 is a collection of recent contributions to spatial theory.

The model of political reform is taken directly from Geddes 1991. I refer the reader to this article for the full development of the model and a discussion of the evidence from Latin American democracies.

Once again, the standard references are good on n-person games. Ordeshook 1986 should be your first source for n-person game theory and its applications to political science. Owen 1982 is the best technical textbook on n-person game theory. Harsanyi 1977 provides a very good treatment of the Nash bargaining solution, including extensions of the Nash bargaining solution and explication of the links between it and the Zeuthen bargaining model (a behavioral model where the player with the greatest risk makes the next small concession).

Chapter Five
Solving Extensive-Form Games: Backwards Induction and Subgame Perfection

Chapters Five through Nine analyze games in their extensive form rather than their strategic form. The extensive form of a game is more fundamental than its strategic form because it specifies all of the moves and their order. Nash equilibria analyze the interaction of strategies in the strategic form. Henceforth in this book, we analyze individual moves in the extensive form. Working with extensive-form games reinforces the connection between utility theory and game theory. Each individual move in a game is treated as a separate decision under uncertainty. A player projects what responses are likely to be made to each of its possible moves and chooses the move that produces the best final outcome. A player assumes the other players are rational and uses its knowledge of the game to project other players' future moves. Often, a player cannot be certain what the other players have done or will do. Players in such cases calculate expected utilities by using subjective probability distributions that express their beliefs about what has occurred or what will occur in the game.

This chapter and the next two present the central equilibrium concepts for extensive-form games—subgame perfect, perfect Bayesian, sequential, and perfect equilibrium. All four eliminate Nash equilibria that allow the players to make unreasonable moves. These equilibrium concepts reduce the number of Nash equilibria by imposing additional rationality conditions. Perfect Bayesian equilibrium is the most widely used definition of equilibrium. It is more general than subgame perfect equilibrium and easier to solve for than sequential or perfect equilibria. Figure 5.1 gives a Venn diagram that relates these four types of equilibria and Nash equilibrium. Any equilibrium that is perfect must also be sequential, perfect Bayesian, subgame perfect, and Nash. Any subgame perfect equilibrium is also a Nash equilibrium. All types of equilibria are strategy pairs (or sets for games with more than two players).

Extensive forms allow a more detailed analysis of the strategic interaction between the players than strategic forms do. Nash equilibria compare complete strategies of the players to see if they are optimal against each other. But the sequence of a game should affect how the players play the game. How do prior moves affect a player's decisions? What expectations about the other players' future moves does a player form during the game? How do its anticipations change as the game progresses?

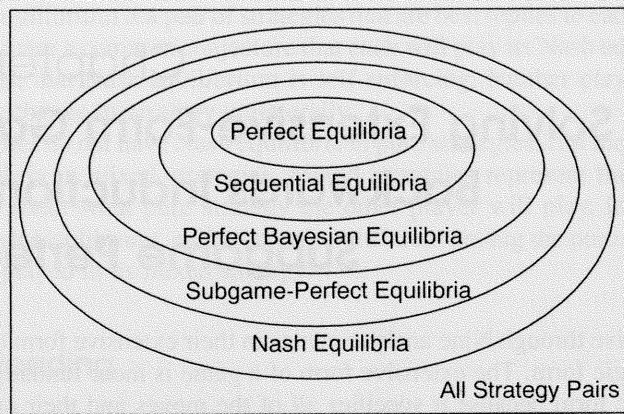

Figure 5.1 The Relationship of Five Different Types of Equilibria

The moves the players make at nodes reached in an equilibrium are called their behavior **on (or along) the equilibrium path**. Decisions about moves at nodes that are not reached in an equilibrium are behavior **off the equilibrium path**. Nash equilibria examine the rationality of moves on the equilibrium path. Sometimes, a player's choice on the equilibrium path depends on another player's choice off the equilibrium path. The latter player could make an irrational move off the equilibrium path and thus alter the first player's choice on the equilibrium path in a Nash equilibrium.

In Nash equilibria, players can precommit themselves to carrying out incredible threats and promises. The strategic form assumes that players choose complete strategies before the game begins. The players can precommit themselves at the start of the game to make moves in the future. A player can commit itself to threats and promises that are against its own interest to carry out. A precommitment to an incredible threat could induce the other player to move so that the threat does not need to be carried out. But games are sequential strings of decisions: players cannot make moves until they reach them. Precommitment—making a binding commitment to a move before that move has been reached—violates the idea of a game as a sequence of decisions. Subgame perfection, unlike Nash equilibrium, tests the credibility of moves on and off the equilibrium path.

I begin with subgame perfection in this chapter, introduce perfect Bayesian equilibrium in Chapter Six, and discuss sequential and perfect equilibrium in Chapter Seven. To introduce subgame perfection, I present backwards induction first. Backwards induction is a simple way to solve for the subgame perfect equilibrium of a game of perfect and complete information. In such games, a player always knows all prior moves and the other players' payoffs when it must move. It can anticipate future moves of the other players if it assumes they will play rationally. Backwards induction analyzes the game "backwards"—

begin with the players' choices that lead directly to terminal nodes, choose their best moves at those nodes, and then work "backward" to determine which preceding moves are optimal by using the projections of moves later in the tree.

Backwards induction requires that the players always make optimal moves from each node in the tree. In a game of complete and perfect information, optimal moves can be judged because players can anticipate future moves. Subgame perfection generalizes this idea to some moves in games of imperfect information. A subgame is a portion of a game, starting at a single node and including all subsequent nodes, that forms a game by itself. Subgame perfection requires players to play Nash equilibria in all subgames of a game: players must make optimal moves at all nodes that begin a subgame. Unfortunately, not all nodes begin subgames. To test the rationality of moves at such information sets, we need the concept of beliefs, which are central to perfect Bayesian equilibrium. That is the focus of Chapter Six.

The solution concepts developed for extensive-form games provide a way to solve games with limited information, where the players do not know some aspect of the game. A player forms conjectures about the uncertainties of the game. It revises these conjectures as it learns about the game from the moves of the other players. These games are particularly interesting models of political and social behavior because actors rarely know the exact situation they face. Bluffing and other strategic misrepresentations are common in these games. Limited-information games allow us to analyze communication formally, as well as misrepresentation and deception. Chapter Eight introduces these games.

A quick review of the elements of extensive-form games from Chapter Three should help you recall the detailed structure of those games. Game trees specify the players' moves in sequence. Choices are given as branches out of nodes, the decision points in the tree. Each node is assigned to one and only one player, including Chance as a player. Each player's nodes may be grouped into information sets that express the player's knowledge of prior moves. Terminal nodes give endpoints of the game, and a payoff for each player is associated with each terminal node. We assume the game is common knowledge. Perfect information means that a player knows all prior moves whenever it must move (i.e., all information sets are singletons); complete information that all players' payoffs are common knowledge.

This chapter illustrates backwards induction with several models from legislative studies. Models in legislative studies were using backwards induction before noncooperative game theory entered political science. Legislative agendas often require multiple votes to resolve an issue. I begin with sophisticated voting, where voters consider how the outcome of earlier votes affects later votes. Because the order of the agenda can influence the outcome, control of the agenda can be powerful. Both sophisticated voting and agenda control are essential for understanding structure-induced equilibria, a model of legislative rules. I discuss a very simple model of noncooperative bargaining, the Rubinstein

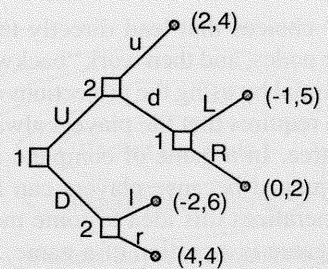

Figure 5.2 An Example for Backwards Induction

bargaining model, and connect it to the Nash bargaining model discussed in Chapter Four. The last example extends the logic of the Rubinstein bargaining model to bargaining in Congress. This model provides a second argument about why congressional rules exist and how they shape outcomes. This chapter concludes with a discussion of the limitations of backwards induction.

Backwards Induction

Solving a sequential game with perfect information is easy. We begin with decisions that lead only to terminal nodes of the game, choose the action that maximizes the utility of the player choosing at that node, and then work backwards through the earlier nodes of the game. Each player can predict what the other players will do at subsequent nodes, so it can predict the exact consequences of its possible moves from each node. For earlier nodes, we substitute the eventual outcome that will be reached for each move, using the anticipated future moves of the other players. We can determine all players' optimal choices by working backwards in this fashion. Because every information set is a singleton under perfect information, the players know all prior moves and can anticipate all future moves when they must choose their moves.

Example: Solve the game in Figure 5.2 by backwards induction. Begin with Player 1's final move. He will choose Right, labeled R in Figure 5.2, over Left, labeled L; his payoff for choosing Right is 0 and for Left it is −1. Player 2 will choose left, labeled l, in her lower move; she gains a payoff of 6 instead of 4 for choosing right, labeled r. At Player 2's upper move, she gains a payoff of 4 if she chooses up (u). If she chooses down (d), then Player 1 will choose Right, and Player 2's payoff will be 2. She chooses up over down, then. Player 1's initial move depends on Player 2's future moves. If he plays up (U), she will play up (u), giving him a payoff of 2. If he plays Down (D), she will play left (l), giving Player 1 a payoff of −2. Player 1 prefers Up. The result is (U,R;u,l) producing an outcome of (2,4).

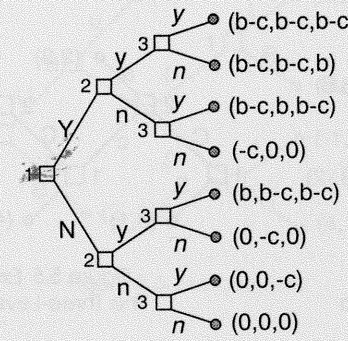

Figure 5.3 The Pay Raise Voting Game

Backwards induction is so simple that one may think that it cannot provide any interesting insights. But consider the following problem.

Example: Three legislators are voting on whether to give themselves a pay raise. All three want the pay raise; however, each faces the same small cost in voter resentment, c, if he or she votes for the pay raise. The benefit for the raise, b with b > c, exceeds the cost of voting for it. If they vote sequentially on the raise, is it better to vote first or last?

At first blush, you might think that voting last is preferable because then you can decide whether the raise passes if the first two members split their votes. The extensive form in Figure 5.3 represents this game. Label the players 1, 2, and 3, according to the order in which they vote on the pay raise. Each player can vote yes (Y, y, or *y*) or no (N, n, or *n*) on the pay raise. They vote sequentially; Player 3 knows the votes of Players 1 and 2 when she must vote. The pay raise passes if two of the three vote yes. Any player voting yes pays cost c regardless of whether the raise passes. Each of the eight terminal nodes gives a different combination of votes, with the players' payoffs.

The backwards induction begins with Player 3's vote. If both or neither of Players 1 and 2 vote for the raise, she votes against it. In the former case (her top node), her payoff for voting yes is b − c; for voting no, it is b. In the latter case (her bottom node), her payoff is −c for voting yes and 0 for voting no. If Players 1 and 2 split their votes on the pay raise (her two middle nodes), Player 3 votes yes. Her payoff for voting yes is b − c, versus a payoff of 0 for voting no.

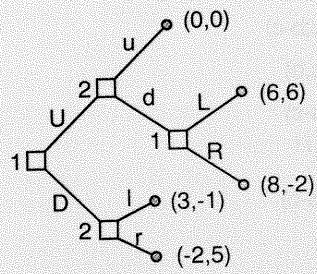

Figure 5.4 Exercise 5.1a

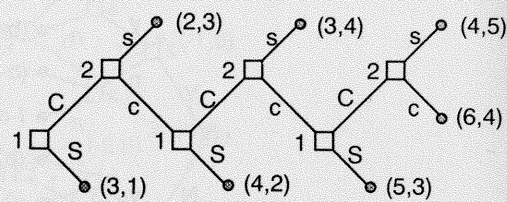

Figure 5.5 Exercise 5.1b:
The Three-Level Centipede

Player 2 considers how Player 3 will vote when she casts her vote. At her upper node (when Player 1 has voted yes), Player 3 will vote yes if Player 2 votes no, producing a payoff of b for Player 2. If Player 2 votes yes from her upper node, Player 3 will vote no, giving Player 2 a payoff of b – c. Voting no is better than voting yes for Player 2 if Player 1 has already voted yes. If Player 1 votes no (Player 2's lower node), Player 3 will vote yes if Player 2 votes yes, leading to a payoff of b – c for Player 2. If Player 2 votes no, Player 3 will also vote no, producing a payoff of 0 for Player 2. Player 2 prefers voting yes when Player 1 votes no.

Finally, Player 1 can anticipate the votes of Players 2 and 3 when he votes. If he votes yes, Player 2 will vote no and Player 3 will vote yes, leading to a payoff of b – c for him. If he votes no, Players 2 and 3 will vote yes, giving him a payoff of b. Voting no is better for Player 1.

Write down a set of strategies for the game in Figure 5.3 as follows: (Player 1's vote; Player 2's vote if Player 1 votes yes, Player 2's vote if Player 1 votes no; Player 3's vote if Players 1 and 2 vote yes, Player 3's vote if Player 1 votes yes and Player 2 votes no, Player 3's vote if Player 1 votes no and Player 2 votes yes, Player 3's vote if Players 1 and 2 vote no). The equilibrium of the game in Figure 5.3 is (N;n,y;*n,y,y,n*). In equilibrium, Player 1 votes against the raise, forcing Players 2 and 3 to vote in favor of it. Voting first is better than voting last; you can force the others to take the heat and still get the raise.

> Exercise 5.1: Solve the games in Figures 5.4 through 5.6 using backwards induction.
>
> **a)** Solve the game in Figure 5.4.
>
> **b)** Solve the game in Figure 5.5. This game is called the three-level centipede. Each move of s stops the game, and each move of c continues it. The three levels are the three repetitions of the basic choice of stopping or continuing the game.
>
> **c)** Solve the game in Figure 5.6.

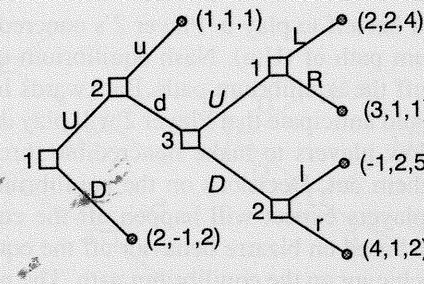

Figure 5.6 Exercise 5.1c

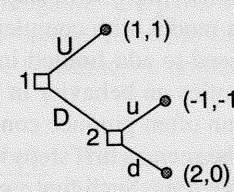

Figure 5.7 A Noncredible-Threat Game

		Player 2 u	Player 2 d
Player 1	U	(1,1)	(1,1)
	D	(-1,-1)	(2,0)

Figure 5.8 The Strategic Form of the Noncredible-Threat Game

How does the result of a backwards induction compare to the Nash equilibria of a game? Backwards induction always produces a Nash equilibrium, but some Nash equilibria are not found by a backwards induction. Nash equilibrium assesses the rationality only of moves on the equilibrium path. Backwards induction assesses the rationality of all moves in a strategy, both on and off the equilibrium path. Thus Nash equilibria where a player plays irrationally off the equilibrium path will not be found by a backwards induction.

> Example: Perform a backwards induction on the game in Figure 5.7. Player 2 chooses down, d, if she has to move because a payoff of 0 is better than –1. Anticipating her move, Player 1 prefers Down, D, to Up, U. Backwards induction leads to a strategy pair of (D;d).

But the game in Figure 5.7 has two Nash equilibria: (U;u) and (D;d). Figure 5.8 gives the strategic form of this game. Consider how each of these Nash equilibria translates into behavior in the extensive form in Figure 5.7. (D;d) corresponds to the backwards induction above. In (U;u), Player 2 uses the threat of u to coerce Player 1 into playing U. But this threat is not credible because Player 2 will prefer to play d if Player 1 chooses D. However, if Player 1 plays U, Player 2 never has to choose. The Nash equilibrium (U;u) allows

Player 2 to precommit herself to play u. Player 2's noncredible threat to play u is off the equilibrium path of (U;u). Nash equilibrium does not judge the rationality of moves off the equilibrium path. Backwards induction does and finds that Player 1 should anticipate that Player 2 will play d if she must move.

Nash equilibria allow players to make noncredible threats provided they never have to carry them out. Decisions on the equilibrium path are driven in part by what the players expect will happen off the equilibrium path. If those expectations are based on bizarre behavior off the equilibrium path, we can produce bizarre behavior on the equilibrium path. The players should play optimally even in portions of the game that they do not expect to reach in equilibrium. Backwards induction requires rational play at all nodes when we can specify what the other players will do in future moves.

Backwards induction can be a useful tool in solving more complex games. Often the analysis of a problem begins with a model with complete and perfect information. Later, the model can be revised to add limited information. We can ascertain the effects of those uncertainties on behavior in the model. Models under complete and perfect information often produce conclusions at variance with reality. But such models should be seen as first steps in the modeling process rather than as definitive representations. Such first steps are also helpful in providing some intuition about what the later models under limited information may look like.

Subgame Perfection

Backwards induction requires that the players make optimal choices from each node in the game. However, it can only judge an optimal move when the precise outcome of each available move can be determined. Information sets with multiple nodes confound a backwards induction. A player moving at an information set with multiple nodes cannot determine which node it is at in the information set. If it prefers different moves for different nodes of the same information set, it cannot determine which move is best from that information set.

Subgame perfection generalizes the idea of checking the optimality of all moves to some moves in games with information sets that contain multiple nodes. Subgame perfection includes backwards induction but is more general. Players should choose optimally from every point in the game whether those nodes are reached in equilibrium or not. To formalize this idea, we need to know what "from every point in the game" means.

> **Definition**: A **proper subgame** is a subset of the nodes of a game starting with an initial node and including all its successors that preserves all information sets of the game and over which a (new) game is defined by the restriction of the original game elements (acts, payoffs, chance moves, information sets, etc.) to these nodes.

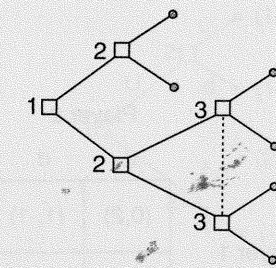

Figure 5.9 A Game Tree That Illustrates Subgames

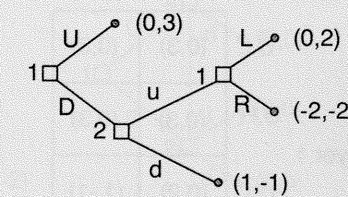

Figure 5.10 A Game of Complete and Perfect Information

Proper subgames are parts of a game that can be treated as games in their own right—hence the name *subgame*. In the game in Figure 5.7, Player 2's move alone constitutes a proper subgame. The entire game is also a subgame of itself.

> Example: The game tree in Figure 5.9 has three proper subgames, the two sections of the game starting at either of Player 2's nodes and the entire game itself. I have deliberately left out the strategies and payoffs to focus on the game tree. No proper subgame can be started at either of Player 3's nodes. Player 3's information set would be split. Subgames, like all games, must begin with a single node.

To ensure that each player is rational at all stages of the game, we require strategies to be optimal in all proper subgames. To see if a Nash equilibrium is subgame perfect, decompose the game into all its subgames, and then check whether the restriction of the equilibrium strategies to each subgame constitutes a Nash equilibrium for that subgame.

> **Definition**: A set of strategies is **subgame perfect** if for every proper subgame, the restriction of those strategies to the subgame forms a Nash equilibrium.

> Example: The game in Figure 5.10 has four Nash equilibria in pure strategies: (U,L;u), (U,R;u), (D,L;u), and (D,R;d). You can find them in Figure 5.11, which gives the strategic form of this game. How many of these Nash equilibria are subgame perfect?

The game in Figure 5.10 has three subgames: the entire game, the second and third moves, and the third move alone. The subgame consisting of just the third move has only one Nash equilibrium, L. The subgame of the second and third moves has two Nash equilibria, (L;u) and (R;d). Figure 5.12 gives the

		Player 2	
		u	d
Player 1	U,L	(0,3)	(0,3)
	U,R	(0,3)	(0,3)
	D,L	(0,2)	(1,-1)
	D,R	(-2,-2)	(1,-1)

Figure 5.11 The Strategic Form of Figure 5.10

		Player 2	
		u	d
Player 1	L	(0,2)	(1,-1)
	R	(-2,-2)	(1,-1)

Figure 5.12 The Strategic Form of a Subgame of Figure 5.10

strategic form of this subgame. Thus any subgame-perfect equilibrium must have (L;u) for its last two moves. For example, (U,R;u) is not subgame perfect because the restriction of it to either of the proper subgames is not a Nash equilibrium of that subgame. R is not a Nash equilibrium of the subgame of just the third move, and (R;u) is not a Nash equilibrium of the subgame of the last two moves. Thus there are two subgame-perfect equilibria in pure strategies, (D,L;u) and (U,L;u).

<u>Exercise 5.2</u>: Find the Nash equilibria of the games in Figures 5.13 through 5.17. You do not need to search for mixed equilibria, except in Figure 5.17. Determine which of these Nash equilibria are subgame perfect and which are not.

a) Find the Nash equilibria for the game in Figure 5.13. Which ones are subgame perfect?

b) Find the Nash equilibria for the game in Figure 5.14. Which ones are subgame perfect?

c) Find the Nash equilibria for the game in Figure 5.15. Which ones are subgame perfect? Does your analysis change if Player 2 cannot tell what Player 1's move has been when she must decide? (Hint: What game is this?)

d) Find the Nash equilibria for the game in Figure 5.16. Which ones are subgame perfect?

e) Find the Nash equilibria for the game in Figure 5.17. Which ones are subgame perfect? (Hint: Start by finding the Nash equilibria for the subgame that Players 3 and 4 play in the last two moves. Extra Hint: What well-known game are Players 3 and 4 playing? For each of those equilibria, calculate the values of Players 1 and 2

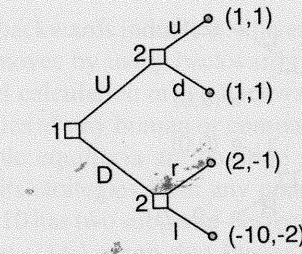

Figure 5.13 Exercise 5.2a

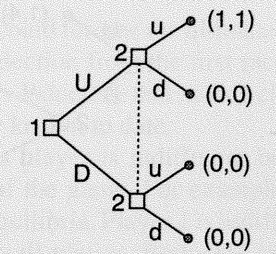

Figure 5.14 Exercise 5.2b

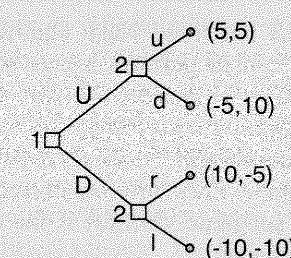

Figure 5.15 Exercise 5.2c

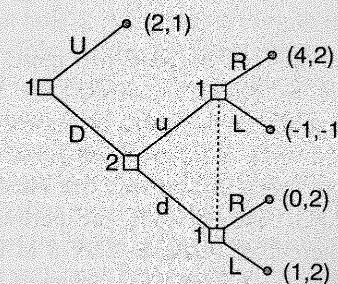

Figure 5.16 Exercise 5.2d

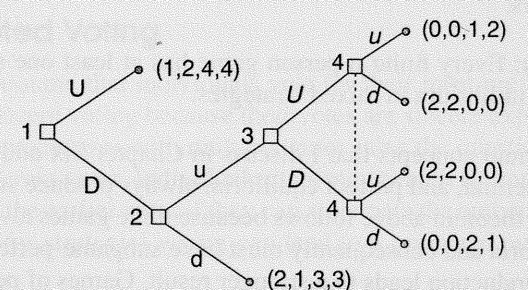

Figure 5.17 Exercise 5.2e

for that subgame before determining their Nash equilibria over the first two moves.)

Backwards induction is a special case of subgame perfection. Games with perfect information decompose into proper subgames from every node. Backwards induction, like subgame perfection, requires equilibrium play in each of those subgames. But subgame perfection is stronger than backwards induction. It can eliminate Nash equilibria where backwards induction is powerless by ruling out equilibria where a player makes an noncredible threat in a subgame containing an information set with multiple nodes.

		Player 2 u	Player 2 d
Player 1	U,L	(0,3)	(0,3)
	U,R	(0,3)	(0,3)
	D,L	(0,2)	(1,-1)
	D,R	(-2,-2)	(1,-1)

Figure 5.11 The Strategic Form of Figure 5.10

		Player 2 u	Player 2 d
Player 1	L	(0,2)	(1,-1)
	R	(-2,-2)	(1,-1)

Figure 5.12 The Strategic Form of a Subgame of Figure 5.10

strategic form of this subgame. Thus any subgame-perfect equilibrium must have (L;u) for its last two moves. For example, (U,R;u) is not subgame perfect because the restriction of it to either of the proper subgames is not a Nash equilibrium of that subgame. R is not a Nash equilibrium of the subgame of just the third move, and (R;u) is not a Nash equilibrium of the subgame of the last two moves. Thus there are two subgame-perfect equilibria in pure strategies, (D,L;u) and (U,L;u).

<u>Exercise 5.2</u>: Find the Nash equilibria of the games in Figures 5.13 through 5.17. You do not need to search for mixed equilibria, except in Figure 5.17. Determine which of these Nash equilibria are subgame perfect and which are not.

a) Find the Nash equilibria for the game in Figure 5.13. Which ones are subgame perfect?

b) Find the Nash equilibria for the game in Figure 5.14. Which ones are subgame perfect?

c) Find the Nash equilibria for the game in Figure 5.15. Which ones are subgame perfect? Does your analysis change if Player 2 cannot tell what Player 1's move has been when she must decide? (Hint: What game is this?)

d) Find the Nash equilibria for the game in Figure 5.16. Which ones are subgame perfect?

e) Find the Nash equilibria for the game in Figure 5.17. Which ones are subgame perfect? (Hint: Start by finding the Nash equilibria for the subgame that Players 3 and 4 play in the last two moves. Extra Hint: What well-known game are Players 3 and 4 playing? For each of those equilibria, calculate the values of Players 1 and 2

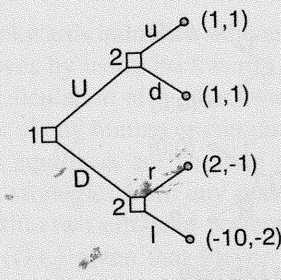

Figure 5.13 Exercise 5.2a

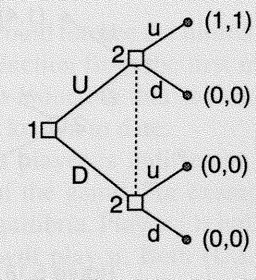

Figure 5.14 Exercise 5.2b

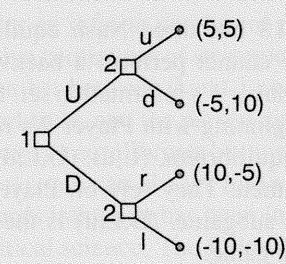

Figure 5.15 Exercise 5.2c

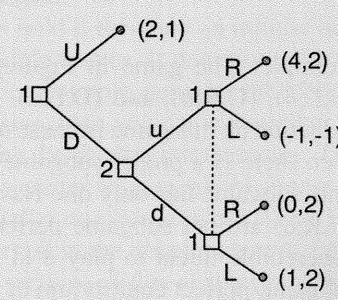

Figure 5.16 Exercise 5.2d

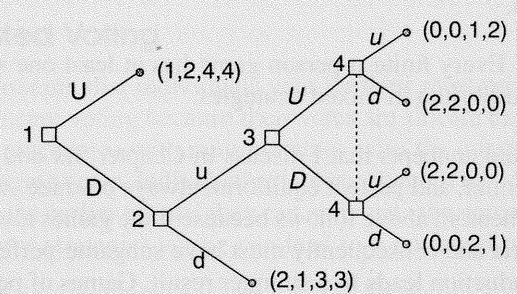

Figure 5.17 Exercise 5.2e

for that subgame before determining their Nash equilibria over the first two moves.)

Backwards induction is a special case of subgame perfection. Games with perfect information decompose into proper subgames from every node. Backwards induction, like subgame perfection, requires equilibrium play in each of those subgames. But subgame perfection is stronger than backwards induction. It can eliminate Nash equilibria where backwards induction is powerless by ruling out equilibria where a player makes an noncredible threat in a subgame containing an information set with multiple nodes.

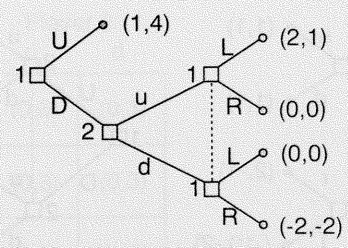

Figure 5.18 An Example of Subgame Perfection

Example: The game in Figure 5.18 has three Nash equilibria: (U,L;d), (U,R;d), and (D,L;u). We cannot perform a backwards induction on this game because of Player 1's information set. However, there is a proper subgame beginning with Player 2's move. This subgame has only one Nash equilibrium, (L;u). (U,L;d) and (U,R;d) are not subgame perfect, then. They rely on Player 2's noncredible threat to play d in the subgame. (D,L;u) is the only subgame-perfect equilibrium of this game.

As the following theorem states, subgame-perfect equilibria always exist in mixed strategies.

Theorem: Every finite n-person game has at least one subgame-perfect equilibrium in mixed strategies.

The equilibrium concepts that I discuss in Chapter Six and Seven, perfect Bayesian, sequential, and perfect equilibria, always produce subgame-perfect equilibria. The theorem above follows because finite games always have those types of equilibria and consequently must have subgame-perfect equilibria.

Backwards induction leads to a stronger result. Games of perfect and complete information not only have equilibria; they typically have an unique equilibrium in pure strategies.

Theorem (Kuhn-Zermelo): Every finite n-person game of perfect and complete information typically has a unique subgame-perfect equilibrium in pure strategies. There may be multiple pure strategy equilibria if a player is indifferent between two or more of its pure strategies.

The first version of this theorem, formulated by Zermelo (1913), is the first result in game theory. Zermelo showed that chess has a winning strategy: White can force a victory, Black can force a victory, or either can force a draw.

He used backwards induction. Players could foresee the ultimate consequences of their moves by using backwards induction from the first move. Of course, the actual calculation of the backwards induction that solves chess is beyond the abilities of any human or computer known to date.

Multiple equilibria are possible if a player is indifferent between two or more of its strategies from any node of the game. For example, the game in Figure 5.10 has two subgame-perfect equilibria. Player 1 is indifferent between (U,L) and (D,L) given that Player 2 will play u. Both (U,L;u) and (D,L;u) are subgame-perfect equilibria of this game, as are all mixed strategies where Player 1 mixes between these two strategies.

The Kuhn-Zermelo theorem does not hold if there are an infinite number of strategies or an infinite number of possible moves. It does not hold, then, for games with a continuum of strategies. For example, consider the game where I name a number between 0 and 1 exclusive, then you name a number between 0 and 1 exclusive, and the person naming the smaller number wins. This game does not have an equilibrium because no matter what number I choose between zero and one, you can always find a smaller number between 0 and 1.

Backwards induction has been important in the development of formal theory in political science. The following sections of this chapter present several models that use backwards induction and subgame perfection.

Sophisticated Voting

We typically assume that individuals vote for their most preferred outcomes. We call this sincere voting because their votes are sincere expressions of their interests. But procedures with multiple votes raise the question of the actors' expectations across votes. Individual actors may benefit by voting against their preferences in the earlier votes if they can anticipate the outcome of later votes. Sophisticated (or strategic) voting can lead to a preferred outcome for those actors.

Example: Assume there are three alternatives—x, y, and z—and three voters—Players 1, 2, and 3—with the following preferences:

Player 1: xPyPz;

Player 2: yPzPx; and

Player 3: zPxPy.

Sincere voting requires each actor to vote according to its preferences in any pairwise comparison of alternatives. An actor casts a sophisticated vote when it votes against its preferences in a pair-