Instituto Tecnológico Autónomo de México
Maestría en Finanzas
Economía Financiera (Eco-44105), 2015
Decision Trees

Ricard Torres

1 Unipersonal decision trees

A *unipersonal decision tree* is a set of *nodes*, some of which are connected by *edges*. There is a distinguished node, the *root*, such that there is a unique path that connects any other node with it. Nodes can be either *decision nodes*, in which the decision maker must choose an alternative, *nature or chance nodes*, that represent a random event, or *terminal nodes*, that are labelled with the utility the decision maker attains if he or she reaches it.

At each decision node, the decision maker must choose one of a set of possible *alternatives or actions*; we associate each such action with an edge coming out (ie, away from the root) of the decision node.

The decision nodes are partitioned into *information sets*: an information set is a set of decision nodes that the decision maker is unable to distinguish. That is, when one of the nodes is reached, the decision maker knows he/she is somewhere within the information set, but does not know at which of the nodes in it. The actions available at any two decision nodes that belong to the same information set are the same. We will also make the convention that actions available at different information sets are considered different (even though they might correspond to the same physical act, say "invest 1 million dollars," the fact that the information when taking the two actions is different justifies our convention). By definition, all decision nodes belong to some information set, even if it contains only one node.

A node B *follows* another node A if A is in the path that joins B with the root; we also say that A *precedes* B. Similarly, an action a precedes a node B if there is an edge associated with action a in the path that joins B with the root (note that more than one edge may be associated with the same action, if they follow decision nodes that are in the same information set). The edges that immediately follow a nature node correspond to the different results of the random event, and they are labelled with the corresponding probability.

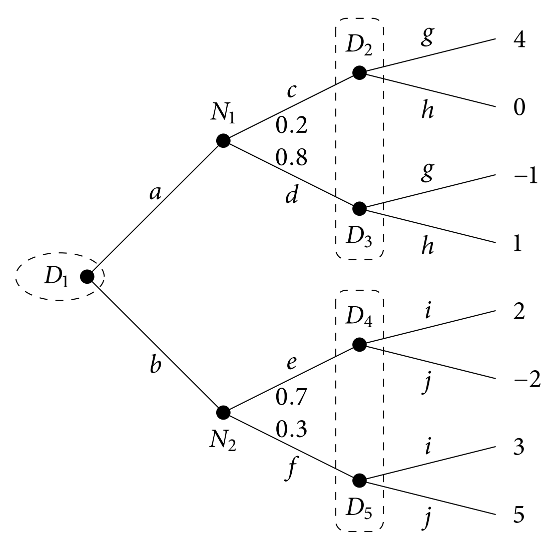


Figure 1: In this unipersonal decision tree, there are five decision nodes, labelled D_1 (the root) through D_5, two nature nodes, N_1 and N_2, and eight terminal nodes, labelled with the corresponding utilities. There are three information sets: $\{D_1\}$, $\{D_2, D_3\}$, and $\{D_4, D_5\}$. Note that the actions available at nodes in the same information set are the same. Note also that the edges coming out of nature nodes are labelled with their probabilities. Node N_1 precedes node D_3, and action a also precedes node D_3.

2 Multipersonal decision trees

A *multipersonal decision tree* is a decision tree in which there is more than one decision maker, so each decision node belongs to one of the possible decision makers. As in the case of unipersonal decision trees, all the decision nodes belonging to a particular decision maker are partitioned into information sets, which express the knowledge of that individual at each instance in which he or she is called to choose. Each terminal node is labelled with a *vector of payoffs*, one for each decision-maker.

We say that a (uni- or multipersonal) decision tree has *perfect information* if each information set is a singleton (contains only one element), so that each individual knows precisely what the past history of choices is when he or she is called to choose a move.

Let us look at some examples:

Figure 2: This is a simple multipersonal decision tree with two decision-makers, 1 and 2, each of which has a single decision node. For simplicity, we just label each decision node with the number of the corresponding decision maker. There is perfect information, because no information set contains more than one decision node.

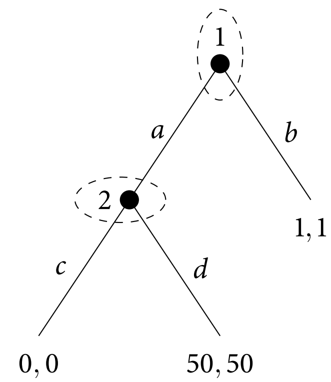


Figure 3: This is a multipersonal decision tree with two decision-makers, 1 and 2, both of which have multiple decision nodes. The root of the tree is a chance node. Since each of the information sets of individual 2 has two decision nodes, information is not perfect.

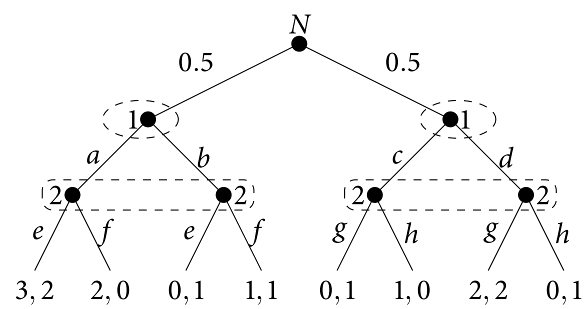
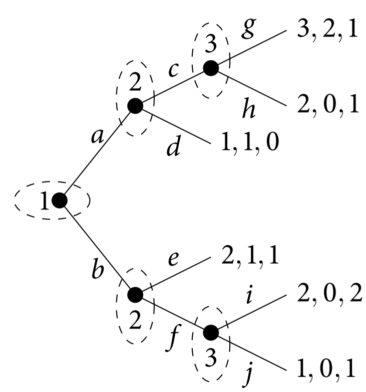


Figure 4: This is a multipersonal decision tree with three decision-makers, 1, 2, and 3. The tree has perfect information, because each information set contains a single node.



3 Strategies

A *strategy* is a rule that associates with each decision node one of the actions available there. We impose the restriction that the action chosen at any two decision nodes that belong to the same information set must be the same, otherwise we would have an inconsistency with our interpretation of what an information set means.

For instance, in the tree depicted in Figure 1, a possible strategy consists of choosing a at node D_1, h at nodes D_2 and D_3, and i at nodes D_4 and D_5. In order to represent this more compactly we may order the information sets, and then represent a strategy as an *ordered vector of actions*, one for each information set. Thus, the previous strategy would be represented as (a, h, i). In multipersonal decision problems, we usually represent this as an ordered sequence of choices with no commas, eg *ahi*, in order to distinguish it from the vector of strategies of *all* of the individuals.

In the two-person decision tree of Figure 3, a strategy for player 2 specifies a move at each of its information sets, for instance *eh*. Analogously, in the three-person decision tree of Figure 4, a possible strategy for player 3 is, for example, *hi*.

By construction, if there are no chance nodes, a choice of a strategy by each decision-maker gives rise to a *unique path* from the root to precisely one of the terminal nodes. The corresponding vector of utilities indicates the payoffs the individuals derive from that choice of strategies.

If there are chance nodes, a choice of a strategy by each decision-maker gives rise to a certain *probability distribution* over terminal nodes. For instance, in the unipersonal decision tree depicted in Figure 1, with the strategy (a, h, i) the terminal node that has utility 0 is reached with probability 0.2, and the terminal node that has utility 1 is reached with probability 0.8. The strategy (b, g, i) leads to the terminal node that has utility 2 with probability 0.7, and to the terminal node that has utility 3 with probability 0.3.

Consider now the decision tree of Figure 3. The pair of strategies (ac, fh) by both individuals gives rise to the probability distribution that places probability 1/2 on $(2, 0)$, and probability 1/2 on $(1, 0)$, so that the expected utilities of the individuals from that choice of strategies are $(1.5, 0)$. In general, when there are chance nodes a choice of a strategy for each individual allows us to compute their corresponding *expected utilities*.

When we select a strategy for each of the individuals in a multipersonal decision tree, the resulting vector of strategies is also called a *strategy profile*.

4 Perfect information and backward induction

Consider a decision tree with perfect information in which the total number of nodes is finite. The procedure of *backward induction* leads to a strategy profile in any such decision tree that has the following property: each player acts optimally at each decision node, under the assumption that, at all decision nodes that follow it, the corresponding players will also do so.

The procedure can be described as follows. Consider a decision tree with n decision nodes. Start from a decision node that is not followed by other decision nodes (this is always possible because of finiteness and the nonexistence of cycles in the tree), and select one of the alternatives in which the player obtains the highest payoff. Next, substitute that decision node by the (possibly expected) payoff vector that results from the choice made. This results in a decision tree with $n - 1$ decision nodes. Iterate the procedure until we are left with a single decision node.

For example, consider the two-person decision tree of Figure 2. The decision node of player 2 is not followed by any other decision node. If 2 chooses c she obtains 0, if she chooses d she obtains 50, therefore the latter is the optimal choice. Given that, if player 1 chooses a the resulting payoffs will be $(50, 50)$, and if player 1 chooses b the players will obtain $(1, 1)$, so the optimal solution for 1 is to choose a. Concluding, the *solution by backward induction* of this tree is (a, c), and the corresponding payoffs $(50, 50)$. Note that this is a Nash equilibrium, but there is also another Nash equilibrium, which is *not* a backward induction solution, in which the players choose (b, c) and obtain $(1, 1)$.

Consider now the decision tree of Figure 4. By backward induction, player 3 will choose i in the decision node at the bottom. In the decision node at the top, player 3 is indifferent between g and h, and therefore either choice is valid according to the procedure. If 3 chooses *gi*, then the procedure of backward induction will imply that 2 chooses *ce*, and therefore 1 chooses *a*, yielding the backward induction solution (a, ce, gi). On the other hand, if 3 chooses *hi*, then 2 will choose *de* and 1 will choose *b*, yielding the backward induction solution (b, de, hi). This illustrates how there may be multiple backward induction solutions if a player is indifferent at some step.

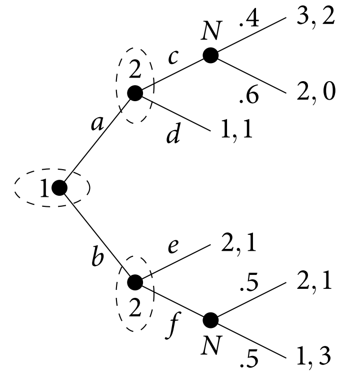


Figure 5: At the top node, player 2 chooses *d*, because *c* has an expected payoff $.8 < 1$. So we substitute that decision node with the payoffs $(1, 1)$ for both players. At the bottom node, 2 chooses *f* and obtains an expected payoff 2, so we substitute this decision node by the expected payoffs $(1.5, 2)$. So the choice of 1 is between *a*, with which she obtains 1, and *b*, that yields 1.5. The latter is the optimal choice, so the unique backward induction solution is (b, df).

Note that, if information is not perfect, one can construct easy examples in which the procedure of backward induction does not lead to a solution.

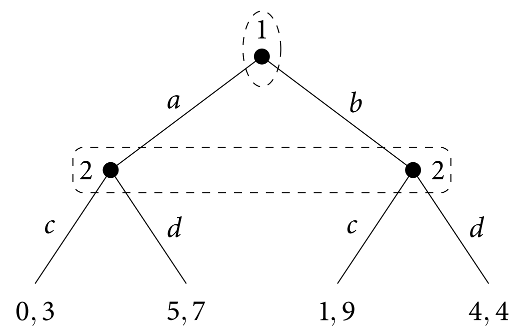


Figure 6: This tree has imperfect information, because player 2 has an information set with two nodes. If player 2 knew she was at the left node, she would like to choose *d*, and if she knew she was at the right one, she would like to choose *c*. But since 2 does not know, she cannot choose, so the backward induction procedure cannot be applied.

5 Subgames and subgame perfect equilibria

The solution by backward induction is of limited applicability, because it is only well defined for decision trees that are finite and have perfect information. We are going to show here that this concept can be generalized.

In a (uni- or multipersonal) decision tree, a *subgame* is a (nature or decision) node and all of the nodes and edges that follow it, with the requirement that, if a decision node belongs to the subgame, then the entire information set that contains that decision node must also be included. In particular, a decision node that belongs to an information set in which there are other decision nodes cannot be the starting point of a subgame. By construction, if we detach the subgame from the decision tree, it becomes a decision tree on its own, and our objective here is to analyze this new decision tree as a separate entity. Since the original decision tree itself can be trivially viewed as a subgame, we say that a subgame is *proper* if it is different from it.

The underlying idea is that, when the play enters a subgame, it stays in it and all players know it, so the subgame can be analyzed as a unit. Whatever the players chose in the past does not matter: as soon as the play enters the subgame, the past plans of the players are "sunk," and only the choices within the subgame matter as regards the output.

A strategy profile for the players in the decision tree consists of the choice of a possible action for each information set of each player. Therefore, a strategy profile in the decision tree induces a strategy profile in the subgame, when we view the latter as a separate entity.

The Nobel prize in Economics *Reinhard Selten* introduced the following concept: a strategy profile for the players in a decision tree is said to be a *subgame-perfect equilibrium* if the strategies it induces in any subgame constitute a Nash equilibrium of the subgame when viewing the latter as a separate entity.

This concept reduces to the backward induction solution in games with perfect information, and Selten proved that *all* finite decision trees have at least one subgame-perfect equilibrium (possibly resorting to mixed strategies).

Figure 7: In this decision tree there are two (proper) subgames: one begins at the decision node of 2 that follows action a of 1, and the other at the decision node of 2 that follows action b. At the former subgame, the only Nash equilibrium is (f, c), with payoffs $(3, 2)$. At the latter subgame, the equilibrium is the optimal choice g of 2, with payoffs $(0, 2)$. Therefore, in the overall tree the optimal strategy of 1 is a. So this tree has a unique subgame-perfect equilibrium, (af, cg), with payoffs $(3, 2)$.

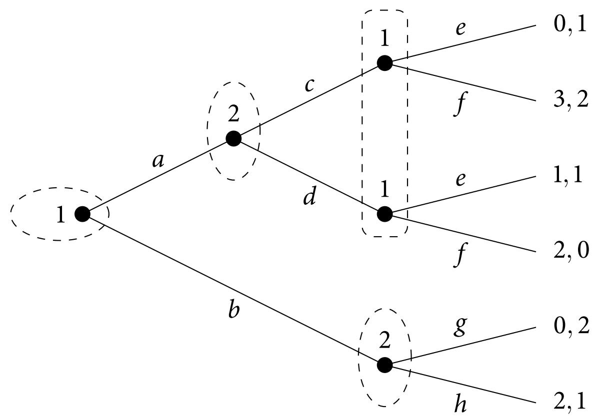
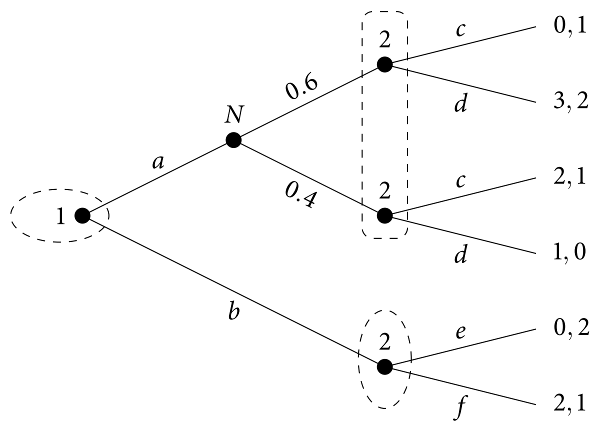


Figure 8: In this decision tree there are two (proper) subgames: one begins at the nature node, and the other at the decision node of 2 that follows action b of 1. At the former subgame, the optimal choice of 2 is d, with expected payoffs $(2.2, 1.2)$. At the latter subgame, the optimal choice of 2 is e, with payoffs $(0, 2)$. Therefore, in the overall tree the optimal strategy of 1 is a. There is a unique subgame-perfect equilibrium, (a, de), with expected payoffs $(2.2, 1.2)$.



Appendix: Perfect recall

Besides the requirements in the definition of decision trees, we almost always impose an additional restriction: all players have *perfect recall* of their own past history of play.

We say that a decision tree satisfies *perfect recall for player i* if, whenever a decision node of i follows a particular action of i, then all of the other decision nodes that are in the same information set follow the same action. Note that in decision trees with perfect information all players have perfect recall.

Since this concept involves only the information sets and actions of a particular player i, it is easiest to illustrate it with unipersonal decision trees: let us consider a couple of examples.

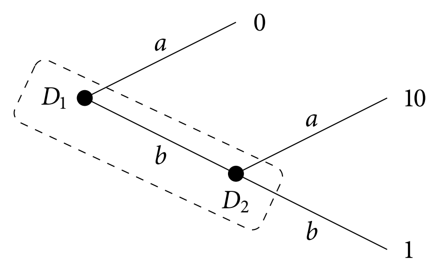


Figure 9: A unipersonal decision tree with imperfect recall for the player. Node D_2 follows action b of the player, but node D_1 is in the same information set and does not follow b.

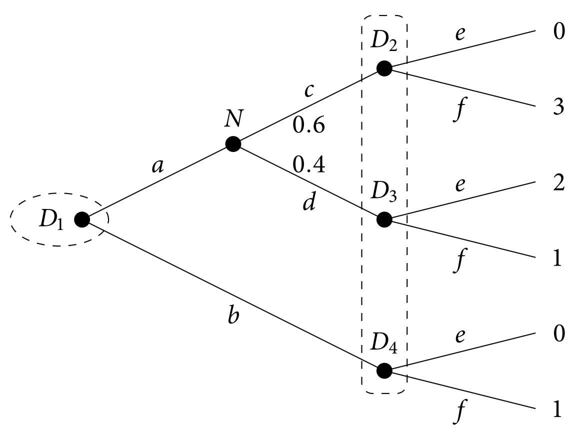


Figure 10: This is another unipersonal decision tree with imperfect recall for the player. For example, node D_2 follows action a of the player, but node D_4 does not, even though it is in the same information set.

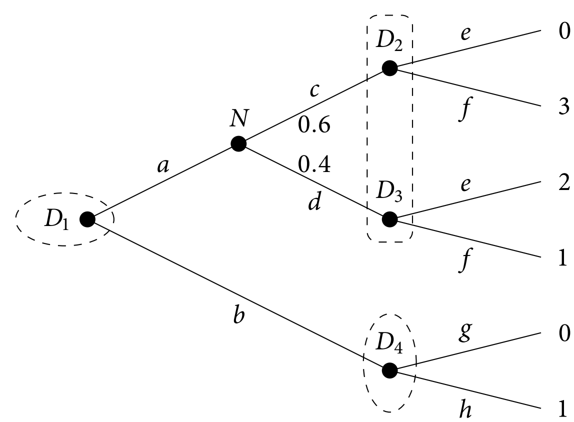


Figure 11: This is a modification of the previous unipersonal decision tree that *does* satisfy perfect recall for the player. Now each node in the same information set is preceded by the same actions of the player.

In the decision tree of Figure 9, the decision-maker cannot distinguish between the nodes D_1 and D_2, so, when at D_2 she cannot tell whether she has already taken action b or not. On the other hand, in the decision tree of Figure 10, the decision-maker cannot distinguish between the nodes D_2 and D_4, so again she cannot tell whether she has taken action b or not. Hence the denomination of "imperfect recall."

In order to see that with imperfect recall something is not quite right, consider the example of Figure 9. Since both nodes belong to the same information set, when choosing a the decision-maker reaches the terminal node with utility 0, and when choosing b she reaches the one with utility 1: the terminal node with utility 10 is unreachable. In finite trees in which all players have perfect recall (and all outcomes of chance nodes have strictly positive probability), for any given terminal node there always exists a strategy profile under which this node is reached with strictly positive probability.