

1 Introduction

One of the greatest challenges for computer science is to build computer systems that can work together. The integration of automated systems has always been a challenge, but as computers have become more sophisticated, the demands for coordination and cooperation have become increasingly important and pervasive. It is not only basic level components, such as printers, disks, and CPUs, but also high-level complex systems that need to coordinate and cooperate.

Cooperation and coordination are required for those complex problems that cannot be solved by a single system in isolation, but require several systems to work together interactively. For example, providing a user with an integrated picture of his investment portfolio over time, using the information resources already available over the Internet, requires the collaboration of a set of intelligent systems (Sycara and Zeng 1996). Furthermore, there are heterogeneous intelligent systems that were built in isolation, and their cooperation may be necessary to achieve a new common goal. For example, intelligent personal agents, each of whom is responsible for scheduling the meetings of the person for whom it works, may need to cooperate to schedule a joint meeting of all the people for whom they work (Sen, Haynes, and Arora 1997). In other situations, several autonomous systems may work in the same environment, on different goals, and may need to coordinate their activities or to share resources. They may also benefit from cooperation. For example, information servers (possibly owned by different organizations), which need to store very large documents, may unite and decide that together they will keep only one copy of each document. They will need to decide where each shared document will be located.

Other examples of intelligent systems that need to cooperate and coordinate their activities include automated systems (agents) that monitor electricity transformation networks (Jennings 1995; Brazier et al. 1998) and multiagent systems that support hospital patient scheduling (Decker and Li 1998). Additional examples include computers that handle complex problems with distributed information that cannot be solved efficiently by a single system alone, and robots or teams of robotic systems (Balch and Arkin 1995) or helicopter pilot agents (Kaminka and Tambe 2000) that must cooperate in hostile environments to reach a common goal. Transportation centers that deliver packages can cooperate to reduce expenses (Sandholm and Lesser 1997) even though they are autonomous, since they are working in the same environment, and they can benefit from cooperation.

Problems of coordination and cooperation are not unique to automated systems; they exist at multiple levels of activity in a wide range of populations. People pursue their own goals through communication and cooperation with

other people or machines. Countries and nations must cooperate and coordinate their activities to increase the well-being of their people (Axelrod 1984). Since the earliest days people have used negotiation as a means to compromise to reach mutually beneficial agreements.

In social sciences there are two main approaches to the development of theories relating to negotiation. The first approach is the formal theory of bargaining (e.g., Roth 1979; Osborne and Rubinstein 1990), constituting a formal, game-theoretic approach that provides clear analyses of various situations and precise results concerning the strategy a negotiator should choose. However, this approach can be applied only to situations that satisfy very restricted assumptions. In particular, this approach assumes that the agents act rationally, have large computation capabilities, and follow strict negotiation protocols.

The second approach, the negotiation guides approach, comprises informal theories that attempt to identify possible general beneficial strategies for a negotiator. The works based on this approach advise a negotiator how to behave in order to reach beneficial results in a negotiation (see, e.g., Raiffa 1982; Fisher and Ury 1981; Druckman 1977; Karrass 1970; Johnson 1993; Hall 1993). These negotiation guides do not presuppose the strong restrictions and assumptions presented in the game-theoretic models. Applying these methods to automated systems is more difficult than using the first approach, however, since they use neither formal theories nor strategies.¹

In this book I adopt the formal game-theoretic approach. I present a strategic-negotiation model and apply it to various domains for coordination and cooperation. During strategic negotiations, negotiators communicate their respective desires and they compromise to reach mutually beneficial agreements. This strategic negotiation is a process that may include several iterations of offers and counteroffers.

A major goal in the development of the strategic-negotiation model has been to reduce overhead costs resulting from the time spent on planning and negotiation. This is necessary since one of the presumed difficulties in using negotiation as a way of reaching mutual benefit is that negotiation is a costly and time-consuming process and, consequently, it may increase the overhead of coordination (see Bond and Gasser 1988a). Thus, in the presence of time constraints, time spent on planning and negotiation should be taken into consideration.

The strategic-negotiation model provides a unified solution to a wide range of problems. It is appropriate for dynamic real-world domains. In this book, I describe the application of the strategic-negotiation model to data allocation

problems in information servers, resource allocation and task distribution problems, and the pollution allocation problem. In all these domains the strategic-negotiation model provides the negotiators with ways to reach mutually beneficial agreements without delay. The application of the strategic-negotiation model to high pressure, human crisis negotiations will also be presented.

The strategic-negotiation model has been applied to domains representing a wide range of possible applications. A resource allocation mechanism is important in any environment in which systems and/or people need to share resources. Examples include almost any workshop, such as a carshop. A conflict resolution mechanism concerning resource usage is also needed when organizations such as airlines need to share a scarce resource, for instance, airport resources. In addition, resource allocation is important in organizations and governmental systems. In such domains task allocation mechanisms are also essential. For example, task assignment is necessary in self-management maintenance groups of an airline (Hackman 1991) or in research teams.

The information servers' case represents increasingly important areas of computer and information sciences, namely, digital libraries and the Internet. In these domains servers with different interests may interact when there is no central controller, and thus use of the strategic-negotiation model for resolving conflicts seems very promising. The problem of data allocation in information-server environments is similar to the problem of deciding where to store parts, in an integrated national network of spare-parts inventories within a large company such as IBM.

Human negotiation during a crisis is a clear example of a situation where negotiation is costly and support is needed. Other examples include international negotiations and buying and selling on the Internet.

1.1 Rational Self-Interested Agents

The strategic-negotiation model was developed to address problems in distributed artificial intelligence (DAI), an area concerned with how automated agents can be constructed to interact in order to solve problems effectively. In the last few years, there have been several attempts to define an agent (e.g., Etzioni and Weld 1995; Wooldridge and Jennings 1995a; Foner 1993; Moulin and Chaib-Draa 1996; Jennings and Wooldridge 1998; Subrahmanian et al. 2000). For example, Etzioni and Weld (Etzioni and Weld 1995) require an agent to be goal-oriented, collaborative, flexible, and capable of making independent

decisions on when to act. In addition, they determined that an agent should be a continuously running process and be able to engage in complex communication with other agents, including people. It should automatically customize itself to the preferences of its user and to changes in the environment.

Subrahmanian et al. (2000) concentrate on the interaction of an agent with other agents and the environment. They define a software agent as a body of software that:

- provides one or more useful *services* that other agents may use under specified conditions;
- includes a *description* of the services offered by the software, which may be accessed and understood by other agents;
- includes the ability to *act autonomously* without requiring explicit direction from a human being;
- includes the ability to describe succinctly and declaratively how an agent determines *what actions to take* even though this description may be kept hidden from other agents; and
- includes the ability to *interact* with other agents—including humans—either in a cooperative or in an adverse manner, as appropriate.

There are two aspects to the development of agents: what is the architecture of each agent, and how do they interconnect, coordinate their activities, and cooperate. There are many approaches to the development of a single agent (see, e.g., the survey in Wooldridge and Jennings 1995a). These approaches can be divided into three main categories (Wooldridge and Jennings 1995b):

1. deliberative architectures
2. reactive architectures
3. hybrid architectures

A *deliberative agent architecture* is one that contains an explicitly represented, symbolic model of the world, and one in which decisions (e.g., about what actions to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbol manipulation. Examples of such architecture includes the intelligent resource-bounded machine architecture (IRMA) (Bratman, Israel, and Pollack 1988), HOMER (Vere and Bickmore 1990), Agent0 (Shoham 1993), Etzioni's softbots for UNIX environments (Doorenbos, Etzioni, and Weid 1997), the Kabash system (Chavez and Maes 1996), and

many others. The main criticism of this approach is that the computational complexity of symbol manipulation is very high, and some key problems appear to be intractable.

A *reactive architecture* is usually defined as one that does not include any kind of central symbolic world model and does not use any complex symbolic reasoning. One of the first architectures of this type is Brooks's *subsumption architecture* (Brooks 1985). Another such architecture is Maes's agent network architecture (Maes 1990). These types of agents work efficiently when they are faced with "routine" activities.

Many researchers suggest that neither a completely deliberate nor a completely reactive approach is suitable for building agents. They use *hybrid* systems, which attempt to combine the deliberate and the reactive approaches, for example, the PRS architecture (Georgeff 1987) and the TouringMachine (Ferguson 1992). Muller (1999) presents additional types of agents (e.g., layered agents, interacting agents) and tries to assist readers in deciding which agent architecture to choose for a specific application.

This book concentrates only on the second aspect of the development of agents: the ability of the agent to coordinate its activity with other agents and to cooperate with them. I provide an independent module for strategic negotiation, and thus, I am willing to adopt any definition or model of a single agent. My only assumptions are that the agents can communicate with each other using a predefined language, that they have some computation and memory resources, and that the negotiation module can be added to the agents. In some applications we will consider people as agents and assume that they can follow the negotiation protocols and strategies, if they wish.

An important issue in the development of a coordination and cooperation module is the level of cooperation among the agents: there are cooperative agents, which work toward satisfying the same goal, and there are environments where the agents are self-interested and try to maximize their own benefits.² There are intermediary cases where self-interested agents join together to work toward a joint goal. In this book we study the interactions among *self-interested*, *rational*, and *autonomous* agents. The agents do not share a common goal, and each agent has its own preferences and acts according to them.

Cooperation and coordination techniques may be required in various environments and situations where the agents act according to their own preferences and do not share a goal. For example, in situations where airplanes belonging to different airlines need to share the limited resources of the same airport, it is

necessary to find a mechanism that will give priority to planes with less fuel on board (Rosenschein and Zlotkin 1994). Other examples include an electronic market populated with automated agents representing different enterprises that buy and sell products (e.g., Chavez and Maes 1996; Fischer et al. 1996; Tsvetovaty and Gini 1996; Zeng and Sycara 1998); information servers that form coalitions for answering queries (Klusch and Shehory 1996); and autonomous agents that negotiate to reach agreements for providing a service by one agent to another (Sierra, Faratin, and Jennings 1997). In all these examples the agents are self-interested and try to maximize their own benefits. The strategic-negotiation model is applicable to these types of examples.

1.2 Motivating Examples

Two motivating examples are presented in order to illustrate different settings where negotiation among automated agents is required. These examples will be revisited throughout the book to illustrate basic concepts and the results that will be provided. The first example involves software agents, and the second mobile robots.

EXAMPLE 1.2.1 (DATA ALLOCATION IN LARGE DATABASES) There are several information servers in different geographical areas. Each server stores data, which has to be accessible by clients not only in its geographical area but also in other areas. The topics of interest to each client change over time, and the set of clients may also change over time. Periodically, new data arrive at the system and have to be located at one of the servers in the distributed system.

Each server is independent and has its own commercial interests. The servers would like to cooperate with each other in order to make more information available to their clients. Since each server has its own preferences regarding possible data allocations, its interests may conflict with the interests of some of the other servers.

A specific example of a distributed information system is the data and information system component of the earth-observing system (EOSDIS) of NASA (NASA 1996). It is a distributed system that supports archival data and distribution of data at multiple and independent data centers (called DAACs). The current policy for data allocation in NASA is static: each DAAC specializes in some topics. When new data arrive at a DAAC, the DAAC checks if the data are relevant to one of its topics, and, if so, it uses criteria such as storage cost to

determine whether or not to store the data in its database. The DAAC communicates with other DAACs in those instances in which the data item encompasses the topics of multiple DAACs, or when a data item presented to one DAAC is clearly in the jurisdiction of another DAAC, and then a discussion takes place among the relevant DAACs' managers. However, this approach does not take into consideration the location of the information clients, and this may cause delays and excessive transmission costs if data are stored far from their potential users. Moreover, this method can cause rejection of data if they do not fall within the criteria of any DAAC, or if they fall under the criteria of a DAAC that cannot support this new product because of budgetary problems.

EXAMPLE 1.2.2 (AGENTS ON MARS) NASA has embarked on a scientific mission to Mars that involves sending several mobile robots. The European Space Agency (ESA) has also sent several mobile robots to Mars. Both NASA's and ESA's robots work in the same environment. The missions of the robots involve collecting earth samples from different locations and at different depths. A robot will need one or more digging tools to complete its tasks. The tools were sent to Mars by a third company, which charges NASA and ESA according to their use of the equipment. As a result, it might be beneficial for the two groups of robots to share resources.

It may also be the case, say, that the NASA group's antenna is damaged during landing, and it is expected that communications between NASA's center and its robots on Mars will be down for repairs for one day. NASA can use a weaker and less reliable backup line, but this would mean diverting this line from other costly space experiments, and consequently the expense of using this line is very high. NASA would like to share the use of the ESA line during the one-day period so that it can conduct its planned research program. Only one group can use the line at a time, and that line will be in use for the entire duration of the particular experiment. A negotiation ensues between the two labs over division of use of the ESA line, during which time the ESA has sole access to the line, and NASA cannot conduct any of its experiments (except by use of the very expensive backup).

Given the 1997 success of NASA's Pathfinder on Mars (Golombek et al. 1997), and the 1994 success of DANTE II in exploring the crater on the Mt. Spurr volcano in Alaska, it seems that such scenarios may be realistic in the near future (see also Berns 1998). NASA's current plan is to send a pair of wheeled robots to search for evidence of water on Mars in 2003.

1.3 Characteristics That Differentiate Negotiation Protocols

Evaluation of the results of negotiations in the systems considered is not easy. Since the agents are self-interested, when a negotiation is said to be successful we must ask “successful for whom?” since each agent is concerned only with its own benefits or losses from the resolution of the negotiation. Nevertheless, there are some parameters that can be used to evaluate different protocols.

Only those protocols that satisfy the following conditions are considered.

Distribution: The decision-making process should be distributed. There should be no central unit or agent required to manage the process.

Symmetry: The coordination mechanism should not treat agents differently in light of nonrelevant attributes. In the situations we consider, the agents’ utility functions and their role in the encounter are the relevant attributes. All other attributes, such as an agent’s color, name, or manufacturer, are not relevant. That is, symmetry implies that given a specific situation, the replacement of an agent with another that is identical with respect to the above attributes will not change the outcome of the negotiation.

The first requirement is desirable since we consider self-interested agents and it may be difficult for such agents to agree on a centralized controller that will be a fair mediator. In addition, a centralized controller may become a performance bottleneck. The symmetry restriction will encourage the designers of the agents to adopt the negotiation protocol.

The following parameters will be used to evaluate the results of the negotiation that are presented in this book.

Negotiation time: Negotiations that end without delay are preferable to negotiations that are time-consuming.

It is assumed that a delay in reaching an agreement causes an increase in the cost of communication and computation time spent on the negotiation. We want to prevent the agents from spending too much time on negotiation resulting in not keeping to their timetables for satisfying their goals.

Efficiency: It is preferred that the outcome of the negotiations will be efficient. It increases the number of agents that will be satisfied by the negotiation results and the agents’ satisfaction levels from the negotiation results.

Thus it is preferable that the agents reach *Pareto optimal* agreements.³ In addition, if there is an agreement that is better for all the agents than opting out, then it is preferred that the negotiations will end with an agreement.

Simplicity: Negotiation processes that are simple and efficient are preferable to complex processes. Being a “simple strategy” means that it is feasible to build it into an automated agent. A “simple strategy” is also one that an agent will be able to compute in a reasonable amount of time.

Stability: A set of negotiation strategies for a given set of agents is stable if, given that all the other agents included in the set are following their strategies, it is beneficial to an agent to follow its strategy too. Negotiation protocols that have stable strategies are more useful in multiagent environments than protocols that are unstable. If there are stable strategies, we can recommend to **all** agent designers to build the relevant strategies into their agents. No designer will benefit by building agents that use any other strategy.

Money transfer: Money transfer may be used to resolve conflicts. For example, a server may “sell” a data item to another server when relocating this item. This can be done by providing the agents with a monetary system and with a mechanism for secure payments. Since maintaining such a monetary system requires resources and efforts, negotiation protocols that do not require money transfers are preferred.

1.4 An Overview of This Book

This book is organized as follows. Chapter 2 introduces the reader to the main components of the strategic-negotiation model. It presents Rubinstein’s protocol of alternating offers, defines basic concepts such as agreements and strategies, and introduces the concept of equilibrium that will be used to identify strategies for the negotiating agents.

Chapter 3 presents the application of the strategic-model to the data-allocation problem. It considers situations characterized by complete as well as incomplete information, and proves that the negotiation model yields better results than the static allocation policy currently used for data allocation for servers in distributed systems.

Chapter 4 considers resource allocation in environments where agents need to share a common resource. It considers situations with both complete and

incomplete information, single encounters and multiple encounters. It will be shown that in all the situations considered in this chapter, the negotiations end no later than in the second time period, and usually with an agreement.

Chapter 5 continues to study the problem of resource allocation where the agents have goals with deadlines that they need to meet. Emphasis is placed on the issue of multiple attributes of the negotiation and it is shown that in these settings the negotiation ends with no delay. Simulation results reveal that our mechanism performs as well as a centralized scheduler and also has the property of balancing the resources' usage.

Chapter 6 presents the application of the strategic model to the task distribution problem. Here again negotiation ends with no delay.

Chapter 7 considers the pollution sharing problem in which plants should reach agreements on pollution reduction because of external factors such as weather. The application of the strategic-negotiation model to the problem is presented and a comparison is made with market-based methods in situations where the agents have incomplete information.

Chapter 8 presents the application of the strategic-negotiation model to a hostage crisis situation.

Chapter 9 presents a detailed survey of other game-theory and economics-based models of cooperation and compares them with the strategic-model of negotiation.

The book concludes with future directions for automated negotiation.

The appendix includes an annotated bibliography of suggested background readings and a glossary of the notations used in the book.

1.5 Game Theory Concepts

This introductory chapter informally describes the main concepts of game theory. Readers who are familiar with game theory can skip this chapter.⁴ Some of the general concepts described in this section will be redefined and used in the strategic-negotiation model.

Game theory is the study of decision making in multiperson situations, where the outcome depends on everyone's choice. The goal of each participant is to achieve well-defined objectives, while taking into account that the other participants are doing the same and that all their actions affect each other. This is in contrast to decision theories and the theory of competitive equilibrium that are used in economics, in which the other participants' actions are considered

as an environmental parameter, and the effect of the decision maker's actions on the other participants is not taken into consideration.

Game theory, as it is used in this book, is a modeling tool, not an axiomatic system. The main concepts of game theory will be described using a simplified case of a hostage crisis scenario that is presented in (Kraus and Wilkenfeld 1993). The scenario is based on the hypothetical hijacking of a commercial airliner enroute from Europe to India and its forced landing in Pakistan. The passengers are predominantly Indian and the hijackers are known to be Sikhs. The hijackers demand the release of up to 800 Sikh prisoners from Indian security prisons (see Kraus et al. 1992). The three parties must consider several possible outcomes: India or Pakistan launches military operations to free the hostages; the hijackers blow up the plane with themselves aboard; India and the Sikhs negotiate a deal involving the release of security prisoners in exchange for the hostages; Pakistan and the Sikhs negotiate a safe passage agreement; or the hijackers give up. The details of the negotiation process will not be considered here since they are not needed to demonstrate the main concepts of game theory. The negotiation process is detailed in chapter 8.

1.5.1 Describing a Game

The essential elements of a game are *players, actions, information, strategies, payoffs, outcome, and equilibria*. The players, actions, and outcomes are collectively referred to as the *rules of the game*, and the modeler's objective is to use the rules of the game to determine the equilibrium.

The *players* are the individuals who make decisions. It is assumed that each player's objective is to maximize the expected value of his own payoff, which is measured in some *utility* scale. In the hostage crisis scenario, the players are India, Pakistan, and the Sikhs. Each player has a set of objectives, which we identified, and a certain number of utility points is associated with each (see Kraus et al. 1992; Wilkenfeld et al. 1995). Passive individuals, like the UK in this example, who may observe the situation without trying to change anyone's behavior, are not players.

There are essentially three ways to present a social interaction as a game.

Extensive form: The most complete description is the *extensive form*. It details the various stages of the interaction, the conditions under which a player has to move, the information an agent holds at different stages, and the motivation of the players.

Strategic form: More abstract is the *strategic form* (or *normal form*) representation of a game. Here one notes all possible strategies of each agent together with the payoffs that result from strategic choices of all the players. In strategic forms, many details of the extensive form have been omitted.

Coalitional form: The *coalitional form* (or *characteristic function form*) is a description of social interactions where binding agreements can be made and enforced. Binding agreements allow groups of players, or coalitions, to commit themselves to actions that may be against the interest of individual players once the agreement is carried out.

The components of the strategic form will be presented first and then the extensive form will be described. The coalitional form is discussed in, for example, (Kahan and Rapoport 1984).

1.5.2 Strategic Games

In a strategic game each player chooses his final plan of action, and these choices are made simultaneously.⁵ The model consists of a finite set of N players. An *action* or a *move* by player i , denoted as a_i , is a choice he can make. Player i 's action set, $Act_i = \{a_i\}$, is the entire set of actions available to him. An action profile is an order set $a = \{a_j\}$, ($j = 1, \dots, N$), where a_j is an action of player j . The set of all possible action profiles is denoted as \mathcal{A} . For example, in a simplified scenario of the hostage crisis, we have $Act_{Ind} = \{Operation, Deal, Nothing\}$, $Act_{Pak} = \{Operation, Safe_passage, Nothing\}$ and $Act_{Sik} = \{Blow, Deal, Safe_passage\}$. A possible action profile is $(Deal, Nothing, Deal)$, where India and the Sikhs reach an agreement, and Pakistan chooses to do nothing. For each player i , there is a *payoff function* $U_i : \mathcal{A} \rightarrow \mathbb{R}$ (also called a *utility function*), which represents its preferences on the set of action profiles.⁶

A strategic game in which there are two players can be described easily in a table like that given in figure 1.1. One player's actions are identified with the

		India	
		Deal	Op
Sikh	Deal	2 1	-2 0
	Blow	-1 -2	0 -3

Figure 1.1

An example of a two-player strategic game in which each player has two actions.

rows and the other with the columns. The two numbers in the box formed by row r and column c are the players' payoffs when the row player chooses r and the column player chooses c . The top number is the payoff of the column player. In the simple example of figure 1.1 the row player is the Sikhs and the column player is India. India can choose between trying to reach a deal (*Deal*) and launching an operation (*Op*). The Sikhs can choose between trying to reach a deal (*Deal*) and blowing up the plane (*Blow*). If both India and the Sikhs choose *Deal*, the utility of India⁷ is 1 and that of the Sikhs is 2. If, for example, the Sikhs choose *Deal* and India chooses *Op*, then the utility of the Sikhs is -2 and that of India is 0.

The solution concept most commonly used to predict the outcome of a game in game theory is Nash equilibrium (Nash 1953; Luce and Raiffa 1957). This notion captures a *steady state* of a strategic game, in which each player holds the correct expectation about the other players' behavior and acts rationally.⁸

An action profile a is a *Nash equilibrium* of a strategic game, if each player i does not have a different action yielding an outcome that it prefers to that generated when it chooses a_i , given that every other player j chooses a_j . To put it briefly: no agent can profitably deviate, given the actions of the other players.

For example, the game in figure 1.1 (*Deal, Deal*) is a Nash equilibrium. If India changes to *Op*, then, given that the Sikhs choose *Deal*, its utility will decrease to 0. Similarly, if the Sikhs choose *Blow*, given that India chooses *Deal*, their utility will be reduced to -1 . Note that *Deal* is always better to India than *Op*; however, *Blow* is better for the Sikhs than *Deal* when India launches a military operation (*Op*).

1.5.3 Games in Extensive Form

The strategic form is too abstract to model multiple situations, including a negotiation process. The extensive form, on the other hand, is the most explicit description of a game. It notes the sequence of moves, all possible states of information, and the choices at different stages available to all players of the game. That is, the model allows us to study solutions in which each player can consider his plan of action, not only at the beginning of the game, but also at any point of time at which he has to make a decision.

An extensive-form game is a tree together with functions that assign labels to its nodes and edges. An extensive game based on a simple scenario of the hostage crisis is presented in figure 1.2. In this scenario, India acts before the Sikhs. It needs to choose between offering a deal or launching an operation (*Op*).

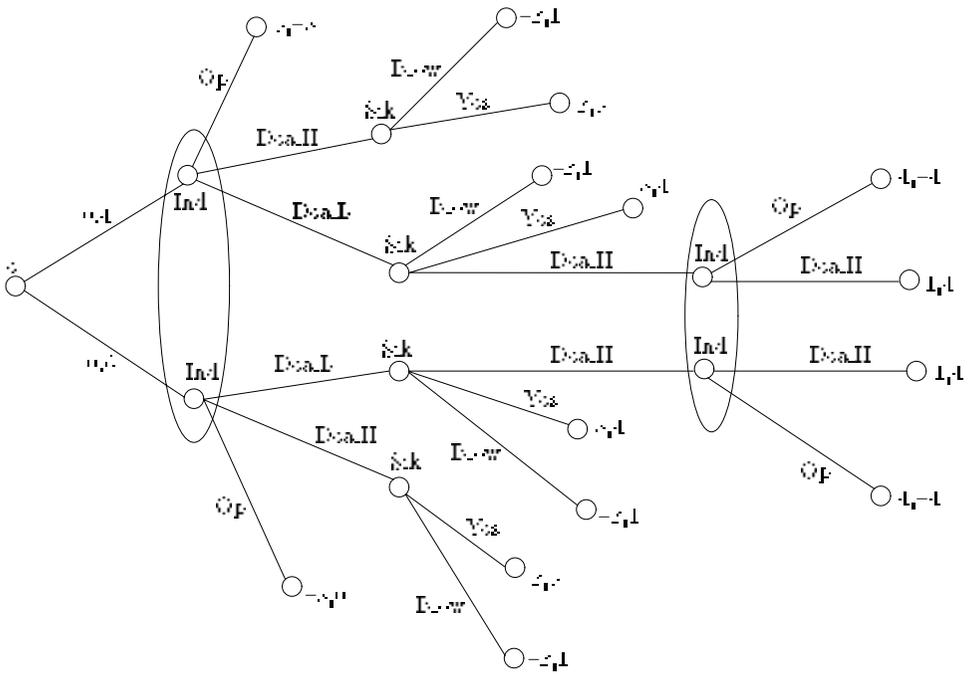


Figure 1.2
An example of an extensive-form game.

The deal can be high (*DealH*) or low (*DealL*), in which India will release a high number of Sikh prisoners or a low number of Sikh prisoners, respectively. India prefers *DealL* to *DealH*, while the Sikhs prefer *DealH*. If India chooses *Op* the game ends: if the weather conditions are good, then the operation will succeed and India’s utility will be 5 and the Sikhs’ utility will be -3 . However, if the weather conditions are bad, the operation will fail and India and the Sikhs’ utilities will be -3 and 0, respectively. India does not know the weather conditions when making a decision. If India chooses *DealH* or *DealL*, then it is the Sikhs’ turn to move. They can either accept the deal (*Yes*) or blow up the plane (*Blow*), or if India chose *DealL* they can make a counteroffer (*DealH*). In the first two cases the game ends. If the Sikhs choose *DealH*, then it is India’s turn again. It can either accept the deal (*Yes*) or launch an operation. The tree in figure 1.2 specifies all these details.

In general, the tree of an n -person extensive-form game consists of nodes (the points) and edges that connect between nodes. The leaves of the tree

(i.e., nodes without edges that start there) are called *terminal nodes* and represent possible ways that the game could end. Each possible sequence of events that could occur in the game is represented by a path of edges from the root to one of these terminal nodes. Each nonterminal node has a *player label*. The nodes with a player-label i are *decision nodes* that are controlled by player i . For example, in figure 1.2, it is India's turn to make a decision in the nodes labeled *Ind*. The labels on the edges that start at a decision node specify the possible moves of the player of the node. For example, consider in figure 1.2 the three edges that start at the highest node at the left that is labeled by *Ind*. These edges are labeled with *Op*, *DealH*, and *DealL*, respectively, specifying that India can choose between launching a military operation, or offering deals.

There may be events that are determined by chance, that is, they are not under the control of any of the players (e.g., the weather conditions in the hostage crisis scenario). They are represented by nodes labeled c and are called *chance nodes*. Each edge that starts at a chance node has a label that specifies its probability. At each chance node, these chance probabilities of the edges are nonnegative numbers that sum to 1. In figure 1.2, the root of the tree (i.e., the first node) is a chance node. The edges leaving it are labeled 0.4 and 0.6. Intuitively, this means that there is a probability of 0.4 that the weather will be good, and a probability of 0.6 that the weather will be bad.

Each terminal node of the tree is associated with a label that specifies a vector of n numbers that represent the payoffs of the players.⁹ For example, in figure 1.2 there are 14 terminal nodes. The pair of numbers at each of the terminal nodes represents the payoffs that India and the Sikhs would get if the events of the path ending at this node will occur. For example, if the weather is good and India launches a military operation, then the terminal node labeled 5, -3 (the left top node) will be reached, indicating that India's utility in this case is 5 and the Sikhs' utility is -3 .

The extensive form can also take into consideration the uncertainty of the players about the game. A player may be unable to observe the moves of opponents or the chance moves. Since actions lead from nodes to other nodes, a player who cannot observe the action of another player will not know at which decision node he is located. For example, in India's first move in the hostage crisis scenario, India doesn't know the weather conditions, that is, it doesn't know whether it is in the upper node or the lower node. To show that a player may not be able to distinguish between nodes, one joins the decision nodes that are indistinguishable in a set, called an *information set*. In diagramming the game trees, information sets are indicated by joining the particular nodes

with a circle. In figure 1.2, the first two nodes of India are joined together by a circle, indicating that India cannot distinguish between them. An information set contains only nodes of the same players, and each node can belong to only one information set. In addition, the possible moves of a player in all the nodes that belong to the same information set are exactly the same.

A sequence of events leading to a node is its *history*. It is generally assumed that the players in an extensive-form game have *perfect recall*. This assumption 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 in the history.

A *strategy* for a player in an extensive-form game is a function that specifies an action for every history after which the player chooses an action. Thus a strategy specifies for a player which action to take at each of its decision nodes. A strategy profile is an ordered set of strategies, one for each player. For example, a possible strategy for India is for it to launch a military operation at its turn to move.

As in the strategic games, the Nash equilibrium concept is also used for predicting the outcome of extensive-form games. A strategy profile $F = (f_1, \dots, f_N)$ is a *Nash equilibrium* of an extensive game, if each player i does not have a different strategy yielding an outcome that it prefers to that generated when it chooses f_i , given that every other player j chooses f_j . For example, the strategies profile in which India always launches a military operation (chooses *Op*) whenever it is her turn to move, and the Sikhs blow up the plane, whenever it is their turn to move, is a Nash equilibrium. If India moves first and chooses *Op*, the Sikhs' action is not relevant anymore. Thus deviating from *Blow* will not improve their expected utility. Given that the Sikhs' strategy is *Blow*, if India will deviate and will not choose *Opt*, but rather choose *DealH* or *DealL*, the game will terminate with *Blow*. India's expected utility in such a case is -2 , which is lower than its expected utility from *Opt*, which is $0.4 \times 5 + 0.6 \times -3 = 0.2$.

This equilibrium demonstrates that the use of Nash equilibrium in extensive-form games may lead to an absurd equilibrium, since accepting *DealH* or *DealL* by the Sikhs yields a higher utility than blowing up the plane. Additional concepts of equilibrium will be discussed and applied to the strategic model.