Extensive-Form Games: Introduction

Introduction

When we model a strategic economic situation we want to capture as much of the relevant detail as tractably possible. A game can have a complex temporal and information structure; and this structure could well be very significant to understanding the way the game will be played. These structures are not acknowledged explicitly in the game's strategic form, so we seek a more inclusive formulation. It would be desirable to include at least the following: 1) the set of players, 2) who moves when and under what circumstances, 3) what actions are available to a player when she is called upon to move, 4) what she knows when she is called upon to move, and 5) what payoff each player receives when the game is played in a particular way. Components 2 and 4 are additions to the strategic-form description; component 3 typically involves more specification in the extensive form than in the strategic form.

We can incorporate all of these features within an *extensive-form* description of the game. The foundation of the extensive form is a *game tree*. First I'll discuss what a *tree* is and then describe the additional specifications and interpretations we need to make in order to transform it into a full-fledged extensive-form description of the game. We'll discuss additional restrictions upon the information structure to reflect a common assumption, viz. *perfect recall*, which asserts that players don't forget. We'll discuss the relation between this traditional, graph-theoretic exposition and a more recent one based on the *arborescence*. After that we'll learn how to ease our analysis of complicated games by breaking them up into simpler *subgames*.

Game Trees

We'll construct our definition of a tree by first defining more primitive graph-theoretic entities. (Iyanaga and Kawada [1980: 234–235]). A *graph* $G=(V,E)$ consists of a finite set V of *vertices* and a finite set E of *edges* of V. An edge $e\in E$ is an unordered pair of distinct elements of V; e.g. the edge $e=(v,v')$ *joins* the two vertices $v,v'\in V$.

Consider an alternating sequence of vertices and edges of a graph G, $P=\{v_0,e_1,v_1,e_2,\ldots,e_n,v_n\}$. P is a *path* if 1 each edge e_i joins its neighbors v_{i-1} and v_i and 2 no edge is encountered more than once (i.e. $e_i\neq e_j$ when $i\neq j$).[1] In this case we say that v_i and v_j, for $i,j\in\{1,\ldots,n\}$, are *connected* by P and that P *runs from* v_0 *to* v_n.[2] (Note that a path may encounter the same vertex more than once.) A path is *closed* if it begins and ends with the same vertex, i.e. $v_0=v_n$. A closed path which never reencounters a vertex, except for the first, is called a *circuit*. I.e. a closed path P is a circuit if $v_i\neq v_j$ whenever $i\neq j$ and $\{i,j\}\neq\{0,n\}$. In other words a circuit is a single "loop." A graph G is *connected* if for every pair of distinct vertices $v,v'\in V$, $v\neq v'$, there exists a path in G which connects v and v'.

Now we're ready to define a tree. A *tree* is a connected graph which contains no circuit.[3] In game theory we often refer to the vertices as *nodes* and, consistent with our dendro-metaphor, to edges as *branches*.[4] Because a tree is a connected graph, any node can be reached from any other node by traversing a sequence of branches. Because it has no circuits, a tree is *unicursal*: for any pair of nodes, there is exactly one path which runs from the first to the second.

Example: graphs and trees

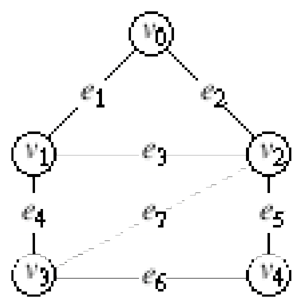

Figure 1: A graph.

The sets of vertices and edges in Figure 1 form a graph. The edges depicted graphically are defined by

$e_1=(v_0,v_1)$, $e_2=(v_0,v_2)$, $e_3=(v_1,v_2)$, $e_4=(v_1,v_3)$,

$e_5=(v_2,v_4)$, $e_6=(v_3,v_4)$, $e_7=(v_2,v_3)$.

The following two alternating sequences of vertices and edges are paths:

$\{v_0,e_1,v_1\}$, □ $\{v_0,e_2,v_2,e_5,v_4,e_6,v_3,e_7,v_2,e_3,v_1\}$.

Note in the second case that the vertex v_2 is encountered twice. The sequence $\{v_0,e_1,v_2\}$ is not a path. Neither is

[1] Let $e=(v,v')$ and $e'=(v',v)$ be edges, where v and v' are vertices. Because edges are *unordered* pairs, we have $e=e'$.

[2] This "runs from… to" terminology is not standard, but I find it useful for expository purposes here.

[3] I.e., no circuit can be constructed from vertices and edges belonging to the graph.

[4] My treatment roughly follows that of Kuhn [1953], which is the hugely influential, seminal paper concerning extensive-form games.

$\{v_0, e_1, v_1, e_3, v_2, e_7, v_3, e_4, v_1, e_3, v_2\}$,

because the edge e_3 is encountered twice.

The two sequences

$\{v_0, e_2, v_2, e_5, v_4, e_6, v_3, e_4, v_1, e_1, v_0\}$, □ $\{v_0, e_1, v_1, e_3, v_2, e_7, v_3, e_6, v_4, e_5, v_2, e_2, v_0\}$

are closed paths. The first is a circuit; the second is not, because the vertex v_2 is encountered twice.

Figure 2 shows two connected graphs. The first is a tree; the second is not.

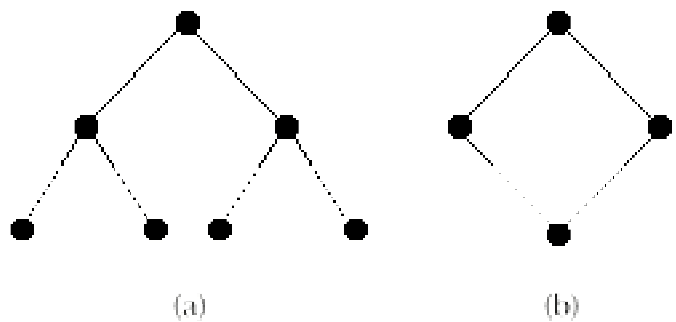

Figure 2: Two connected graphs: (a) a tree and (b) a nontree.

Now we want to enhance our tree to become a *game tree*. We henceforth assume that the tree has at least two vertices. We bestow one vertex of the tree with the honor of being the *initial node*, $\mathbb{O}$. (See Figure 3.) This is where the game begins. It is a *decision node*.

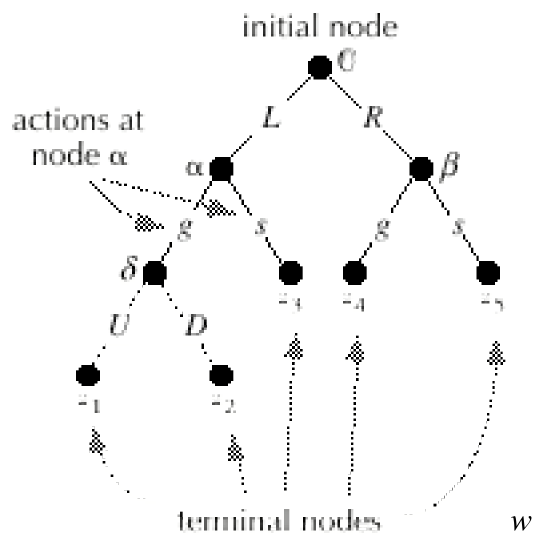

Figure 3: A game tree.

If a noninitial node x is incident with two or more branches, it is also a decision node. Let $X \subset V$ be the set of decision nodes. (In Figure 3, $X = \{\mathbb{O}, \alpha, \beta, \delta\}$.) One of these branches leads *to* the node; the remaining branches lead *away* from the node. Since branches are unordered pairs of nodes, how do you know which branch leads to the node and which lead away? It's simple: For some noninitial decision node $x\in X\backslash\{\mathbb{O}\}$, find the unique path which runs from the initial node $\mathbb{O}$ to node x. That path will contain exactly one of the branches incident with x; this branch is the one which leads *to* x. The other branches incident at v are the branches which lead away from x. For example, in Figure 3, there are three edges incident at node δ: g, U, and D. The unique path running from the initial node to node δ is $\{\mathbb{O}, L, \alpha, g, \delta\}$. This path contains only the g edge incident at δ; therefore g is the branch leading to δ and U and D are the edges leading away.

We interpret each edge $e\in E$ as an *action*. Let $\bar{A}$ be the set of all actions and let $f\colon E\to\bar{A}$ be the function which assigns an action to each edge. The branches which lead away from a decision node represent the actions available at that node. We require that, for a given decision node $x\in X$, the function f assign a unique action to each branch leading away from x. (Otherwise it would not be well specified what path would result from a particular action taken at x.) The actions available at any decision node $x\in X$ are given by the set $A(x)=f(\{e\in E\colon e \text{ leads away from } x\})$.[5] (So A is a correspondence $A\colon X\twoheadrightarrow\bar{A}$.) In Figure 3, $A(\mathbb{O})=\{L,R\}$, $A(\alpha)=A(\beta)=\{g,s\}$, and $A(\delta)=\{U,D\}$. More than one edge may correspond to the same action (as long as any such pair of edges does not lead away from the same node). For example, in Figure 3, there are two edges which correspond to the action g. In what follows I will often refer to an edge $e\in E$ by its associated action $f(e)$.[6]

Any noninitial node with only one incident branch necessarily has no actions available. Such a node is a *terminal node*. Let $Z\subset V$ be the set of terminal nodes. The tree in Figure 3 has five terminal nodes: $Z=\{z_1, z_2, z_3, z_4, z_5\}$. The decision and terminal nodes partition the game tree's nodes: i.e. $V=X\cup Z$ and $X\cap Z=\varnothing$. The game ends whenever a terminal node is reached. We identify the terminal nodes with *outcomes* of the game. (Sometimes *outcome* is used to refer not just to a terminal node but to the unique path from the initial node to that terminal node.)

So we have established that the game begins at the initial node and play ends at a terminal node. In what order are the intermediate decisions made? Consider two distinct nodes $v, v'\in V$, $v\neq v'$. We say that v *precedes* v', or that v is a *predecessor* of v', if v lies on the unique path which runs from the initial node $\mathbb{O}$ to v'. We write $v\prec v'$. Equivalently, we say that v' is a *successor* of v, and write $v'\succ v$. For example, in Figure 3, the unique path running from the initial node $\mathbb{O}$ to δ is $\{\mathbb{O}, L, \alpha, g, \delta\}$. This path contains α but not contain β; therefore α is a predecessor of δ, but β is not.

Predecession, then, defines a binary relation $\prec$ on the set of nodes V, viz.

$$\prec\,=\{(v,v')\in V^2\colon v\prec v'\}.^{7,8} \tag{1}$$

---

5 Recall that if $f\colon X\to Y$ is a function and $S\subset X$, then $f(S)$ is the *image* of S under f, i.e. $f(S)=\{f(x)\colon x\in S\}$.

6 For example, in the previous paragraph I referred to the edge connecting α and δ by its action g.

7 A *binary relation* R on a set X can be thought of as a subset of the Cartesian product of X with itself, i.e. $R\subset X\times X$, which includes all those pairs $(x,y)\in X^2$ such that xRy. I.e. $(x,y)\in R\Leftrightarrow xRy$.

8 The *succession* binary relation $\succ$ is the *inverse relation* of $\prec$; i.e. $\forall v, v'\in V$, $v\prec v'\Leftrightarrow v'\succ v$; it could also be called the *dual order* (except for a minor problem that $\prec$ is irreflexive and therefore not actually providing an *ordering*). (See later footnote.)

It is intuitively obvious—and you can show rigorously—that the predecession relation is *irreflexive* ($\forall v \in V$, not $v \prec v$), *asymmetric* [$v \prec v' \Rightarrow$ not $(v' \prec v)$], and *transitive* [$(v \prec v'$ and $v' \prec v'') \Rightarrow v \prec v''$]. This binary relation need not be complete; there may exist a pair of nodes $v, v' \in V$ such that neither $v \prec v'$ nor $v' \prec v$. For example in Figure 3, α and β are unordered by precedence. A partial listing of the precedence relations in Figure 3 is: $\mathbb{O} \prec z_3$, $\alpha \prec z_2$, and $\alpha \prec \delta$.

The initial node is a predecessor of every noninitial node and has no predecessor itself. For every noninitial node $x \in V \backslash \{\mathbb{O}\}$, let $P(x)$ be the set of predecessors of x; i.e. $P(x) = \{v \in V\colon\ v \prec x\}$. For example in Figure 3, $P(\delta) = \{\mathbb{O}, \alpha\}$ and $P(z_4) = \{\mathbb{O}, \beta\}$. Although predecession provides only a partial ordering in general on the set V of all nodes, you can show that it totally orders the set $P(x)$ of predecessors of any noninitial node $x \in V \backslash \{\mathbb{O}\}$. (I.e. if $v, v' \in P(x)$ and $v \neq v'$, then either $v \prec v'$ or $v' \prec v$.[9])[10]

We say that v is the *immediate predecessor* of v' if there is an action available at v (i.e. a branch incident at v which leads away from the initial node) which leads directly to v'. (This requires that $v \prec v'$ and that a single edge joins v and v'. You can prove that each noninitial node has a unique immediate predecessor.) Alternatively we could say that v is the immediate predecessor of v' if $v = \max_{\prec} P(v')$, which means that all other predecessors of v' must also precede v.[11] In Figure 3, for example, $\mathbb{O}$ is the immediate predecessor of α but not of δ, and α is the immediate predecessor of δ.

We have thus far identified the locations in the tree at which decisions are made—the decision nodes—and created a structure which relates the various decisions by specifying what actions lead to which other decisions and in what temporal order. But we still haven't said who makes which decisions. We retain our standard player set $I = \{1, \ldots, n\}$, but we also allow Nature to be player 0 when required. (We will see later that Nature can be given the job of introducing new information into the game.)

So now we assign to each decision node $x \in X$ exactly one player $i \in I \cup \{0\}$. We specify this assignment by a node-ownership function $\iota\colon X \to (I \cup \{0\})$ where each decision node $x \in X$ is assigned to the player specified by $\iota(x)$. The set of decision nodes assigned to player $i \in I \cup \{0\}$ is $X_i = \{x \in X\colon \iota(x) = i\}$. The node-ownership function ι thus induces a *player partition* $\mathfrak{X} = \{X_0, X_1, \ldots, X_n\}$ of the set X of decision nodes.[12] We say that a node *belongs to* a particular player or that a player *owns* a particular node.

Every terminal node $z \in Z$ represents some outcome which affects the players. Perhaps it's success for

---

9 Note that if a node $x \in V$ has only one predecessor, so that $P(x)$ is a singleton set, then $P(x)$ is trivially totally ordered by the binary relation $\prec$.

10 In my experience it is conventional to require that a binary relation R on a set X be reflexive (i.e. $\forall x \in X$, xRx) in order that it impose a preordering upon X. (See Debreu [1959: 7] and Iyanaga and Kawada [1980: 958, 960].) Kreps [1990: 364] explicitly weakens this notion by defining that "a set is totally ordered by a binary relation if any two *distinct* elements of the set are ordered in one direction or the other." [emphasis added] This allows the concept of totally ordered to be usefully applied to irreflexive binary relations. Note that for reflexive binary relations the two definitions are equivalent. Kreps [1988: 8] provides alternative terminology for this concept by saying that a binary relation R on a set X is *weakly connected* if for all $x, y \in X$, either $x = y$, xRy, or yRx.

11 Because $P(v')$ is totally ordered by the binary relation $\prec$, we can arrange its elements such that $v^1 \prec v^2 \prec \cdots \prec v^k$. Then $v^k = \max_{\prec} P(v')$.

12 A *partition* of a set A is a collection of cells $A_1, \ldots, A_n$, such that 1 each $A_i \subset A$, 2 the collection is exhaustive, i.e. $A_1 \cup A_2 \cup \cdots \cup A_n = A$, and 3 the cells are mutually exclusive, i.e. when $i \neq j$, $A_i \cap A_j = \varnothing$.

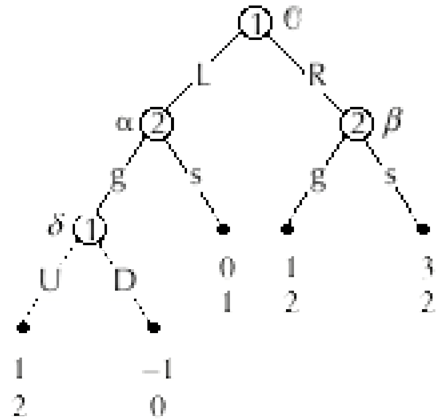

Figure 4: An extensive-form game indicating the player partition and payoffs as a function of outcome.

one and failure for another; the outcomes may represent profit levels for each firm or lengths of prison sentences for each criminal suspect. Whatever the outcomes are, they are events over which each player has preferences. We assume that each player has von Neumann-Morgenstern preferences, which can be represented by a utility function μ_i: $Z \to \mathbb{R}$ with the expected-utility property, and seeks to maximize her expected utility.[13]

Figure 4 shows an extensive-form game elaborated from the tree in Figure 3. I have indicated the player partition, which assigns players to nodes, by numerical labels and assigned payoff vectors to each terminal node. By convention the upper entry in each payoff vector is the first player's utility derived from that outcome; the second entry is the second player's utility. The player partition is given by $X_1 = \{\mathbb{O}, \delta\}$ and $X_2 = \{\alpha, \beta\}$. (Nature does not appear in this game.)

Information sets

In the first paragraph I identified five aspects of an economic scenario we'd capture using the extensive form. So far we've captured the set of players, who moves when and under what conditions, what actions are available to a player when it's her turn to move, and what payoffs are received at the end of the game. The remaining challenge is to represent the information which a player knows when called upon to move.

What kind of information do I have in mind? We need to be concerned about things which have changed during the course of the game as a result of the actions of the players.[14] The key to representing information in the game tree is realizing the connection between nodes and *history*. At any point in the play of the game the history is a record of who did what when. Remember that game trees are

[13] I'm using μ_i to represent the utility to player i of an outcome $z \in Z$ because I want to reserve the symbol u_i for later use as her utility to a strategy profile. (We will see that an outcome $z \in Z$ could be the result of any of several strategy profiles.)

[14] If information is entering the game in some other fashion, this can be modeled by the addition of Nature as a nonstrategic player who chooses her strategy according to a probability distribution. The revelation of Nature's moves then communicates the exogenously generated information to the players.

unicursal—there's precisely one path which runs from the initial node to any given node. Therefore, if you know the node you have reached, you also know precisely the history of play that got you there.[15,16]

We can divide extensive-form games into two classes: In games of *perfect information*, whenever it is your turn to move you know precisely where in the tree you are located. Therefore you know the history of play completely. In games of *imperfect information*, you might be uncertain about the history of play when you are called on to play. This uncertainty about history corresponds to uncertainty about which of several nodes you might be at.

To express this uncertainty we employ the *information set*. An information set $h \subset X$ is a set of decision nodes. In the course of playing the game the players' actions may cause one of your nodes to be reached. When this occurs, you might not know precisely which of your nodes has been reached. This uncertainty is captured by saying that you only know that a particular information set has been reached; all you know is that you've reached one of the nodes $x \in h$ in that information set. It is as if you are sent a message, whenever it is your move, specifying the information set you have reached.

There are some obvious restrictions we need to impose on the composition of information sets. First, each decision node must belong to exactly one information set. (If a node belonged to more than one information set, it wouldn't be well defined what message its owner should receive when that node is reached.) Therefore the set H of all information sets forms a partition of the decision nodes: $X = \bigcup_{h \in H} h$ and $\forall h, h' \in H$, $h \cap h' = \varnothing$. We can also define an information-set membership function $\hat{h}$: $X \to H$, which specifies an information set $h \in H$ for each decision node $x \in X$ according to $\hat{h}(x) = h \Leftrightarrow x \in h$.

Secondly, all of the nodes in an information set must belong to the same player. (I.e. $\forall h \in H$, $\forall x, x' \in h$, $\iota(x) = \iota(x')$.) Otherwise, two or more players may think it is their turn to move.[17] (Therefore the information partition H is a *refinement* of the player partition $\mathcal{X} = \{X_0, X_1, \ldots, X_n\}$, because for each information set $h \in H$ there exists a player $i \in I \cup \{0\}$ such that this information set is entirely contained within player i's set of nodes, i.e. such that $h \subset X_i$.[18])

We can partition the set H of information sets into sets of information sets which belong to each player: For every player $i \in I \cup \{0\}$, the set $H_i = \{h \in H\colon \forall x \in h, \iota(x) = i\}$ contains the information sets

15 Note that this wouldn't be the case if we were playing the game represented by the nontree of Figure 1b. If we had reached the bottom node, we wouldn't know whether we had arrived there via the clockwise or counterclockwise path from the top initial node.

16 You might object that in the real world there often are many different ways to reach the same result. Such a situation can be modeled within this framework. You would still have a different terminal node for each of these histories; but you would assign the same payoff vector to each of these terminal nodes.

17 "But what's wrong with that? Perhaps it's a simultaneous move game. Then it would be perfectly kosher to have two players believing it's their turn," you say. In an extensive-form representation a simultaneous-move game is modeled as a sequential game where the second mover doesn't get to observe the first mover's choice at least until after the second mover has played. We'll discuss simultaneity further later.

18 One partition is a *refinement* of a second partition if each cell of the first is a subset of some cell of the second. More formally…. Let $\mathfrak{M} = \{M_i\}_{i \in \{1,\ldots,m\}}$ and $\mathfrak{N} = \{N_i\}_{i \in \{1,\ldots,n\}}$ be two partitions. $\mathcal{M}$ is a refinement of $\mathcal{N}$, denoted $\mathfrak{M} \prec \mathfrak{N}$, if $\forall M_i \in \mathfrak{M}$, $\exists N_j \in \mathfrak{N}$ such that $M_i \subset N_j$. In such a case we say that $\mathfrak{N}$ is *coarser* than $\mathfrak{M}$.

owned by player i. Therefore, for all $i \in I \cup \{0\}$, $\bigcup_{h \in H_i} h = X_i$.

In general, for any information set $h \in H$, $\iota(h) \equiv \{\iota(x)\colon x \in h\}$ would refer to the image of the set of nodes h under the node-ownership function ι; i.e. $\iota(h)$ would be the set of players who owned one or more nodes in h. However, since all nodes in h are owned by the same player, we see that $\iota(h)$ is single valued; i.e. $\forall h \in H$, $\#\iota(h) = 1$. Therefore we can also consider, with some abuse of notation, ι to be an information-set ownership function $\iota(h) = \{i \in I \cup \{0\}\colon \forall x \in h, \iota(x) = i\}$. As with decision nodes, we say that an information set belongs to a particular player or that a player owns a particular information set.

Thirdly, every node in an information set must have the same set of available actions. Otherwise the player wouldn't be sure which actions were truly available to her. I.e. $\forall h \in H, \forall x, x' \in h, A(x) = A(x')$. It is useful then to refer to the set of actions available at an information set $h \in H$, viz. the image of h under the correspondence $A\colon A(h) = \bigcup_{x \in h} A(x)$, because for all nodes in the information set $A(h)$ is the set of actions available there: $\forall x \in h, A(x) = A(h)$.

We can summarize these last two restrictions by

$$\forall h \in H,\ \forall x, x' \in h,\ \iota(x) = \iota(x') \text{ and } A(x) = A(x'). \tag{2}$$

An information set may be a *singleton*—it may consist of only a single node. If every information set of every player is a singleton, we have a game of perfect information; i.e. $\forall h \in H, \#h = 1$, or equivalently: $\forall i \in I \cup \{0\},\ \forall h \in H_i, \#h = 1$. If any information set contains more than one node—i.e. $\exists i \in I \cup \{0\}, \exists h \in H_i$, such that $\#h > 1$—then the game is one of imperfect information. When an information set contains more than one node, we indicate this graphically on the extensive form by connecting all of the nodes in that information set with a dashed line. Singleton information sets require no special ornamentation. For example, the game in Figure 4 has perfect information; therefore

$$H_1 = \{\{\copyright\}, \{\delta\}\}, \square H_2 = \{\{\alpha\}, \{\beta\}\}. \tag{3}$$

Return to the perfect information game of Figure 4. If we change the information structure so that player 1's first move is unobserved by player 2 by the time player 2 makes his first move, then he doesn't know at which of his two nodes he is located. We indicate this information imperfection by connecting these two nodes with a dashed line. (See Figure 5a.) Each of player 1's two nodes belongs to its own information set. In this game then there are three information sets: two belonging to the first player and one belonging to the second. Note that, as required, player 2 has the same two actions, viz. g and s, available at both nodes of his information set; $A(\alpha) = A(\beta) = \{g, s\}$. Contrast each player's collection of information sets in this game, viz.

$$H_1 = \{\{\copyright\}, \{\delta\}\}, \square H_2 = \{\{\alpha, \beta\}\}, \tag{3}$$

with those given in (3) for the game in Figure 4.

The pseudo–extensive form of Figure 5b has two violations of the restrictions upon information sets. One information set consists of nodes belonging to different players. (Consider the information set $h = \{\alpha, \beta\}$, where $\iota(\alpha) = 2$ but $\iota(\beta) = 3$.) A second information set has different actions available to its

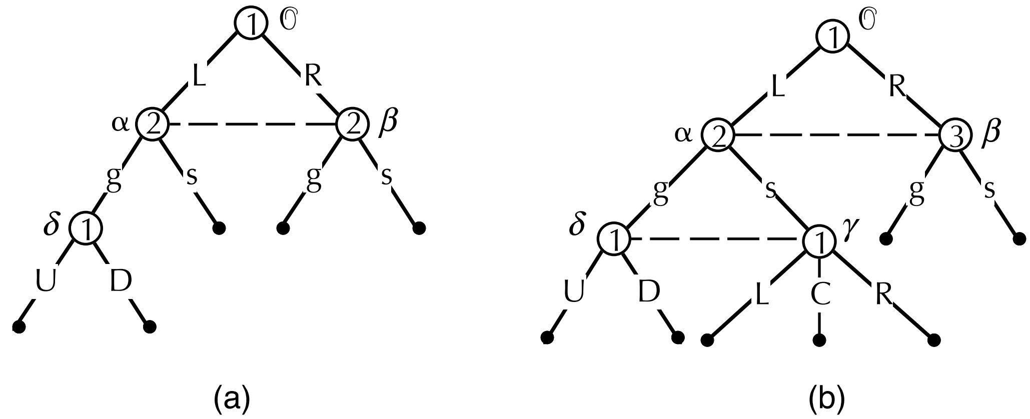

Figure 5: A game of (a) imperfect information and (b) implausible information.

player at different nodes. (Consider the information set $h'=\{\delta,\gamma\}$, where $A(\delta)=\{U,D\}$ but $A(\gamma)=\{L,C,R\}$.)

In summary, then, an extensive-form game Γ is defined by a tree (V,E), a distinguished vertex $\mathbb{O}$, a set of actions $\bar{A}$, an action-for-edge assignment function f: $E\to\bar{A}$, an information partition H of the decision nodes X, a player set I, a node-ownership function ι: $X\to I\cup\{0\}$, and player payoff functions μ_i: $Z\to\mathbb{R}$ for $i\in I$. From this definition the following entities can be derived: the set X of decision nodes, the set Z of terminal nodes, the set of actions $A(x)$ available at each decision node $x\in X$, the player partition $\mathfrak{X}$, and the precedence relation $\prec$ on V. It is further required that, for each information set $h\in H$, every node in h shares the same player-owner and shares the same set of available actions: $\forall h\in H$, $\forall x,x'\in H$, $\iota(x)=\iota(x')$ and $A(x)=A(x')$.

Games of perfect recall

We've established some restrictions on what nodes we can include in the same information set: We can only draw dotted lines which connect nodes which belong to the same player and at which the same set of actions is available. Now I'll present two extensive-form fragments which might suggest additional restrictions.

In Figure 6a, at node α player 1 can choose Left and turn the move over to player 2 or she can choose Right and immediately move again. If she chooses Right, she reaches node β, which is in the same information set as the node α from which she had just come. Note what this particular information set construction implies: When player 1 chose Right at node α, she realized what she was doing. When she then reaches β, she doesn't know whether she's at β or back at α. Therefore she no longer knows that she chose Right when at α; she has forgotten the action she just took.

It is typical in game theory to restrict ourselves to games of *perfect recall.* In such games a player always know what actions she has taken and never forgets anything she ever knew. How do we

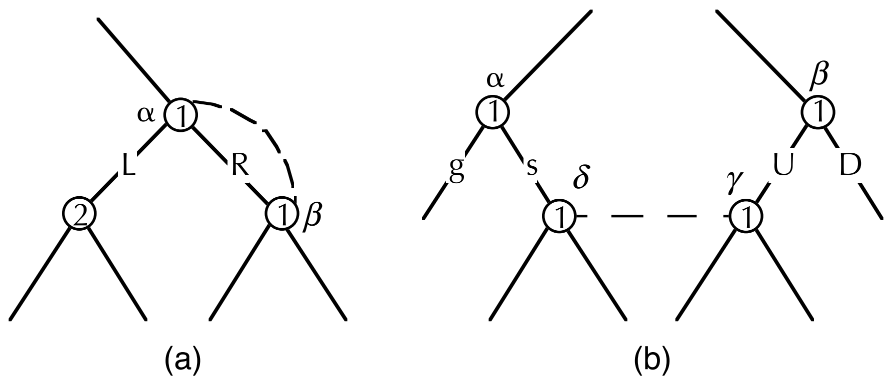

Figure 6: Two game fragments with forgetful players.

formalize this restriction?[19] We can rule out the game fragment in Figure 6a with the following requirement: If two nodes are in the same information set, then neither can precede the other. ($\forall h\in H$, $\forall x,x'\in h$, neither $x\prec x'$ nor $x'\prec x$.)

However, that requirement isn't sufficient. Consider Figure 6b. Nodes δ and γ are both in the same information set. There are two ways player 1 can reach that information set: She can be at node α and choose stop, or she can start at node β and play Up. Note that α and β are not in the same information set; therefore they represent different information—different histories. If player 1 reaches α she knows that she reached α but never passed through β; if she reaches β she knows she reached β but never passed through α. But if she reaches the $\{\gamma,\delta\}$ information set via node α, say, she cannot possibly know that she didn't pass through β. (If she did know this, then she would know for certain that she is at node δ. But that would violate the information set's knowledge restriction that she doesn't know which of the two nodes δ and γ she is at.) Therefore she has forgotten something she knew earlier; so this would be a violation of perfect recall.

The extensive-form fragment of Figure 6b doesn't violate the prohibition that members of the same information set should not be ordered by precedence; so we need a stronger restriction. The formal characterization of this restriction is extremely arcane and abstruse.[20] I'll try to motivate it the best I can.

Consider two nodes δ and γ which belong to the same information set for player 1. Now consider a player-1 node α which precedes δ when player 1 chooses the action a at α. (See Figure 7a.[21]) When

---

19 There are two reasons to be interested in perfect recall. The first is that most if not all games we would be interested in analyzing would satisfy this property. (But for precisely this reason there would be little point in formalizing the restriction.) More important theoretically, however, is that the restriction to games of perfect recall allows a simplification in the analysis of these games: We will see later that in games of perfect recall there is an equivalence between mixed strategies and *behavior* strategies such that we can choose whichever analytical perspective is more convenient.

20 See Fudenberg and Tirole [1991: 81], Kreps and Wilson [1982: 867], Kreps [1990: 374–375], and Myerson [1991: 43–44]. The property so defined will prove sufficient for a later demonstration of the equivalence between mixed and behavior strategies. Kuhn [1953: 213] defines a weaker property, which he also calls perfect recall. Although this is a misnomer, his property is necessary and sufficient for the equivalence.

21 The "" indicates that α is a predecessor of δ but not necessarily an immediate predecessor.

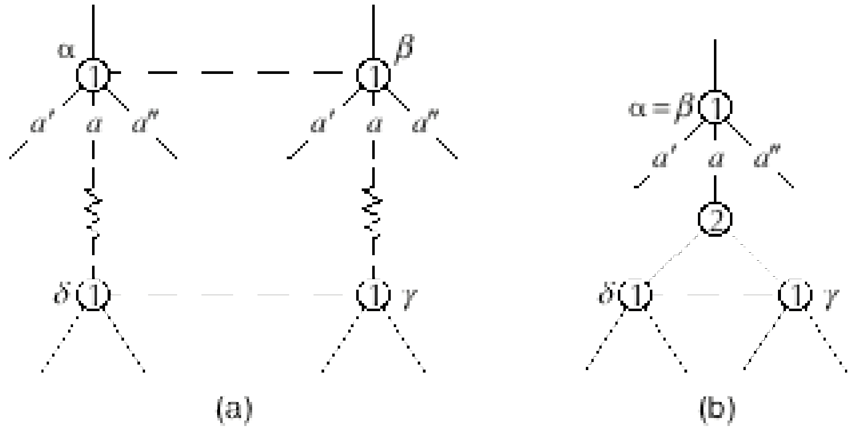

Figure 7: Establishing anti-amnesia requirements.

player 1 chooses a at α, she knows 1 that she has passed through α's information set and 2 that she chose the action a at α's information set. In order that the requirements of perfect recall be satisfied, she must still know these two facts when she gets to δ.

Indeed, even if she instead arrives at a different node, say γ, in δ's information set, she must still know exactly the same information she would know at δ. (If she knew different facts at δ than at γ, she would be able to distinguish between the two nodes, contradicting that they are in the same information set.) Therefore at γ she must also know that she passed through α's information set and chose a there. So we require that every node γ in δ's information set have a predecessor β in α's information set such that choosing a at β leads eventually to δ. (We don't rule out that α and β may be the same node. This case is shown in Figure 7b.)

Hopefully the following requirement which expresses that an extensive form satisfies perfect recall will now make sense: For any player-i nodes α, δ, and γ such that 1 δ and γ belong to the same information set and 2 α precedes δ via action a at α, it must be the case that there exists a player-i node β which belongs to α's information set and which precedes γ via action a at β.[22]

Before we state this more formally, let's develop one additional piece of notation. Consider a node α which precedes a node β. Let a be the action at node α which is required in order for play to arrive at β. Then we say that $\alpha \oplus a \twoheadrightarrow \beta$. In other words the unique path which runs from α to β is of the form: $\{\alpha, a, \ldots, \beta\}$.

Now we can say that a game satisfies perfect recall if: for all triples of decision nodes, $\alpha, \delta, \gamma \in X$, and for all actions $a \in A(\alpha)$ available at node α such that 1 $\iota(\alpha) = \iota(\delta) = \iota(\gamma)$, b $\hat{h}(\delta) = \hat{h}(\gamma)$, and c $\alpha \oplus a \twoheadrightarrow \delta$, there exists a node $\beta \in \hat{h}(\alpha)$ such that $\beta \oplus a \twoheadrightarrow \gamma$.

22 We earlier required that, if two nodes belong to the same information set, neither can precede the other. You can show that this original requirement is implied by the newer condition.

Arborescences

The graph-theoretic formulation of the extensive form we have presented here originated with Harold Kuhn [1953]. Kreps and Wilson [1982] offered an alternative formulation, where they replaced the standard definition of a tree with an entity called an *arborescence*.[23] They claim that their formulation is equivalent to Kuhn's. I will briefly present the relevant part of the Kreps-Wilson formulation in order to demonstrate that the arborescence requirement is unintentionally too weak to guarantee that Krepsonian extensive forms are legitimate game trees. (Ratliff [1991])

We earlier defined a tree as a set of nodes and a set of branches (each of which joined a pair of nodes). Kreps and Wilson define an arborescence as a set of nodes and a binary relation $\prec$, called *precedence*, which is asymmetric, transitive, and such that the set of predecessors of a given node is totally ordered by $\prec$. This "totally ordered" condition means that: if two nodes α and β both precede a third node γ, then either α precedes β or β precedes α.[24]

These properties of the precedence relation $\prec$ are well suited to rule out many node configurations which would not be legitimate trees. For example, transitivity combined with asymmetry rules out the "cycle in play" configuration in Figure 8a: From the diagram we see that $\alpha\prec\beta$, $\beta\prec\gamma$, $\gamma\prec\delta$ and $\delta\prec\alpha$. Transitivity then implies that $\alpha\prec\alpha$, which violates asymmetry.

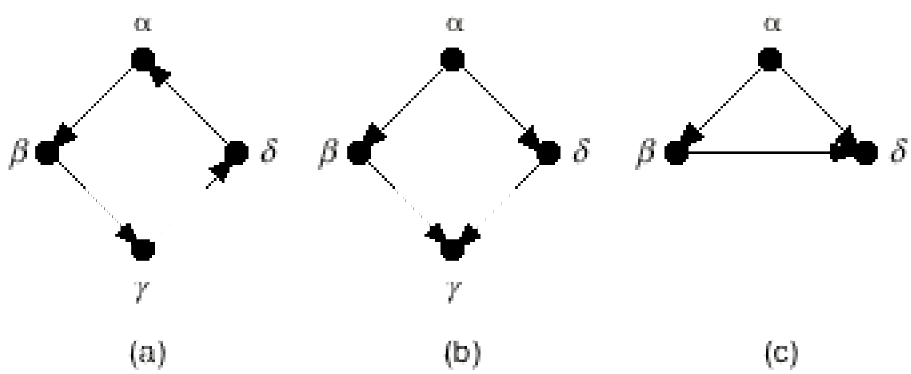

Figure 8: Three nontrees; two are captured by the Kreps-Wilson arborescence filter; one escapes.

The connected graph in Figure 8b is not unicursal; there are two paths joining α and γ, for example. This configuration is ruled out by the requirement that the precedence relation totally order the predecessors of any given node. To see this we observe that α, β, and δ are all predecessors of γ. However, the set $\{\alpha,\beta,\delta\}$ is not totally ordered by $\prec$, because β and δ are not related by precedence; i.e. we have neither $\beta\prec\delta$ nor $\delta\prec\beta$.

An even simpler nontree does slip through their filter, however. Consider the three-node connected

[23] See also Kreps [1990: section 11.2, especially pages 363–365].

[24] Note that these are exactly the properties we have already attributed to the precedence relation we defined in the graph-theoretic tree framework. What Kreps and Wilson are trying to do here is to go the other way: derive a tree from the binary relation itself.

graph in Figure 8c. We observe that the predecessors of each node are totally ordered by the precedence relation $\prec$: The node α has no predecessors, so satisfaction is trivial. The node β has α as its single predecessor; a singleton set is trivially ordered. The node δ has two predecessors, viz. α and β. These are ordered because $\alpha \prec \beta$. Therefore the connected graph of Figure 8c is an arborescence but is definitely not a tree, because it contains a circuit. Therefore there exist arborescences which are not trees; therefore an arborescence need not be a legitimate model for an extensive-form game.

Simultaneity

Consider the case in which two players will each take an action at the same time. Clearly an important implication of this temporal structure is that neither player can observe the other's choice before making her own choice. For example in Figure 9a we depict a game in which player 1 chooses from Up and Down and player 2 chooses from left and right, and these choices are made simultaneously.

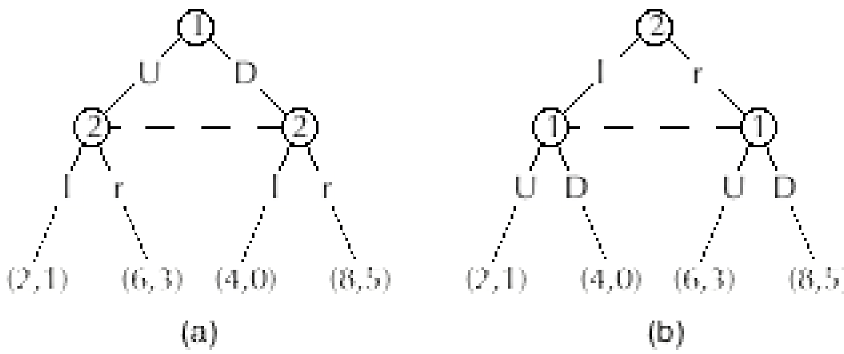

Figure 9: Two extensive forms of the same simultaneous-move game.

Our tree formalism requires that we represent the choices as being sequential. In order to provide equivalence between a sequential representation and a nonsequential game, we extract the relevant information structure from the simultaneous-move game and replicate it in the extensive form. In other words we connect player 2's nodes into a single information set. In this way we ensure that, even though his move is depicted as coming after player 1's, he does not observe player 1's move before he makes his own choice. By making his move in ignorance of his opponent's, player 2 is making his choice with exactly the same information he would have if the two players moved simultaneously. Note that it was arbitrary that we depicted player 1's choice as coming before player 2's. We could just as easily have reversed the roles. This would result in the extensive form of Figure 9b.

Divide and conquer: subgames

Extensive-form games can be very complicated. Tackling a huge game tree in a single bite can be an impossibly formidable undertaking. Fortunately, we can often identify pieces of large games which are themselves simpler games. If we first take the time to identify and understand these simpler pieces we can greatly ease our analytical burden. We will also be able to strengthen our to-be-introduced extensive-form solution concepts by imposing them not only on the game as a whole but also upon these pieces of a game. The pieces of a game we seek are called *subgames*.[25]

The concept of a subgame is intended to capture the notion of "today is the first day of the rest of your life." A subgame is a subset of the original game. In order to define a subgame we impose conditions which guarantee that 1 the subset makes sense as a game in its own right and 2 it must be played under the same informational conditions under which that subset would be played if encountered in the original game. (I.e. if this subgame were reached in the larger game, it would be common knowledge that this is the game remaining to be played.)

Consider an extensive-form game Γ. In order to define the subgames of Γ we first define how we would *decompose* the game Γ at some decision node $v^*\in X$ to create the extensive-form object Γ^*.[26] (Γ^* may not be a legitimate game for our purposes, so I describe it agnostically as merely an object.) If v^* happens to be the initial node $\mathbb{O}$, then let $\Gamma^*=\Gamma$. Otherwise, find the unique edge which immediately precedes v^* and delete it. Now we're left with two trees, only one of which contains v^*. (See Figure 10.) Let this tree be the graph $(\tilde{V},\tilde{E})$, where $\tilde{V}\subset V$ and $\tilde{E}\subset E$. This is the subtree we want. Designate $v^*\in\tilde{V}$ as the initial node of Γ^*. We note that $\tilde{V}$ consists only of v^* and its successors in the original game. This implies that Γ^* is *closed under succession*: i.e. if $v\in\Gamma^*$ and $v'\succ v$ in Γ, then $v'\in\Gamma^*$.

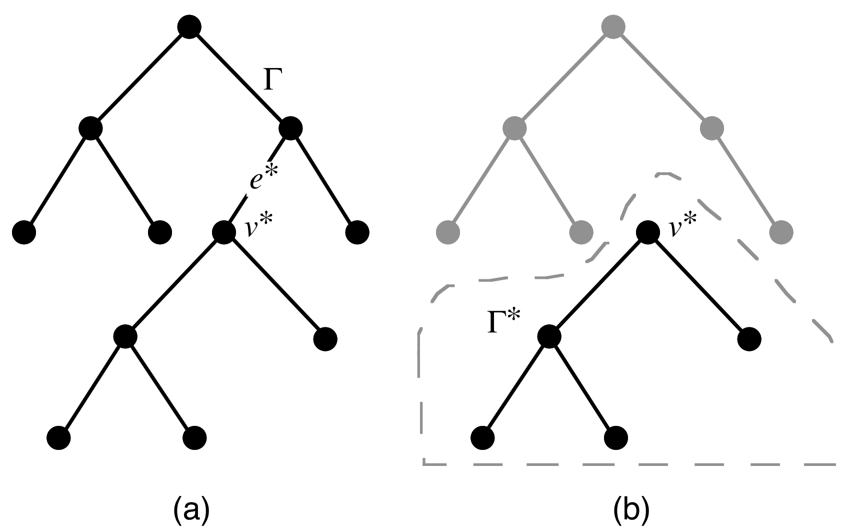

Figure 10: (a) A tree Γ. (b) Γ decomposed at v^* into two components by deleting the edge e^*.

To complete the specification of Γ^* we stipulate that it inherits much of the structure of the original extensive-form game Γ by *restriction* when necessary of the original entities to the new game tree.[27] Specifically we restrict the original action-for-edge assignment function f to the residual set of edges $\tilde{E}$,

---

[25] See Kreps and Wilson [1982: 865–869] and Fudenberg and Tirole [1991: 94–95].

[26] See Kuhn [1953: 203–204].

[27] Let $f\colon X\to Z$ be a function and let Y be a set with a nonempty intersection with the domain X, i.e. $X\cap Y\neq\varnothing$. Then we can define a function $\tilde{f}$, the *restriction* of f to Y, as a function whose domain is $X\cap Y$ and which agrees with f for all points in $X\cap Y$. I.e. $\tilde{f}\colon (X\cap Y)\to Z$ and $\forall x\in(X\cap Y), \tilde{f}(x)=f(x)$. In the special case where $Y\subset X, \tilde{f}\colon Y\to Z$ and $\forall x\in Y, \tilde{f}(x)=f(x)$.

and we restrict the node-ownership function ι and the payoff functions μ_i to the residual set of nodes $\tilde{V}$. We create a new information partition $\tilde{H}$ by replacing each information set $h \in H$ with $\tilde{h} = h \cap \tilde{V}$. The remaining entities—the residual set of decision nodes $\tilde{X} = X \cap \tilde{V}$ (note that any decision node $x \in X$ in the original game which also belongs to the new game is also a decision node in the new game), the residual set of terminal nodes $\tilde{Z} = Z \cap \tilde{V}$ (ditto for terminal nodes), the correspondence $A: \tilde{X} \twoheadrightarrow \bar{A}$ describing the actions available at each decision node, and the binary relation $\prec \subset \tilde{V} \times \tilde{V}$—can be derived from these restricted primitive objects.

Although Γ^* as we have defined it here *is* a well-formed game, it might not be a legitimate subgame for our purposes. We want to ensure that, if this new game were reached in the course of play of the original game Γ, it would be common knowledge to the players that it was indeed this game Γ^* remaining to be played. In particular we want a subgame to "preserve" or "respect" information sets: If $h \in H$ is an information set in the original game, then we want this information set to either appear completely intact as an information set in the new game or be totally excluded from the new game. We can state this requirement as $h \in H \Rightarrow (h \in \tilde{H}$ or $h \cap \tilde{X} = \varnothing)$.[28]

Note that this implies—as long as Γ is a game of perfect recall—that the new initial node v^* must have been a singleton information set in the original game.[29] Note that the entire game is always a subgame of itself.[30]

In summary, a subset of an extensive-form game (as formally defined by restriction above) forms a subgame if you can find a node v^* in the subset such that all of its successors and only its successors in the original game belong to this subset and "information sets are preserved."[31] This last condition requires that any information set of the original game is either totally absent from the subset or appears completely intact in the subset. The payoffs to the terminal nodes of the subgame are obtained by restriction of the payoffs of the original game to the terminal nodes of the subgame.

Figure 11 shows a game in extensive form with four conjectured subgames—A, B, C, and D—outlined.[32] A and B are true subgames. However, C is not a subgame because it cannot be formed from the original tree through decomposition by the deletion of a single edge. (Alternatively: because for

28 The extensive-from object Γ^* derived by decomposition from Γ is often called a *subgame* and, after imposing the restriction that information sets are preserved, becomes a *proper subgame*. However, proper subgames are the only kind of subgames we will be interested in. Therefore it would be quite inconvenient to have to attach the "proper" modifier every time we wanted to refer to one. Therefore I will use subgame only to refer to proper subgames in this sense.

29 Γ^* contains only v^* and its successors in the original game. If v^* shared a common information set in Γ with a distinct node v, then v would be required to belong to Γ^* (by preservation of information sets) and hence v must be a successor of v^* in Γ. However, if Γ is a game of perfect recall, then all nodes within an information set in Γ must be unordered by precedence, contradicting that $v^* \prec v$.

30 There is an occasional confusion about the meaning of "proper subgame": the definitions in Kreps and Wilson [1982: 868] and Kreps [1990: 423] agree with the explicit statement in Fudenberg and Tirole [1991: 94–95] that "proper" does *not* refer to strict inclusion as it does in the term "proper subset." Friedman [1990: 44] says indeed that a proper subgame *is* a subgame which is strictly smaller than the original game. Reader beware!

31 An alternative statement of the requirement that the subset contain exactly v^* and its successors would be: the subset is closed under succession and v^* precedes every other node in the subset.

32 The outlining boxes are not meant to indicate that the edge going into the first node is part of the conjectured subset. The conjectured subset begins with the first node itself.

any choice of decision node in C there would be nonsuccessors in C.[33]) D is not a true subgame because it does not preserve information sets: the information set to which α belongs in the original game is neither wholly contained nor wholly excluded from the conjectured subgame.

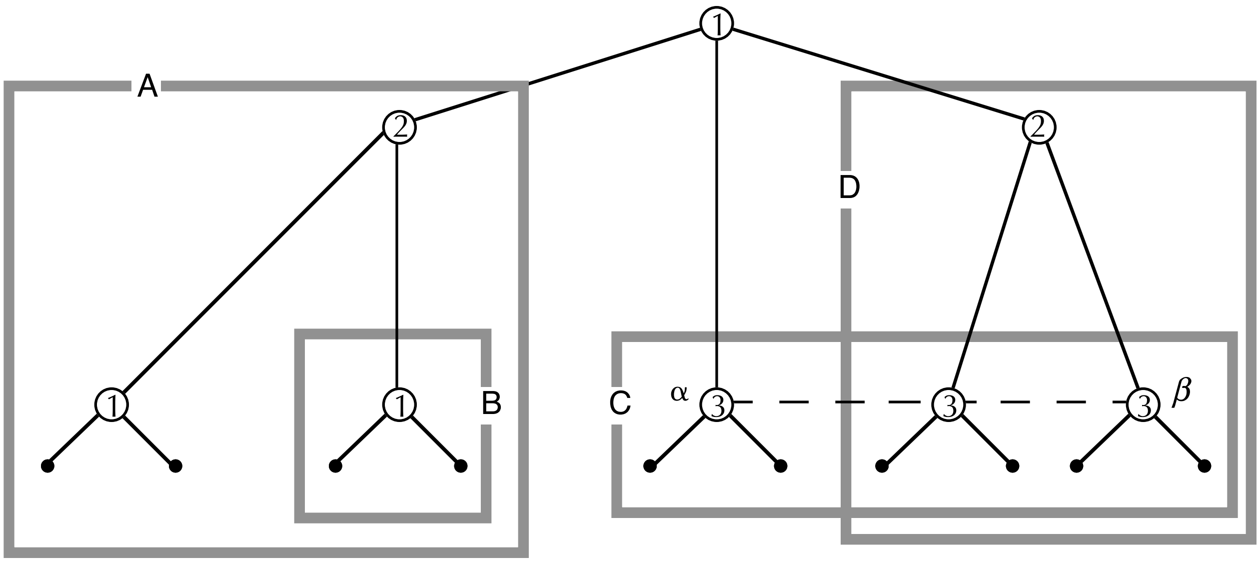

Figure 11: An extensive form and four conjectured subgames.

33 Another alternative argument: although C is closed under succession, there does not exist a node in C which precedes all other nodes in C.

References

Debreu, Gerard [1959] *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*, Series: Cowles Foundation Monographs, 17, Yale University Press.

Friedman, James W. [1990] *Game Theory with Applications to Economics*, second edition, Oxford University Press.

Fudenberg, Drew and Jean Tirole [1991] *Game Theory*, MIT Press.

Iyanaga, Shôkichi and Yukiyosi Kawada, eds. [1980] *Encyclopedic Dictionary of Mathematics*, MIT Press.

Kreps, David M. [1988] *Notes on the Theory of Choice*, Series: Underground Classics in Economics, Westview Press.

Kreps, David M. [1990] *A Course in Microeconomic Theory*, Princeton University Press.

Kreps, David M. and Robert B. Wilson [1982] "Sequential Equilibria," *Econometrica* **50** (July): 863–894.

Kuhn, Harold W. [1953] "Extensive Games and the Problem of Information," in *Contributions to the Theory of Games*, eds. Harold W. Kuhn and A.W. Tucker, Vol. 2, Princeton University Press, pp. 193–216.

Myerson, Roger B. [1991] *Game Theory: Analysis of Conflict*, Harvard University Press.

Ratliff, James D. [1991] "Arborescences and Game Trees," U.C. Berkeley, mimeo.